

DELHI TECHNOLOGICAL UNIVERSITY



Database Management Project

MC-302

SUBMITTED TO:

Ms. Goonjan Jain

SUBMITTED BY:

Kashvi Srivastava – 2K16/MC/042

Khyati Mahendru – 2K16/MC/045

Mihir Ahlawat – 2K16/MC/051

ACKNOWLEDGEMENT

We are thankful to our Database Management teacher Ms. Goonjan Jain for her constant guidance and support. She enlightened us whenever required and helped us tremendously in the successful completion of this report. Moreover, we worked as an efficient team and learnt interpersonal skills as well.

ABSTRACT

The following report aims at providing a comprehensive summary to our database management project. The database is designed for a group of fire stations which come under the administration of a single management organization. This can be limited to a particular geographical area handled by the organization or can be used for administration of fire stations spread across a large area. Further, the database is designed from the point of view of admin of the organization. We plan to incorporate many entities and inter-entity relationships to truly depict the management of fire stations and its functions over the course of time. The database is supported with admin-based interface to view, modify and manipulate the data with efficiency and convenience. Further, the user based interface allows a guest user to create an account and report fire incidents with all the details.

INTRODUCTION

The database is designed on the basis of an entity-relationship model. A copy of the mentioned model is attached after this. The model is further supported by an easy to use interface for the convenient modification and retrieval of information. This information can be entered and accessed by the administrating body. An official creates credentials of a login portal through which he can access the data regarding various entities such as a fire station, staff member or a dated fire report. He can further add, update or delete the data through the portal. He can also view the entire data specific to a certain entity and its attributes at one place through the same. Alike an admin, a user can create his or her account on the portal to report a fire incident with all its details but he only gets access to the feature of adding a report. Further detailed break-down of the model and its interface is given in the following sections.

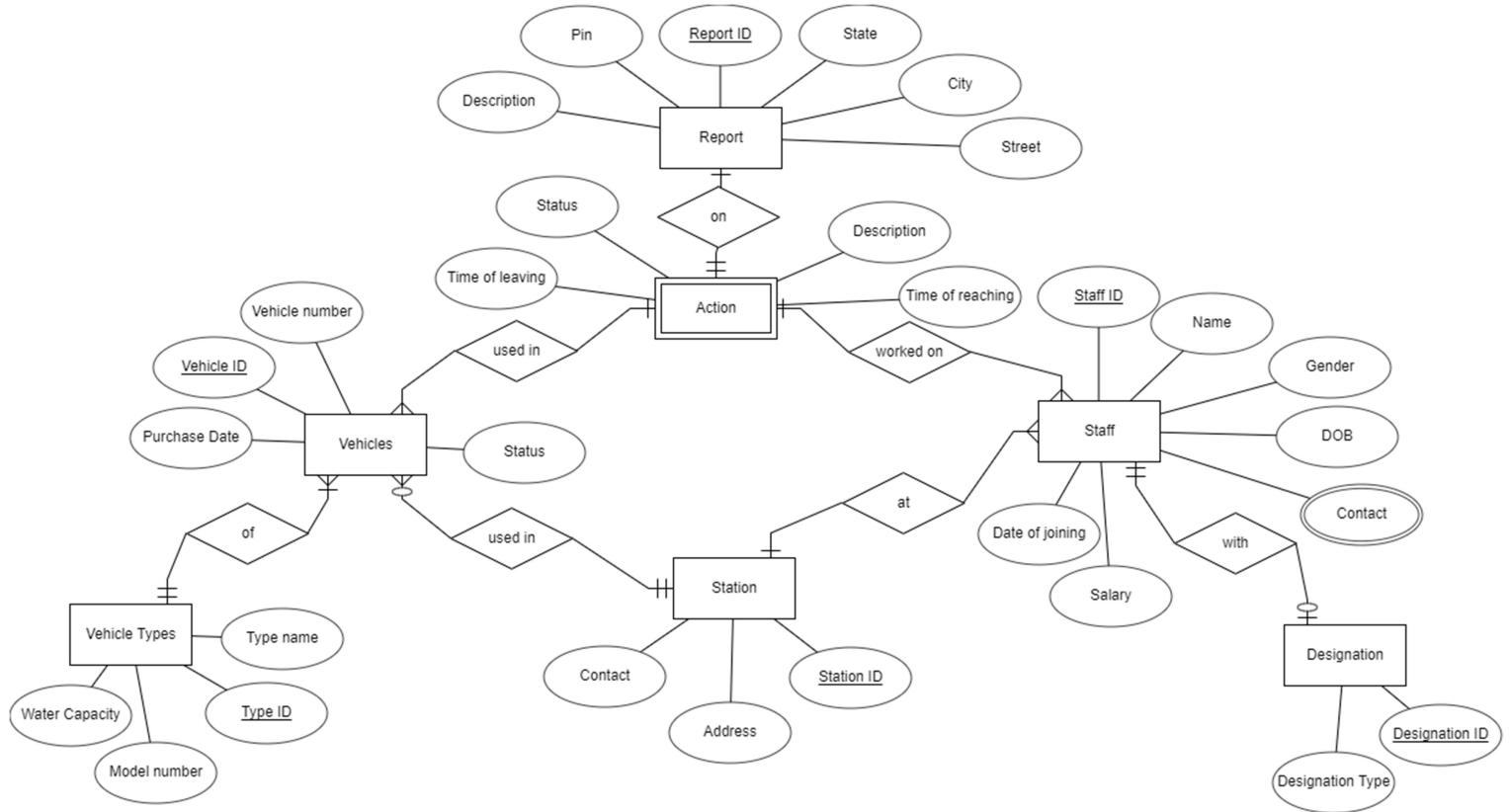
For the introduction part of our proposed database model, we shall consider an example of Delhi Fire Service and its functionalities. The entities used in our model and their examples as per the Delhi Fire Management are given as follows:

