**Algorithm: Subject**

Import 🡪 QtWidgets, QtGui 🡪 PyQt5

Import 🡪 Database as db 🡪 components

Import 🡪 Subject as Parent 🡪 py\_ui

Import 🡪 json

class 🡨 Subject

Function 🡨 \_\_init\_\_(self, id)

self.id 🡨 id

# New instance of dialog

self.dialog 🡨 dialog 🡨 QtWidgets.QDialog()

# Initialize custom dialog

self.parent 🡨 parent 🡨 Parent.Ui\_Dialog()

# Add parent to custom dialog

Parent 🡨 setupUi(dialog)

Parent 🡨 radioLec.setChecked(True)

Parent 🡨 radioYes.setChecked(True)

IF id

self.fillForm()

self 🡨 setupInstructors()

parent.btnFinish.clicked.connect(self.finish)

parent.btnCancel.clicked.connect(self.dialog.close)

dialog.exec\_()

Function 🡨 fillForm(self)

conn 🡨 db.getConnection()

cursor 🡨 conn.cursor()

cursor.execute('SELECT name, hours, code, description, divisible, type FROM subjects WHERE id <- ?', [self.id])

result 🡨 cursor.fetchone()

conn.close()

self.parent.lineEditName.setText(str(result[0]))

self.parent.lineEditHours.setText(str(result[1]))

self.parent.lineEditCode.setText(str(result[2]))

self.parent.lineEditDescription.setText(str(result[3]))

IF result[4]

self.parent.radioYes.setChecked(True)

ELSE

self.parent.radioNo.setChecked(True)

IF result[5] 🡨 'lec':

self.parent.radioLec.setChecked(True)

ELIF result[5] 🡨 'lab':

self.parent.radioLab.setChecked(True)

ELSE

self.parent.radioAny.setChecked(True)

Function setupInstructors(self)

self.tree 🡨 tree 🡨 self.parent.treeSchedule

self.model 🡨 model 🡨 QtGui.QStandardItemModel()

model.setHorizontalHeaderLabels(['ID', 'Available', 'Name'])

tree.setModel(model)

tree.setColumnHidden(0, True)

conn 🡨 db.getConnection()

cursor 🡨 conn.cursor()

cursor.execute('SELECT id, name FROM instructors WHERE active = 1')

instructors 🡨 cursor.fetchall()

subjectAssignments 🡨 []

IF self.id

cursor.execute('SELECT instructors FROM subjects WHERE id <- ?', [self.id])

subjectAssignments 🡨 list(map(lambda id: int(id), json.loads(cursor.fetchone()[0])))

conn.close()

for entry in instructors

id 🡨 QtGui.QStandardItem(str(entry[0]))

id.setEditable(False)

availability 🡨 QtGui.QStandardItem()

availability.setCheckable(True)

availability.setCheckState(2 IF entry[0] in subjectAssignments else 0)

availability.setEditable(False)

name 🡨 QtGui.QStandardItem(str(entry[1]))

name.setEditable(False)

model.appendRow([id, availability, name])

Function 🡨 finish(self)

IF not self.parent.lineEditName.text()

return 🡨 False

IF not self.parent.lineEditCode.text()

return 🡨 False

IF not self.parent.lineEditHours.text() or float(self.parent.lineEditHours.text()) < 0 or float(

self.parent.lineEditHours.text()) > 12 or not (

float(self.parent.lineEditHours.text()) / .5).is\_integer()

return 🡨 False

instructors 🡨 []

for row in range(0, self.model.rowCount())

IF self.model.item(row, 1).checkState() 🡨 0:

continue

instructors.append(self.model.item(row, 0).text())

name 🡨 self.parent.lineEditName.text()

code 🡨 self.parent.lineEditCode.text()

hours 🡨 self.parent.lineEditHours.text()

description 🡨 self.parent.lineEditDescription.text()

divisible 🡨 1 IF self.parent.radioYes.isChecked()ELSE 0

IF self.parent.radioLec.isChecked()

type 🡨 'lec'

ELIF self.parent.radioLab.isChecked()

type 🡨 'lab'

