Design:

Step1: import the class and functions from the bokeh

from bokeh.io import output\_file, show

from bokeh.layouts import row, column, layout

from bokeh.plotting import figure, curdoc

from bokeh.layouts import widgetbox

from bokeh.models import ColumnDataSource

from bokeh.models.widgets import Button, RadioButtonGroup, Select, Slider, TextInput

from bokeh.models.widgets import RadioGroup, CheckboxGroup, MultiSelect, Dropdown

from bokeh.models.widgets import DataTable, DateFormatter, TableColumn, Panel, Tabs

from bokeh.models.widgets import Paragraph

from bokeh.events import ButtonClick

from datetime import date

from random import randint

import string

import pyodbc

import functools

from functools import partial

from bokeh.core.properties import value, Instance

Step2: try to connect the data base by using SQL methods:

obdcConnection = None

def connectSQLServer():

global obdcConnection

attr = dict(

server = '10.20.213.10',

database = 'csc1002',

username = 'csc1002',

password = 'csc1002',

port = 1433,

driver = 'ODBC Driver 13 for SQL Server'

)

conn\_str = 'DRIVER={driver};' \

'SERVER={server};' \

'PORT={port};' \

'DATABASE={database};' \

'UID={username};' \

'PWD={password}'

conn\_str = conn\_str.format(\*\*attr)

try:

return pyodbc.connect(conn\_str)

except Exception as e:

print(e)

quit()

Step4: using the data from the database lgu.course to build a dictionary named data to prepared for the creation of the column:

obdcConnection = connectSQLServer()

tsql = "SELECT \* FROM lgu.course;"

cursor = obdcConnection.cursor()

with obdcConnection:

rows = cursor.execute(tsql).fetchall()

id\_list=list()

title\_list=list()

dept\_name\_list=list()

credit\_list=list()

instructor\_list=list()

for row in rows:

id\_list.append(row.course\_id)

title\_list.append(row.title)

dept\_name\_list.append(row.dept\_name)

credit\_list.append(row.credits)

instructor\_list.append(row.instructor)

data=dict(

id=id\_list,

title=title\_list,

dept=dept\_name\_list,

credit=credit\_list,

instructor=instructor\_list

)

Step5: using the dept\_name\_list to create a new dept\_name list which only contains the name of the departname and there’s no repetition, which is prepared for the Select Button:

new\_dept\_list=list()

for i in dept\_name\_list:

if not i in new\_dept\_list:

new\_dept\_list.append(i)

Step6: Build all the buttons:

refresh = Button(label="Refresh")

btnGroupTitle = RadioButtonGroup(name='title', labels=["begins with...", "...contains...", "...ends with"], active=1)

btnGroupDept = RadioButtonGroup(name='dept', labels=["begins with...", "...contains...", "...ends with"], active=1)

paragraph = Paragraph(text="option")

optionGroup = RadioGroup(labels=["and", "or"], active=0, width=100, inline=True)

btnGroupLetters = RadioButtonGroup(labels=list(string.ascii\_uppercase), active=-1)

title\_input = TextInput(value="", title="Title:", placeholder="contains....")

dept\_input = TextInput(value="", title="Department:", placeholder="contains....")

dept\_select = Select(title='Department',value='',options=new\_dept\_list)

Step7: build the column, and it use the data from the dictionary which is named data by us in the former step:

columns = [

TableColumn(field='id',title='Course ID'),

TableColumn(field='title',title='Title'),

TableColumn(field='dept',title='Department'),

TableColumn(field='credit',title='Credit'),

TableColumn(field='instructor',title='Instructor')

]

table\_master=DataTable(source=ColumnDataSource(),

columns=columns,width=800,height=200)

table\_master.source.data=data

Step8: build a Statistical chart using the data from lgu.student. The default data is based on the default selection of the select button:

gpa=['A+','A','B+','B','C+','C','D+','D','F']

years=['2015','2016','2017']

colors=['#c9d9d3','#718dbf','#e84d60']

data={}

data['gpa']=gpa

data['2015']=[1, 4, 2, 4, 1, 6, 0, 2, 1]

data['2016']=[5, 2, 5, 8, 3, 10, 0, 1, 1]

data['2017']=[2, 4, 3, 12, 7, 12, 0, 2, 1]

source= ColumnDataSource(data=data)

p=figure(x\_range=gpa,plot\_height=500, title='GPA Counts by Year', toolbar\_location=None,tools='')

p.vbar\_stack(years,x='gpa', width=0.9, color=colors, source=source,legend=[value(x) for x in years])

p.y\_range.start=0

p.y\_range.end=30

Step10: define a onclick function. On the first tabs, there are 3 kind of buttons we need to consider. First is the letter button group, second is the refresh button, third is the button group like ‘contains…’, ‘begin with…’,’end with…’.