Entity	Referred Example
Station	East division – New Delhi Fire Zone
Vehicle	Fire Truck
Staff	Chief Fire Officer
Report	Fire report from Dwarka dated 21-04-19
Action on Report	South division associated

Other attributes of the entities and their relationship among one another are further explained after the entity-relationship model.

Our aim with the above database model is to efficiently manage related information. Along with this, one of the prospective benefits of designing such a project is to analyse the current requirements of management of fire incidents which are not being met by the fire organization. Studying the frequency and distribution of reports and how prompt is the action on these reports can help conclude if there is a lag in management in fire prone areas. Moreover, details about the staff members and vehicles can be easily managed.

Entity Relationship Diagram



Database model

1. Login Page

```
class LoginPage(QMainWindow, login_ui):
    def __init__(self):
        QMainWindow.__init__(self)
        self.setupUi(self)
        self.Handle_Buttons()
        self.style_3()

    def Handle_Buttons(self):
        self.createAccount.clicked.connect(self.create)
        self.cr_clear.clicked.connect(self.create_clr)
        self.log_in.clicked.connect(self.login)
        self.log_clear.clicked.connect(self.login_clr)

    def adminpass(self):
        self.window = MainApp()
        self.close()
        self.window.show()

    def userpass(self):
        self.window = UserPage(self)
        self.close()
        self.window.show()

    def create(self):
        self.db = pymysql.connect(host='localhost', user= 'root',
password= 'admin', db='firestation')
        self.cur = self.db.cursor()

        nam = self.firstname.text()
        user = self.cr_username.text()
        passw = self.cr_password.text()
        cont = self.contact_line.text()
        if self.type_admin.isChecked():
            self.cur.execute('''INSERT INTO admins
(name,username,password)
VALUES (%s,%s,%s);
''', (nam,user,passw))
            self.db.commit()
            self.statusBar().showMessage('Account Created')

        if self.type_user.isChecked():
            self.cur.execute('''INSERT INTO users
(name,username,password,contact)
VALUES (%s,%s,%s,%s);
''', (nam,user,passw,cont))
            self.db.commit()
            self.statusBar().showMessage('Account Created')

        self.firstname.setText('')
        self.cr_username.setText('')
        self.cr_password.setText('')
        self.contact_line.setText('')
```

```

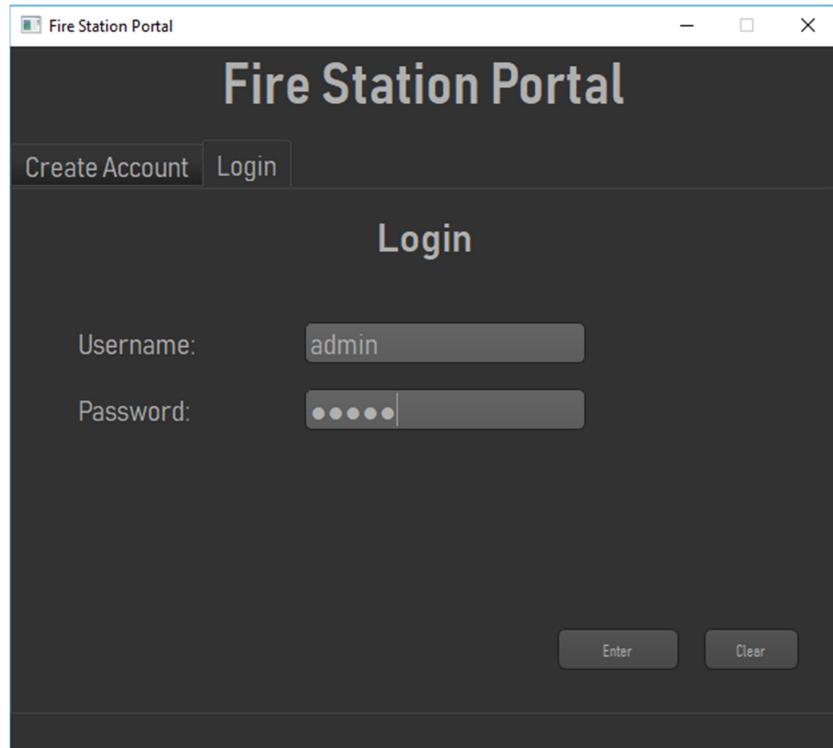
def create_clr(self):
    self.firstname.setText('')
    self.cr_username.setText('')
    self.cr_password.setText('')
    self.contact_line.setText('')
def login(self):
    self.db = pymysql.connect(host='localhost', user= 'root',
password= 'admin', db='firestation')
    self.cur = self.db.cursor()

    user = self.log_username.text()
    password = self.log_password.text()

    self.cur.execute('''SELECT * FROM admins WHERE username=%s
AND password=%s;''', (user,password))
    data = self.cur.fetchone()
    if data != None:
        self.adminpass()
    else:
        self.cur.execute('''SELECT * FROM users WHERE username=%s
AND password=%s;''', (user,password))
        self.data = self.cur.fetchone()
        if self.data != None:
            print('found')
            self.userpass()
        else:
            self.label_11.setText('Make sure you entered your
username and password correctly')
def login_clr(self):
    self.log_username.setText('')
    self.log_password.setText('')

