# Data Visualization – Part I (Interactive Query)

OVERVIEW

In this assignment, you are asked to design and develop a Data Visualization application to provide school administrators and students with a web-enabled interface to perform online query on school information, such information as course description, student GPA, course location, instructor name and so on; based on the responses from the users, the application then generates a SQL statement to fetch the data needed; based on the data returned from SQL server, the screen will be refreshed with the requested information shown in a tabular format or as graphical plot. The application will be composed of 3 primary components: Graphical User Interface (GUI), SQL Interface and Graphical Tool depicted as follows:
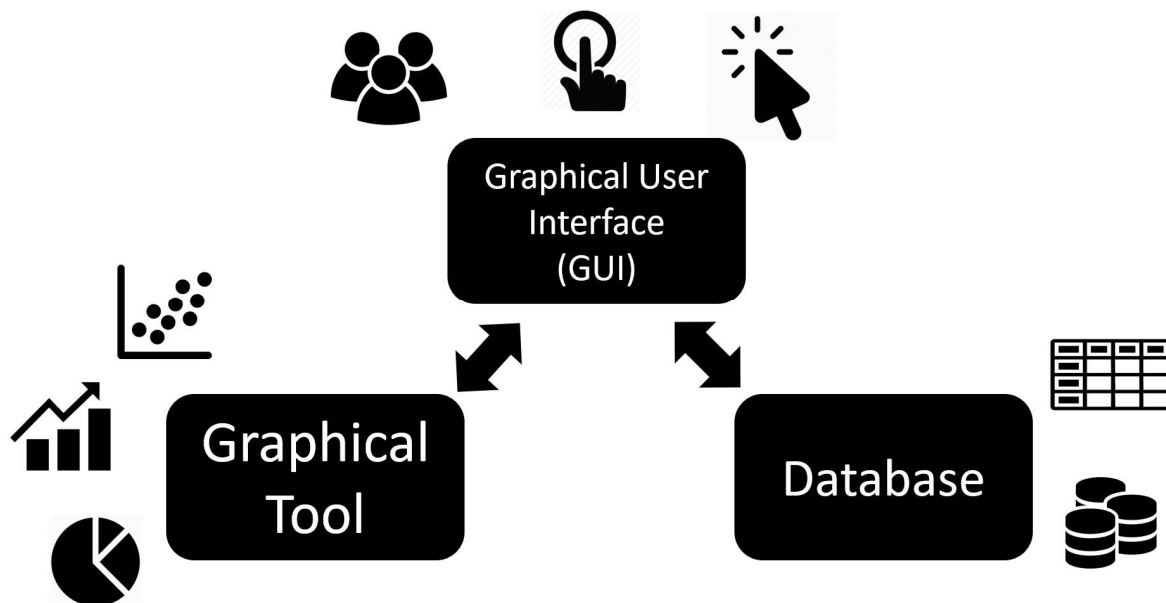
The GUI component will be composed of 2 web pages, one for online course query and the other for statistical information on student GPA distribution. The format of the query page is shown as follows:

Course Info	Statistics

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |

| begins with... | ...contains... | ...ends with | | begins with... | ...contains... | ...ends with |

Title:

contains....

option

◉ and ○ or

Department:

contains....

Refresh

#	Course ID	Title	Department	Credit	Instructor
0	843	Environmental Law	Math	4	Lembr
1	362	Embedded Systems	Finance	4	Mingoz
2	362	Embedded Systems	Finance	4	Mingoz
3	362	Embedded Systems	Finance	4	Mingoz
4	158	Elastic Structures	Cybernetics	3	Bietzk
5	158	Elastic Structures	Cybernetics	3	Dale

Whereas the format of the statistics page is shown as follows:

Course Info	Statistics

Department:

Elec. Eng.