Letter button group and the refresh button can be put into a single group, because their core thoughts are the same, that is select the exact data from the original data base. Therefore, the if statement can put them into the same frame, and the if statement only controls the different ‘value ’ of the tsql

In details, for the letter button group, we want the data which the couse title is begin with the letter. Therefore, we can use this statement:

tsql="SELECT \* FROM lgu.course where title like '"+letter+"%'"

for the refresh button, the data is depend on the results of 5 things, btnGroupTitle, btnGroupDept, title\_input, dept\_input, optionGroup. The situation is kind of complicated. So we use the following code to discuss it:

elif name=='refresh':

cursor = obdcConnection.cursor()

if btnGroupTitle.active==0 and btnGroupDept.active==0:

if optionGroup.active==0:

tsql = "SELECT \* FROM lgu.course where title like '"+title\_input.value+"%' and dept\_name like '"+dept\_input.value+"%'"

elif optionGroup.active==1:

tsql = "SELECT \* FROM lgu.course where title like '"+title\_input.value+"%' or dept\_name like '"+dept\_input.value+"%'"

elif btnGroupTitle.active==0 and btnGroupDept.active==1:

if optionGroup.active==0:

tsql = "SELECT \* FROM lgu.course where title like '"+title\_input.value+"%' and dept\_name like '%"+dept\_input.value+"%'"

elif optionGroup.active==1:

tsql = "SELECT \* FROM lgu.course where title like '"+title\_input.value+"%' or dept\_name like '%"+dept\_input.value+"%'"

elif btnGroupTitle.active==0 and btnGroupDept.active==2:

if optionGroup.active==0:

tsql = "SELECT \* FROM lgu.course where title like '"+title\_input.value+"%' and dept\_name like '%"+dept\_input.value+"'"

elif optionGroup.active==1:

tsql = "SELECT \* FROM lgu.course where title like '"+title\_input.value+"%' or dept\_name like '%"+dept\_input.value+"'"

elif btnGroupTitle.active==1 and btnGroupDept.active==0:

if optionGroup.active==0:

tsql = "SELECT \* FROM lgu.course where title like '&"+title\_input.value+"%' and dept\_name like '"+dept\_input.value+"%'"

elif optionGroup.active==1:

tsql = "SELECT \* FROM lgu.course where title like '&"+title\_input.value+"%' or dept\_name like '"+dept\_input.value+"%'"

elif btnGroupTitle.active==1 and btnGroupDept.active==1:

if optionGroup.active==0:

tsql = "SELECT \* FROM lgu.course where title like '%"+title\_input.value+"%' and dept\_name like '%"+dept\_input.value+"%'"

elif optionGroup.active==1:

tsql = "SELECT \* FROM lgu.course where title like '%"+title\_input.value+"%' or dept\_name like '%"+dept\_input.value+"%'"

elif btnGroupTitle.active==1 and btnGroupDept.active==2:

if optionGroup.active==0:

tsql = "SELECT \* FROM lgu.course where title like '%"+title\_input.value+"%' and dept\_name like '%"+dept\_input.value+"'"

elif optionGroup.active==1:

tsql = "SELECT \* FROM lgu.course where title like '%"+title\_input.value+"%' or dept\_name like '%"+dept\_input.value+"'"

elif btnGroupTitle.active==2 and btnGroupDept.active==0:

if optionGroup.active==0:

tsql = "SELECT \* FROM lgu.course where title like '&"+title\_input.value+"' and dept\_name like '"+dept\_input.value+"%'"

elif optionGroup.active==1:

tsql = "SELECT \* FROM lgu.course where title like '&"+title\_input.value+"' or dept\_name like '"+dept\_input.value+"%'"

elif btnGroupTitle.active==2 and btnGroupDept.active==1:

if optionGroup.active==0:

tsql = "SELECT \* FROM lgu.course where title like '%"+title\_input.value+"' and dept\_name like '%"+dept\_input.value+"%'"

elif optionGroup.active==1:

tsql = "SELECT \* FROM lgu.course where title like '%"+title\_input.value+"' or dept\_name like '%"+dept\_input.value+"%'"

elif btnGroupTitle.active==2 and btnGroupDept.active==2:

if optionGroup.active==0:

tsql = "SELECT \* FROM lgu.course where title like '%"+title\_input.value+"' and dept\_name like '%"+dept\_input.value+"'"

elif optionGroup.active==1:

tsql = "SELECT \* FROM lgu.course where title like '%"+title\_input.value+"' or dept\_name like '%"+dept\_input.value+"'"