```

Admin Login Page



 Fire Station Portal

[Create Account](#) [Login](#)

Create Account

Name:

Username:

Password:

Contact No.:

Account type: Admin User

[Create Account](#) [Clear](#)

 Fire Station Database





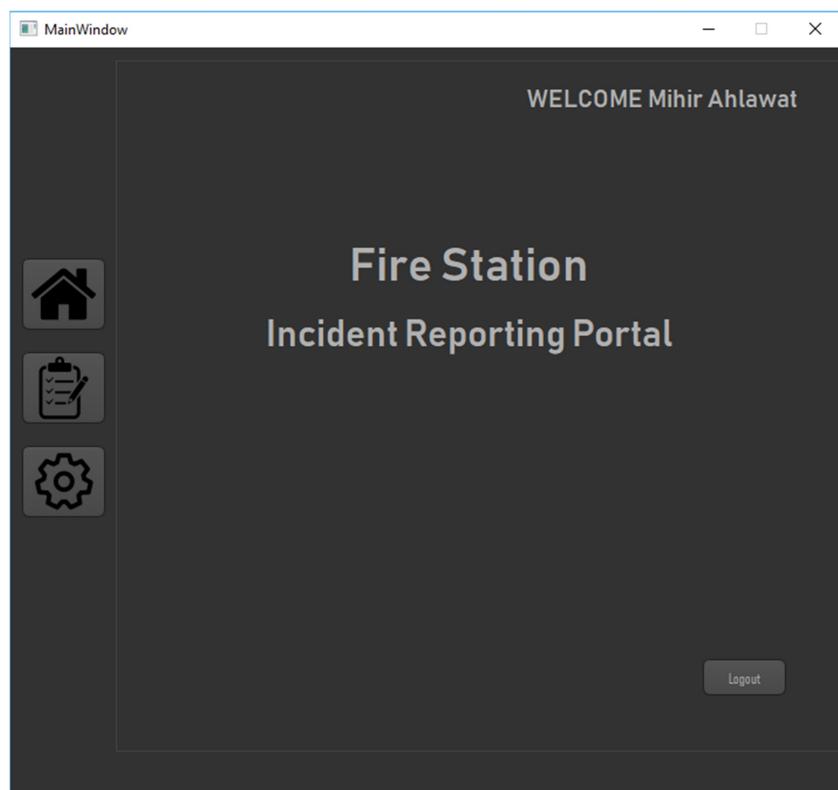
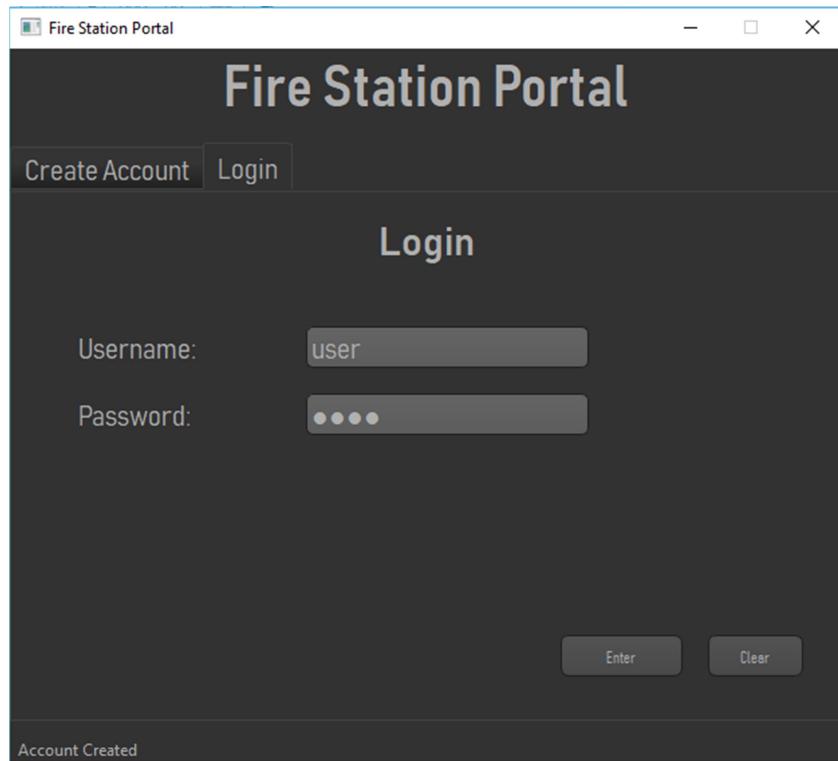



Fire Station Database Management System

Please follow the tabs to manage the database.