GPA Counts by Year

2015
2016
2017
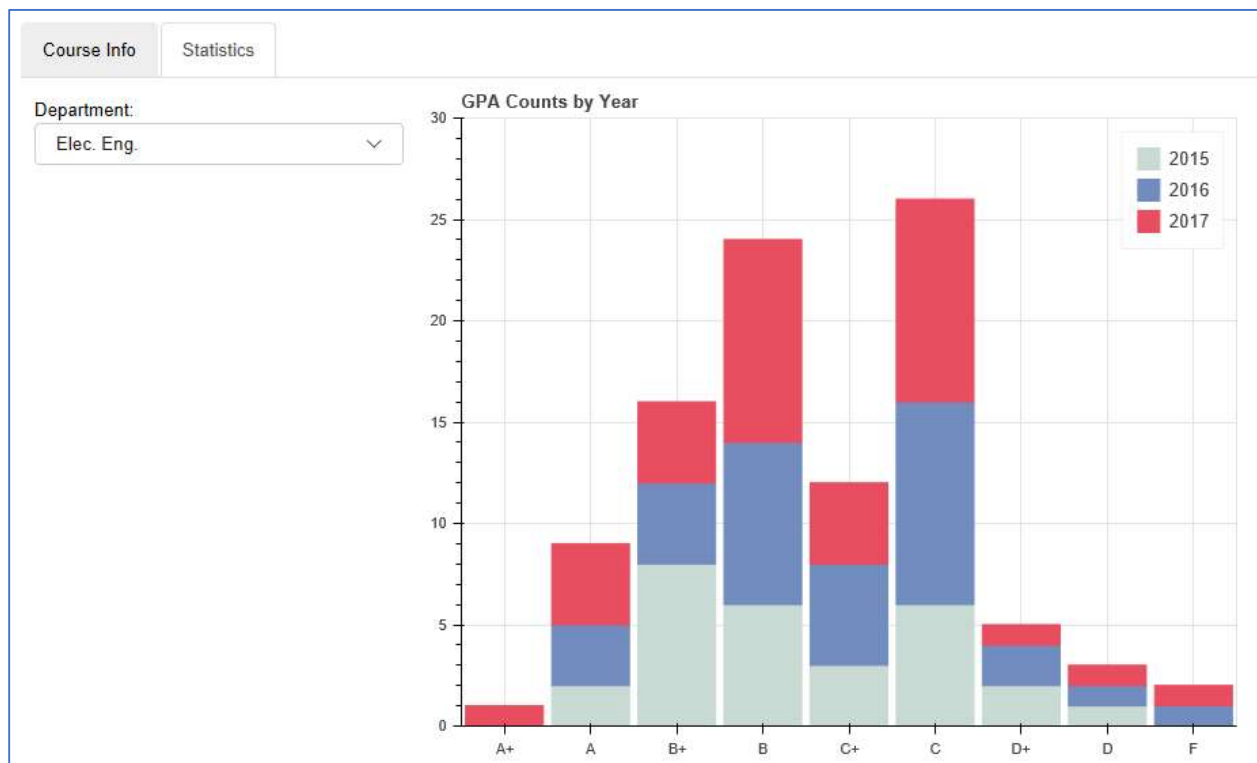
SPECIAL SKILLS

- Data Visualization: using "Bokeh" to construct the web-based GUI to interface with the end users for online query
- Relationship Database: constructing simple SQL statements to extract the relevant information from a database.
- Using data visualization tool to turn raw data into interactive 2-D graphics such as lines, scatter plots, bar graphs, stacked bars and so on.
- Using ODBC to interface the SQL server
- Using widgets to capture user responses: text input, radio buttons, button groups, tables, dropdown list and so on.

S<small>COPE</small> – T<small>AB</small> P<small>ANEL</small>
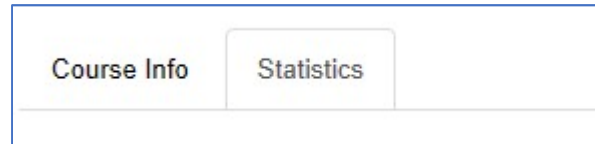
1. Using bokeh tool to construct a multi-tab panel for the home page:
   a. First panel as "Course Info"
   b. Second panel as "Statistics"
2. The panel "Course Info" holds the online course query page, refer to corresponding section for more details.
3. The panel "Statistics" holds the statistics page, refer to corresponding section for more details.