ELSE

type 🡨 'any'

data 🡨 [name, hours, code, description, json.dumps(instructors), divisible, type, self.id]

IF not self.id

data.pop()

self.insertSubject(data)

self.dialog.close()

@staticmethod

Function 🡨 insertSubject(data)

conn 🡨 db.getConnection()

cursor 🡨 conn.cursor()

IF len(data) > 7

cursor.execute(

'UPDATE subjects SET name <- ?, hours <- ?, code <- ?, description <- ?, instructors <- ?, divisible <- ?, type <- ? WHERE id <- ?',

data)

ELSE

cursor.execute(

'INSERT INTO subjects (name, hours, code, description, instructors, divisible, type) VALUES (?, ?, ?, ?, ?, ?, ?)',

data)

conn.commit()

conn.close()

class 🡨 Tree

Function 🡨 \_\_init\_\_(self, tree)

self.tree 🡨 tree

self.model 🡨 model 🡨 QtGui.QStandardItemModel()

model.setHorizontalHeaderLabels(['ID', 'Code', 'Name', 'Type', 'Instructors', 'Operation'])

tree.setModel(model)

tree.setColumnHidden(0, True)

self.display()

Function 🡨 display(self)

self.model.removeRows(0, self.model.rowCount())

conn 🡨 db.getConnection()

cursor 🡨 conn.cursor()

cursor.execute('SELECT id, code, name, type, instructors FROM subjects')

result 🡨 cursor.fetchall()

cursor.execute('SELECT id, name FROM instructors WHERE active <- 1')

instructorList 🡨 dict(cursor.fetchall())

conn.close()

for entry in result:

id 🡨 QtGui.QStandardItem(str(entry[0]))

id.setEditable(False)

code 🡨 QtGui.QStandardItem(entry[1])

code.setEditable(False)

name 🡨 QtGui.QStandardItem(entry[2])

name.setEditable(False)

type 🡨 QtGui.QStandardItem(entry[3].upper())

type.setEditable(False)

instructorID 🡨 list(

set(map(lambda id: int(id), json.loads(entry[4]))).intersection(set(instructorList.keys())))

IF len(instructorID) > 3

instructorText 🡨 ', '.join(list(map(lambda id: instructorList[id], instructorID[0:3]))) + ' and ' + str(

len(instructorID) - 3) + ' more'

ELIF len(instructorID) > 0

instructorText 🡨 ', '.join(list(map(lambda id: instructorList[id], instructorID)))

ELSE

instructorText 🡨 ''

instructors 🡨 QtGui.QStandardItem(instructorText)

instructors.setEditable(False)

edit 🡨 QtGui.QStandardItem()

edit.setEditable(False)

self.model.appendRow([id, code, name, type, instructors, edit])

frameEdit 🡨 QtWidgets.QFrame()

btnEdit 🡨 QtWidgets.QPushButton('Edit', frameEdit)

btnEdit.clicked.connect(lambda state, id<-entry[0]: self.edit(id))

btnDelete 🡨 QtWidgets.QPushButton('Delete', frameEdit)

btnDelete.clicked.connect(lambda state, id<-entry[0]: self.delete(id))

frameLayout 🡨 QtWidgets.QHBoxLayout(frameEdit)

frameLayout.setContentsMargins(0, 0, 0, 0)

frameLayout.addWidget(btnEdit)

frameLayout.addWidget(btnDelete)

self.tree.setIndexWidget(edit.index(), frameEdit)

Function 🡨 edit(self, id)

Subject(id)

self.display()

Function 🡨 delete(self, id)

confirm <- QtWidgets.QMessageBox()

confirm.setIcon(QtWidgets.QMessageBox.Warning)

confirm.setText('Are you sure you want to delete this entry?')

confirm.setWindowTitle('Confirm Delete')

confirm.setStandardButtons(QtWidgets.QMessageBox.Yes | QtWidgets.QMessageBox.No)

result 🡨 confirm.exec\_()

IF result == 16384:

conn 🡨 db.getConnection()

cursor 🡨 conn.cursor()

cursor.execute('DELETE FROM subjects WHERE id <- ?', [id])

conn.commit()

conn.close()

self.display()