[Logout](#)

User Login Page

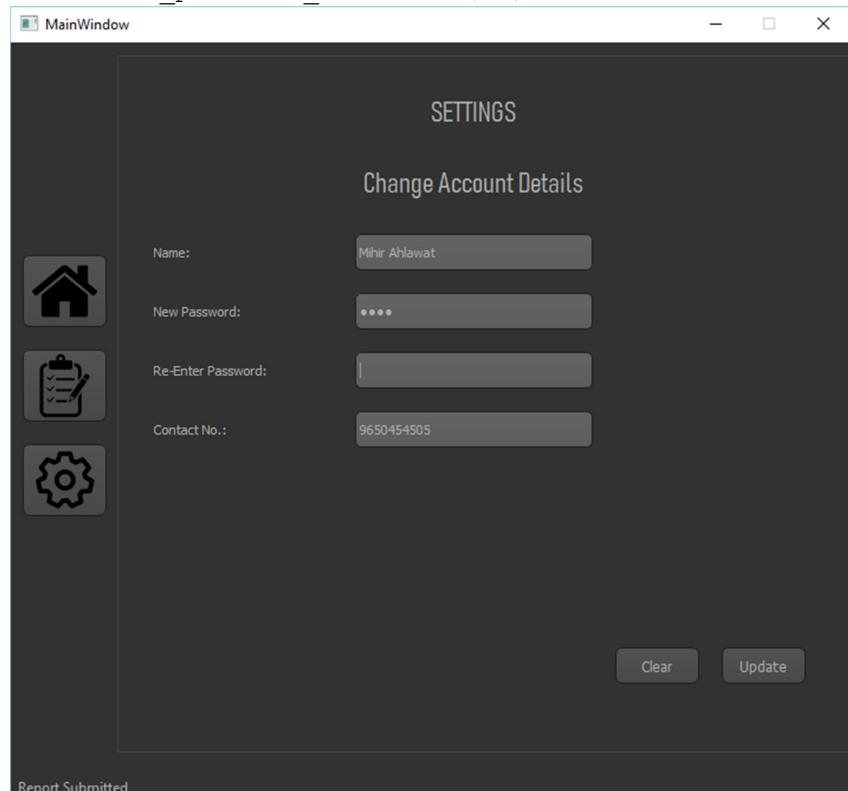


2. Change User Account settings

```
##### User Settings #####
def update_user(self):
    self.label_12.setText('')
    self.db = pymysql.connect(host='localhost', user= 'root',
password= 'admin', db='firestation')
    self.cur = self.db.cursor()
    nam = self.firstname.text()
    passw = self.cr_password.text()
    passw2 = self.cr_password_2.text()
    cont = self.contact_line.text()
    if passw == '' or passw2 == '':
        self.label_12.setText('Passwords cannot be empty')
    else:
        if passw == passw2:
            self.cur.execute(''' UPDATE users SET
name=%s,password=%s,contact=%s
WHERE username=%s;
'', (nam,passw,cont,self.user[1]))
            self.db.commit()
            self.statusBar().showMessage('Details
Updated')
        else:
            self.label_12.setText('Passwords do not
match')

        self.cr_password.setText('')
        self.cr_password_2.setText('')

def update_clr(self):
    self.cr_password.setText('')
    self.cr_password_2.setText('')
```



3. Add a fire report

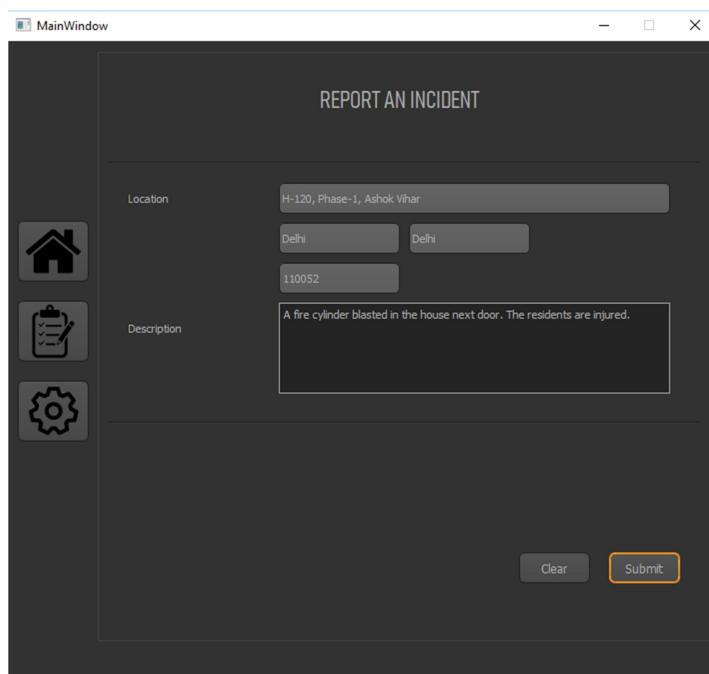
```
##### Report #####
def add_report(self):
    self.db = pymysql.connect(host='localhost', user= 'root',
    password= 'admin', db='firestation')
    self.cur = self.db.cursor()

    street = self.lineEdit_22.text()
    city = self.lineEdit_20.text()
    state = self.lineEdit_23.text()
    pin = self.lineEdit_21.text()
    descrip = self.textEdit.toPlainText()
    date = time.strftime('%Y-%m-%d %H:%M:%S')

    self.cur.execute('''INSERT INTO reports
(STREET_ADDR,CITY,STATE,PINCODE,DESCRIP,INFO_NAME,INFO_NUM,DATE_TI
ME)
VALUES (%s,%s,%s,%s,%s,%s,%s,%s);

''', (street,city,state,pin,descrip,self.user[0],self.user[3],date)
)
    self.db.commit()
    self.statusBar().showMessage('Report Submitted')
    self.lineEdit_22.setText('')
    self.lineEdit_20.setText('')
    self.lineEdit_23.setText('')
    self.lineEdit_21.setText('')
    self.textEdit.setPlainText('')

def add_clr(self):
    self.lineEdit_22.setText('')
    self.lineEdit_20.setText('')
    self.lineEdit_23.setText('')
    self.lineEdit_21.setText('')
    self.textEdit.setPlainText('')
```