SCOPE – COURSE INFO

4. Using bokeh widgets & layout to construct the web-based interactive query screen.
   a. refer to the given template file (ui.py) for information.
5. The A-Z button group, when clicked, will display all courses with title beginning with the letter of the clicked button, the result will be displayed in the data table shown on the bottom of the screen.
6. Two sets of "begin with …."-"…contains…"-"…ends with" button-groups, one for the course title and the other for department name. They behave as radio buttons. They are used together with the two corresponding text inputs, namely "Title" and "Department" respectively.
   a. These 3 buttons indicate whether or not the given string will be used as prefix, infix or suffix match respectively. If "begins with…" is selected, courses will be matched with title beginning with the given text; if "…contains…" is selected, title containing the given text will be matched.
   b. Ex: "begin with…" with input "bio", courses such as "biology", "bioinformatics" are shown.
7. The "and"-"or" radio button-group indicates whether or not to search on title and department as a group ("and") or separately ("or").
8. When the button "Refresh" is clicked, the actual search will take place and the result will be displayed in the data table. The search is based on a database table called "lgu.course".
   a. sp_help "lgu.course" to view its structure (refer to session "Database – Info")
9. It is required that the program execute another SQL statement to fetch the whatever the data requested. It is NOT supposed to cache the entire course table in memory and avoid further database request. Though the size of the course table for this exercise is small in size, however, in reality, it can be potentially large in size; it's therefore best practice to use SQL query to fetch only the data needed.

Examples:

- "begins with…" with "bio" as Title

- "…contains…" with "bio" as Title

- "…ends with" with "bio" as Title

- "…contains…" with "bio" as Title and "finance" as Department, with "and" option

#	Course ID	Title	Department	Credit	Instructor
0	445	Biostatistics	Finance	3	Mingoz

- "…contains…" with "bio" as Title and "finance" as Department, with "or" option

#	Course ID	Title	Department	Credit	Instructor
0	366	Computational Biology	English	3	Kean
1	137	Manufacturing	Finance	3	Mingoz
2	304	Music 2 New for your Instr…	Finance	4	Mingoz
3	319	World History	Finance	4	Mingoz
4	349	Networking	Finance	4	Mingoz
5	362	Embedded Systems	Finance	4	Mingoz

DATABASE – INFO

Using Visual Studio Code, create a file of type "sql", make a database connection to "csc1002" and perform a query execution with the following statement:

- sp_help "lgu.course"
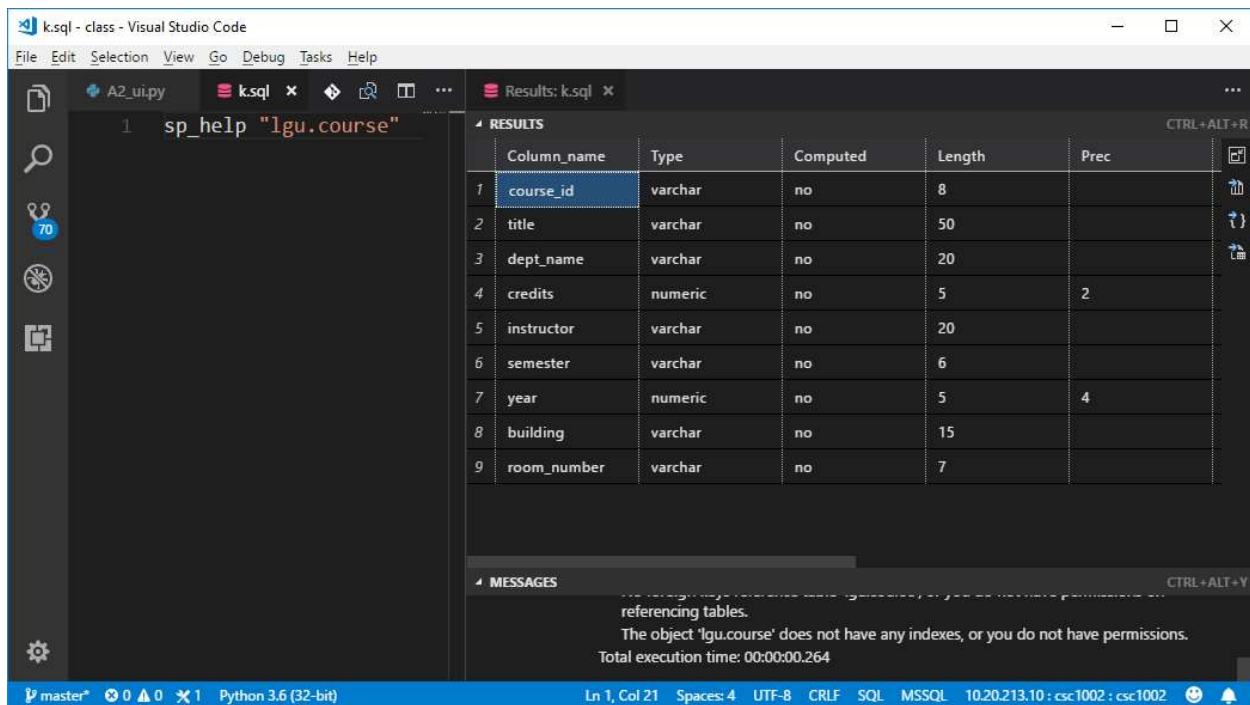
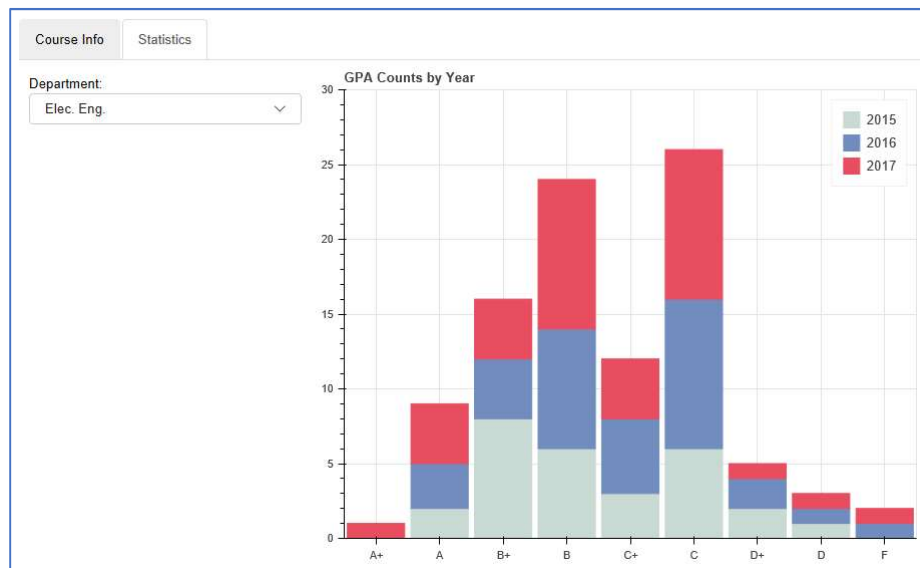- ODBC connection string:

```
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)
```