Then we use the following code to change the dictionary data which is used for the column:

with obdcConnection:

rows = cursor.execute(tsql).fetchall()

id\_list=list()

title\_list=list()

dept\_name\_list=list()

credit\_list=list()

instructor\_list=list()

for row in rows:

id\_list.append(row.course\_id)

title\_list.append(row.title)

dept\_name\_list.append(row.dept\_name)

credit\_list.append(row.credits)

instructor\_list.append(row.instructor)

data=dict(

id=id\_list,

title=title\_list,

dept=dept\_name\_list,

credit=credit\_list,

instructor=instructor\_list

)

table\_master.source.data=data

At last, we can deal with the btnGroupTitle, and btnGroupDept, it is very easy that we change the placeholder of the corresponding title\_input and dept\_input according to the place of the button we clicked in the button group:

elif name=='Title':

if btnGroupTitle.active==0:

title\_input.placeholder='begins with...'

elif btnGroupTitle.active==1:

title\_input.placeholder='...contains...'

elif btnGroupTitle.active==2:

title\_input.placeholder='...ends with'

elif name=='Dept':

if btnGroupDept.active==0:

dept\_input.placeholder='begins with...'

elif btnGroupDept.active==1:

dept\_input.placeholder='...contains...'

elif btnGroupDept.active==2:

dept\_input.placeholder='...ends with'

Step11: define onTableMasterSelect function in order to change the stastics chart according to the different selection in the dept\_select button.

Firstly, we select the data from lgu.student.

cursor = obdcConnection.cursor()

tsql = "SELECT \* FROM lgu.student"

Then we create three lists which contains the information of each line in the original database corresponded one by one:

with obdcConnection:

rows = cursor.execute(tsql).fetchall()

gpa\_list=list()

years\_list=list()

dept\_student\_list=list()

for row in rows:

gpa\_list.append(row.gpa)

years\_list.append(row.year)

dept\_student\_list.append(row.dept\_name)

Then we need to count the number of the students that meet the 3 requirements for gpa, course title and years, and the information will be stored in the three lists. The format of the 3 lists meet the requirement of the original dictionary data:

list\_2015\_new=list()

list\_2016\_new=list()

list\_2017\_new=list()

gpa=['A+','A','B+','B','C+','C','D+','D','F']

for i in gpa:

count\_part=0

count\_total=0

for j in gpa\_list:

if j==i and years\_list[count\_total]=='2015' and dept\_student\_list[count\_total]==new:

count\_part=count\_part+1

count\_total=count\_total+1

list\_2015\_new.append(count\_part)

count\_part=0

count\_total=0

for j in gpa\_list:

if j==i and years\_list[count\_total]=='2016' and dept\_student\_list[count\_total]==new:

count\_part=count\_part+1

count\_total=count\_total+1

list\_2016\_new.append(count\_part)

count\_part=0

count\_total=0

for j in gpa\_list:

if j==i and years\_list[count\_total]=='2017' and dept\_student\_list[count\_total]==new:

count\_part=count\_part+1

count\_total=count\_total+1

list\_2017\_new.append(count\_part)

At last, we use the new dictionary data to replace the original one:

data={}

data['gpa']=gpa

data['2015']=list\_2015\_new

data['2016']=list\_2016\_new

data['2017']=list\_2017\_new

source.data=data

Step12: apply functions on the corresponding buttons:

btnGroupLetters.on\_click(functools.partial(onclick,name='Letters'))

refresh.on\_click(functools.partial(onclick,name='refresh'))

btnGroupTitle.on\_click(functools.partial(onclick,name='Title'))

btnGroupDept.on\_click(functools.partial(onclick,name='Dept'))

dept\_select.on\_change('value',onTableMasterSelect)

Step13: lay out all the visual parts:

layout\_query = layout(

[

[widgetbox(btnGroupLetters, width=1000)],

[widgetbox(btnGroupTitle), widgetbox(btnGroupDept)],

[widgetbox(title\_input), widgetbox(paragraph, optionGroup, width=100), widgetbox(dept\_input)],

[widgetbox(refresh, width=100)],

[widgetbox(table\_master)],

]

)

layout\_chart = layout(

[

[widgetbox(dept\_select),p]

]

)

tab1 = Panel(child=layout\_query, title="Course Info")

tab2= Panel(child=layout\_chart, title='Statistics')

tabs = Tabs(tabs=[tab1, tab2])

curdoc().add\_root(tabs)

Test:

1. For each buttons and button group, we should test one by one to check whether the function is realized well.
2. In total, there 3 main parts need us to test carfully:

The letter group, we should check them one by one.

The refresh button, we should check all the possible situations. In total, there are 18 situations need us to check.

The stastics chart, we need to check all the situations in the select button.

1. Put the ui file in different computers, web browsers to check the situations again.