4. Modify Vehicle Data

```
##### Vehicles #####
    def add_vehicle(self):
        self.db = pymysql.connect(host='localhost', user= 'root',
password= 'admin', db='firestation')
        self.cur = self.db.cursor()

        vehicle_number = self.lineEdit_5.text()
        station_id = self.comboBox_9.currentText()
        vehicle_type = self.comboBox.currentText()
        model_number = self.lineEdit_6.text()
        status = self.comboBox_5.currentText()
        water_cap = self.lineEdit_10.text()
        purchase = self.lineEdit_7.text()

        self.cur.execute('''INSERT INTO vehicles
(VEHICLE_NUM, VEHICLE_STATION, VEHICLE_TYPE, MODEL_NO, VEHICLE_STATUS,
WATER_CAP, PURCHASE)
VALUES (%s,%s,%s,%s,%s,%s);

''', (vehicle_number,station_id,vehicle_type,model_number,status,wa
ter_cap,purchase))
        self.db.commit()
        self.statusBar().showMessage('New Vehicle Added')

        self.lineEdit_5.setText('')
        self.comboBox_9.setCurrentIndex(0)
        self.comboBox.setCurrentIndex(0)
        self.lineEdit_6.setText('')
        self.comboBox_5.setCurrentIndex(0)
        self.lineEdit_10.setText('')
        self.lineEdit_7.setText('')

    def search_vehicle(self):
        self.db = pymysql.connect(host='localhost', user= 'root',
password= 'admin', db='firestation')
        self.cur = self.db.cursor()
        id = self.lineEdit_12.text()

        sql = ''' SELECT * FROM vehicles WHERE VEHICLE_ID =
%s;'''
        self.cur.execute(sql, [(id)])
        data = self.cur.fetchone()

        if data != None:
            self.lineEdit_26.setText(data[1])
            self.comboBox_8.setCurrentText(data[2])
            self.comboBox_3.setCurrentText(data[3])
            self.lineEdit_18.setText(data[4])
            self.comboBox_6.setCurrentText(data[5])
            self.lineEdit_11.setText(data[6])
            self.lineEdit_8.setText(data[7])
        else:
            QMessageBox.critical(self,'Error','Please Enter A
Valid Vehicle ID')
```

```

    def update_vehicle(self):
        self.db = pymysql.connect(host='localhost', user= 'root',
password= 'admin', db='firestation')
        self.cur = self.db.cursor()

        vehicle_id = self.lineEdit_12.text()
        vehicle_number = self.lineEdit_26.text()
        station_id = self.comboBox_8.currentText()
        vehicle_type = self.comboBox_3.currentText()
        model_number = self.lineEdit_18.text()
        status = self.comboBox_6.currentText()
        water_cap = self.lineEdit_11.text()
        purchase = self.lineEdit_8.text()

        self.cur.execute('''UPDATE vehicles SET VEHICLE_NUM =
%s, VEHICLE_STATION = %s, VEHICLE_TYPE = %s, MODEL_NO =
%s, VEHICLE_STATUS = %s, WATER_CAP = %s, PURCHASE = %s
WHERE VEHICLE_ID = %s;

''', (vehicle_number,station_id,vehicle_type,model_number,status,wa-
ter_cap,purchase,vehicle_id))
        self.db.commit()
        self.statusBar().showMessage('Vehicle Data Updated')

        self.lineEdit_12.setText('')
        self.lineEdit_26.setText('')
        self.comboBox_8.setCurrentIndex(0)
        self.comboBox_3.setCurrentIndex(0)
        self.lineEdit_18.setText('')
        self.comboBox_6.setCurrentIndex(0)
        self.lineEdit_11.setText('')
        self.lineEdit_8.setText('')

    def delete_vehicle(self):
        self.db = pymysql.connect(host='localhost', user= 'root',
password= 'admin', db='firestation')
        self.cur = self.db.cursor()
        id = self.lineEdit_12.text()

        warning = QMessageBox.warning(self, 'Delete Vehicle', 'Are
you sure you want to delete this vehicle?', QMessageBox.Yes | |
QMessageBox.No)
        if warning == QMessageBox.Yes :
            sql = ''' DELETE FROM vehicles WHERE VEHICLE_ID = %s;'''
            self.cur.execute(sql, [(id)])
            self.db.commit()
            self.statusBar().showMessage('Vehicle Deleted')