SCOPE – STATISTICS

1. Using bokeh widgets & layout to construct the web-based interactive query screen.
   a. refer to the lab materials for more information
2. On the right side of the page is a stacked bar chart, showing the GPA distribution by department, further categorized by class year.
3. On the left side of the page is a selection list displaying the list of all departments
   a. It is used to select the department for which the chart is created
4. When a new department is selected, the chart will be updated to refresh the corresponding information.
5. The stacked bar chart shows student GPA distribution based on the selected department.
   a. Students are from class of 2015, 2016 & 2017 (year)
   b. As illustrated, GPA counts are stacked in chronological order, each with a unique color
   c. Legend (upper right) is shown to display the color of the classes
6. The database table to use is "lgu.student" and the structure is shown as follows:

STARTUP OPTIONS

"bokeh serve --show <a2.py>"

SKILLS - PYTHON

In this assignment, you will be trained on the use of the followings:

Standard Python Object:

- Use standard objects (strings, numbers & lists)
- Control statements to interact with users
- Variable Scope
- String formatting (method style)
- Functions for program structure and decomposition

DELIVERABLES

1. Design documentation (A2_School_StudentID_Design.doc/pdf)
2. Program source code (A2_School_StudentID_Source.py)
3. Output (A2_School_StudentID_Output.doc/pdf)

Zip all files above in a single file (A2_School_StudentID.zip) and submit the zip file by due date to the corresponding assignment folder under "Assignment (submission)"

For instances, a SME student with student ID "119010001":

- A2_SME_119010001.zip:
  - A2_SME_119010001_Design.doc/pdf
  - A2_SME_119010001_Source.py
  - A2_SME_119010001_Output.doc/pdf

5% will be deducted if any files are incorrectly named!!!

DESIGN DOCUMENTATION

For the design document provide write-up for the following information:

1. Design:
    a. Describe the general structure of the program (functions, variables and program flow).
2. Test Plan:
    a. Refer to assignment 1 for information.