```

VEHICLES

Add Vehicle Edit/Delete Vehicle Details All Vehicles

	Vehicle Number	DL-8C-3884
	Associated Station ID	10001
	Vehicle Type	Fire Brigade
	Model Number	38xc4
	Status	In Use
	Water Capacity(if applicable)	4000
	Purchase Date	18-11-2015

New Vehicle Type Added

VEHICLES

Add Vehicle Edit/Delete Vehicle Details All Vehicles

Enter Vehicle ID

	Vehicle Number	DL-8X-8290
	Associated Station ID	10005
	Vehicle Type	Fire Brigade
	Model Number	38xc4
	Status	<ul style="list-style-type: none">In RepairSelectIn UseIn RepairNot in use
	Water Capacity(if applicable)	
	Purchase Date	

New Vehicle Added

VEHICLES

Vehicle ID	Vehicle No.	Station ID	Vehicle Type	Model No.	Status
1 70	DL-8C-3884	10001	Fire Brigade	38xc4	In Use
2 71	DL-5S-9366	10002	Fire Brigade	38xc4	In Use
3 72	DL-6C-1220	10003	Fire Brigade	38xc4	In Use
4 73	DL-8Q-9842	10004	Fire Brigade	38xc4	In Use
5 74	DL-8X-8290	10005	Fire Brigade	38xc4	Not in use
6 75	DL-8C-1120	10005	Fire Brigade	38xc4	In Use
7 76	X-QWER	10001	Helicopter	F210 R/C	In Use
8 77	DL-8Y-0662	10001	Light Ambulance	Maruti Swift	In Use

Vehicle Data Updated

5. Modify Staff Data

Similar code as vehicle updating

STAFF

Add Staff	Edit/Delete Staff Details	All Staff	
	Rajyavardhan	Singh	Rathore
	Gender	<input checked="" type="radio"/> Male	<input type="radio"/> Female
	Date of Birth	25-07-1968	
	Contact No.	(+91) 9876543210	
	Current Address	Keshav Puram, Delhi	
	Designation	Station Head	Date of Joining 01-02-1996
	Date of leaving(Default if not applicable)	-	
	Associated Station ID	10001	Salary (per annum) 15 Lac
		<input type="button" value="Clear"/>	<input type="button" value="Submit"/>

Vehicle Data Updated

Fire Station Database

STAFF

Add Staff | Edit/Delete Staff Details | All Staff

Enter Identity No.

Rakesh Kumar

Gender Male Female

Date of Birth

Contact No. (+91)

Current Address

Designation Date of Joining

Date of leaving(Default if not applicable)

Associated Station ID Salary (per annum)

Fire Station Database

STAFF

Add Staff | Edit/Delete Staff Details | All Staff

	Last Name	Gender	Date of Birth	Contact No.	Address	Design
1	Rathore	Male	25-07-1965	9876543210	Keshav Puram, De...	Station He...
2	Bhagwan	Male	20-08-1974	9818803321	Ashok Vihar, De...	Station He...
3	Singh	Male	04-01-1968	9012345678	Rohini, Delhi	Station He...
4	Kumar	Male	25-12-1986	9784561230	Pitampura, Delhi	Driver Eng...
5	Chaurasia	Male	05-01-1992	9869786920	Rohini, Delhi	Firefighter
6	Ahlawat	Male	18-11-1997	9650454505	Ashok Vihar, De...	Intern

6. Reports Data

```
##### Only viewing the list of reports is allowed #####
##### Reports #####
    def load_report(self):
        self.db = pymysql.connect(host='localhost', user= 'root',
password= 'admin', db='firestation')
        self.cur = self.db.cursor()

        self.cur.execute('' SELECT * FROM reports '')
        data = self.cur.fetchall()

        self.tableWidget_3.setRowCount(0)

        for row, form in enumerate(data):
            row_pos = self.tableWidget_3.rowCount()
            self.tableWidget_3.insertRow(row_pos)

            for col, item in enumerate(form):
                self.tableWidget_3.setItem(row, col,
QTableWidgetItem(str(item)))
                col += 1

    def export_report(self):
        self.db = pymysql.connect(host='localhost', user= 'root',
password= 'admin', db='firestation')
        self.cur = self.db.cursor()

        self.cur.execute('' SELECT * FROM reports ')
        data = self.cur.fetchall()

        wb = Workbook('Reports.xlsx')
        sheet1 = wb.add_worksheet()

        sheet1.write(0,0,'Report ID')
        sheet1.write(0,1,'Street Address')
        sheet1.write(0,2,'City')
        sheet1.write(0,3,'State')
        sheet1.write(0,4,'Pincode')
        sheet1.write(0,5,'Description')
        sheet1.write(0,6,'Informant Name')
        sheet1.write(0,7,'Informant Contact')
        sheet1.write(0,8,'Date and Time')

        row_num = 1
        for row in data:
            col_num = 0
            for item in row:
                sheet1.write(row_num, col_num, str(item))
                col_num += 1
            row_num += 1

        wb.close()
        self.statusBar().showMessage('Reports Exported
Successfully')
```

Fire Station Database

INCIDENTS REPORTED

Street address	City	State	Pincode	Description
1 H-120, Phase-1,...	Delhi	Delhi	110052	A fire cylinder blasted in the h...
2 Plot No-1, Kew...	Delhi	Delhi	110033	A fire broke out in a nearby m...
3 B/32, Harihar A...	Delhi	Delhi	110034	Help me!! Save me!! Im dying...
4 B/32, Sector-16,...	Delhi	Delhi	110089	Someone set a heap of trash on...