TIPS & HINTS

- For executing and debugging "bokeh", refer to the uploaded file "launch.json" and "tasks.json".
- For general information on "Bokeh", refer to the following link:
    https://bokeh.pydata.org/en/latest/docs/user_guide.html
- For adding widgets (text input, list box, button groups, tables ..etc), refer to the following link:
    https://bokeh.pydata.org/en/latest/docs/user_guide/interaction/widgets.html
- For reference guide, refer to the following link:
    https://bokeh.pydata.org/en/latest/docs/reference.html

DEBUGGING
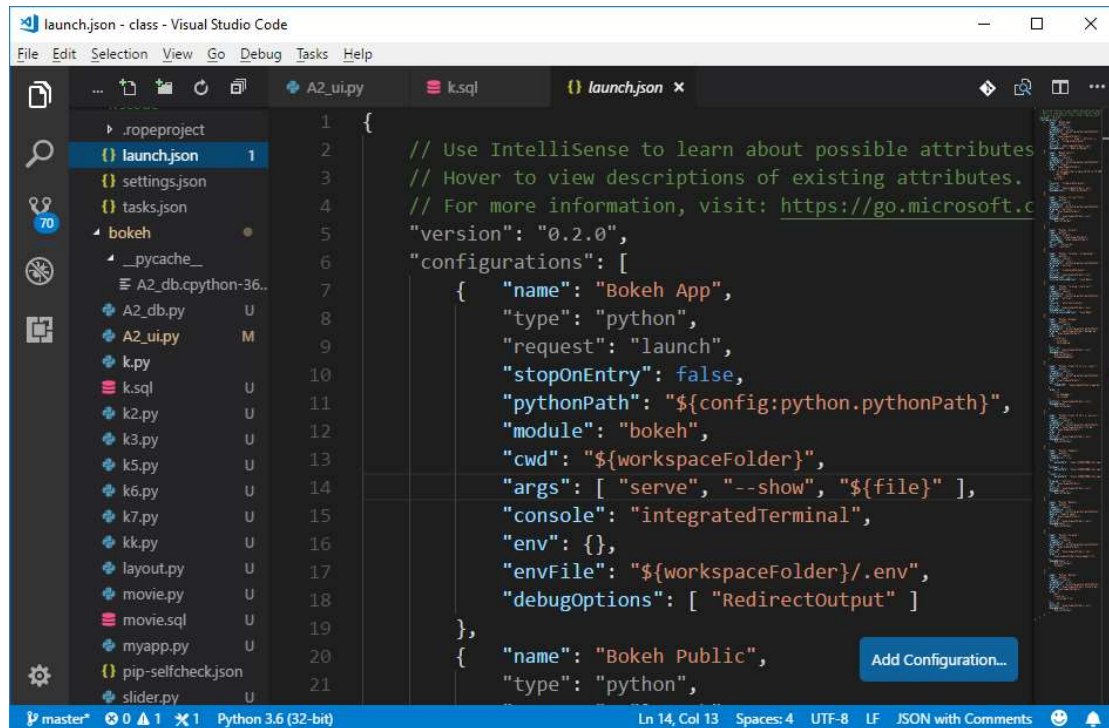
VSC TASKS
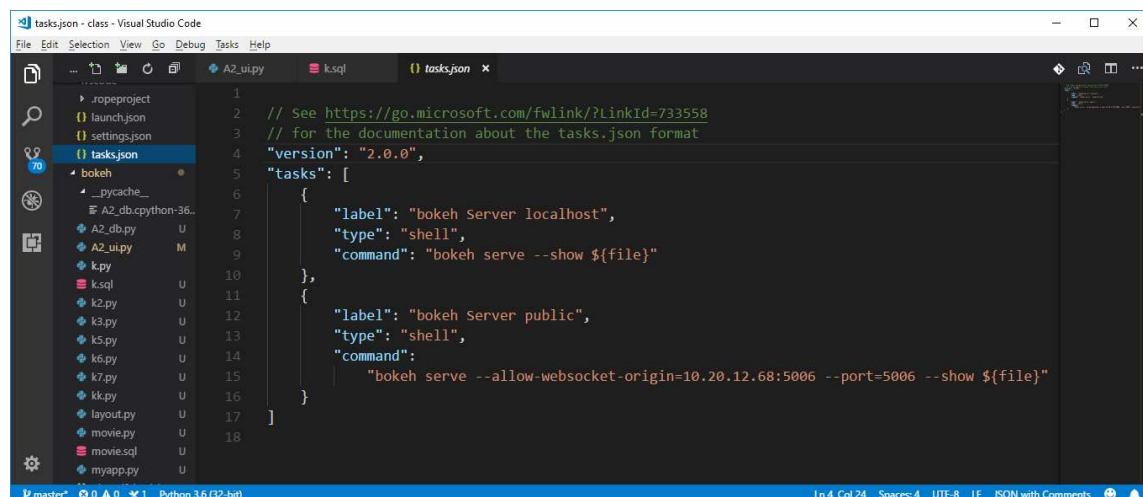
Note: replace IP (20.20.12.68) with your own local IP. 5006 is the http port, do not need to change it.