Load Export data to excel file

7. Modify Action on Reports Data

Same as Vehicle updating

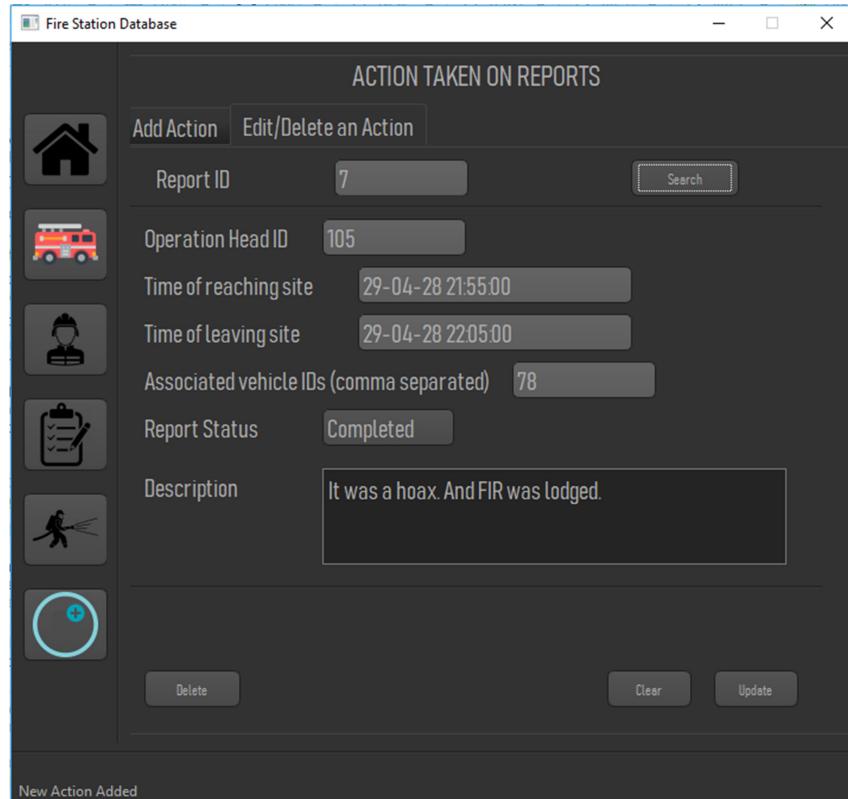
Fire Station Database

ACTION TAKEN ON REPORTS

Add Action Edit/Delete an Action

Report ID	5
Operation Head ID	105
Time of reaching site	19-04-28 22:10:00
Time of leaving site	19-04-29 01:30:00
Associated vehicle IDs (comma separated)	73
Report Status	Completed
Description	The injured were taken to Ambedkar hospital, Rohini.

Clear Submit



8. Update data for drop down menus

```

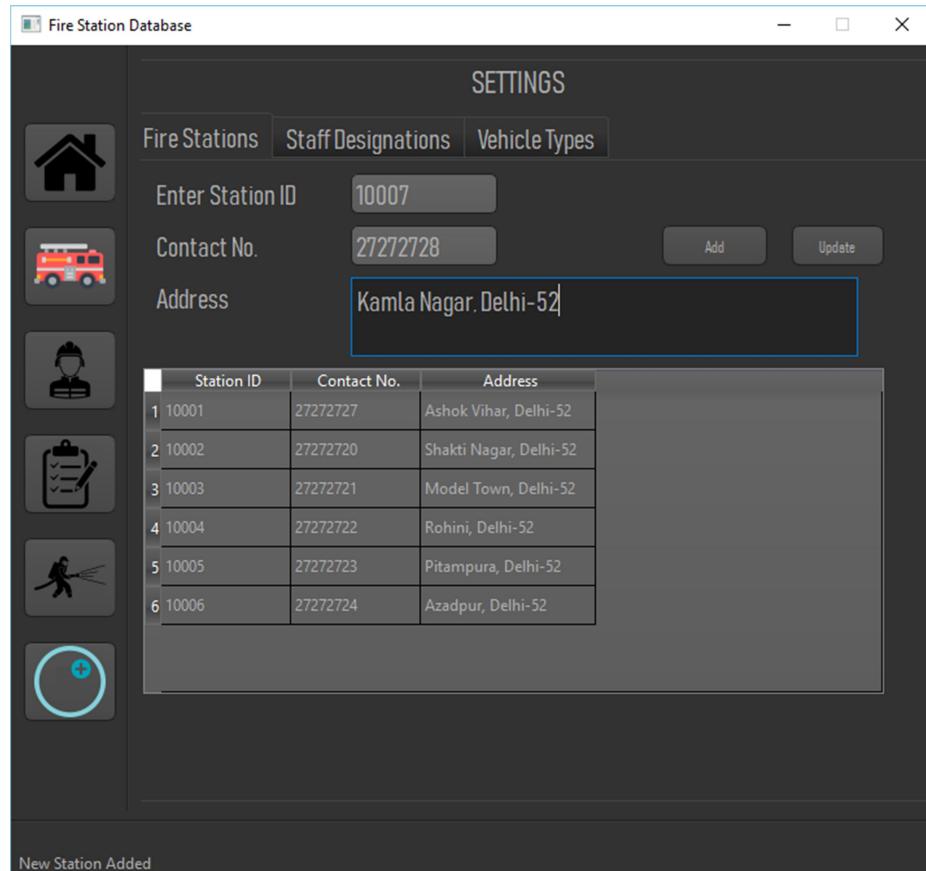
#####
# Settings #####
## Add fire stations ##
def add_station(self):
    self.db = pymysql.connect(host='localhost', user= 'root',
    password= 'admin', db='firestation')
    self.cur = self.db.cursor()

    station_id = self.lineEdit_27.text()
    station_num = self.lineEdit_29.text()
    station_addr = self.plainTextEdit_5.toPlainText()

    self.cur.execute('''
        INSERT INTO stations
    (STATION_ID,STATION_CONTACT,STATION_ADDR) VALUES (%s,%s,%s)
    ''', (station_id,station_num,station_addr) )

    self.db.commit()
    self.lineEdit_27.setText('')
    self.lineEdit_29.setText('')
    self.plainTextEdit_5.setPlainText('')
    self.show_station()
    self.statusBar().showMessage('New Station Added')

```

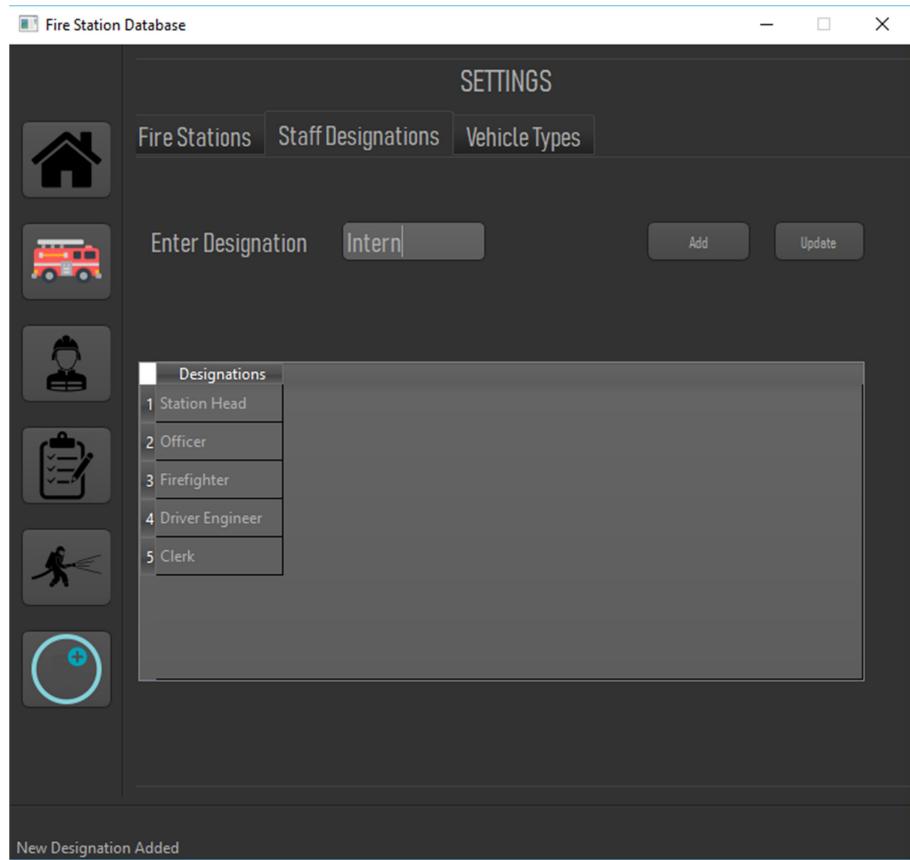


```
## Add designations ##
def add_designation(self):
    self.db = pymysql.connect(host='localhost', user= 'root',
    password= 'admin', db='firestation')
    self.cur = self.db.cursor()

    desig = self.lineEdit_30.text()

    self.cur.execute('''
        INSERT INTO designation (DESIG_NAME) VALUES (%s)
    ''', (desig, ) )

    self.db.commit()
    self.lineEdit_30.setText('')
    self.show_designation()
    self.statusBar().showMessage('New Designation Added')
```



```
## Add vehicle type ##
def add_vehicle_type(self):
    self.db = pymysql.connect(host='localhost', user= 'root',
password= 'admin', db='firestation')
    self.cur = self.db.cursor()

    veh_type = self.lineEdit_31.text()

    self.cur.execute('''
        INSERT INTO vehicle_type (TYPE_NAME) VALUES (%s)
    ''', (veh_type,) )

    self.db.commit()
    self.lineEdit_31.setText('')
    self.show_vehicle_type()
    self.statusBar().showMessage('New Vehicle Type Added')
```