SAMPLE OUTPUT

Course Info	Statistics

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

begins with...	...contains...	...ends with		begins with...	...contains...	...ends with

Title:
`ics`

option
⦿ and ○ or

Department:
`Finance`

[Refresh]

#	Course ID	Title	Department	Credit	Instructor
0	445	Biostatistics	Finance	3	Mingoz
1	527	Graphics	Finance	3	Mingoz

Course Info	Statistics

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

begins with...	...contains...	...ends with		begins with...	...contains...	...ends with

Title:
`contains....`

option
⦿ and ○ or

Department:
`contains....`

[Refresh]

#	Course ID	Title	Department	Credit	Instructor
0	338	Graph Theory	Psychology	3	DAgostino
1	338	Graph Theory	Psychology	3	DAgostino
2	663	Geology	Psychology	3	DAgostino
3	972	Greek Tragedy	Psychology	4	DAgostino
4	527	Graphics	Finance	3	Mingoz
5	735	Greek Tragedy	Geology	3	Mahmoud
6	735	Greek Tragedy	Geology	3	Mahmoud

Course Info | Statistics

Department:
Geology

GPA Counts by Year

Course Info | Statistics

Department:
Math

GPA Counts by Year
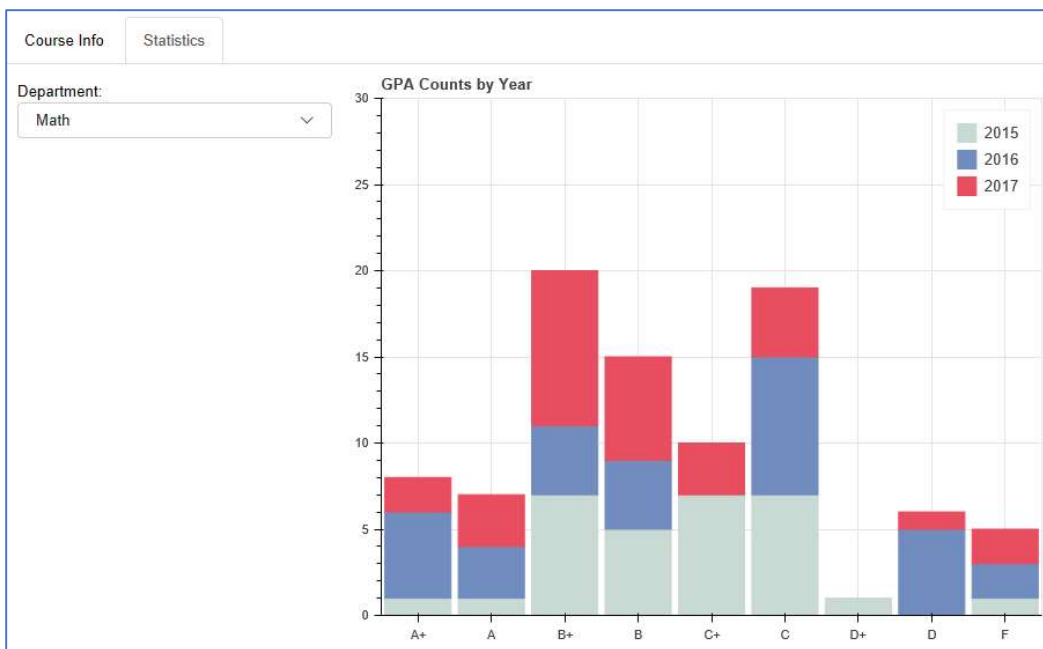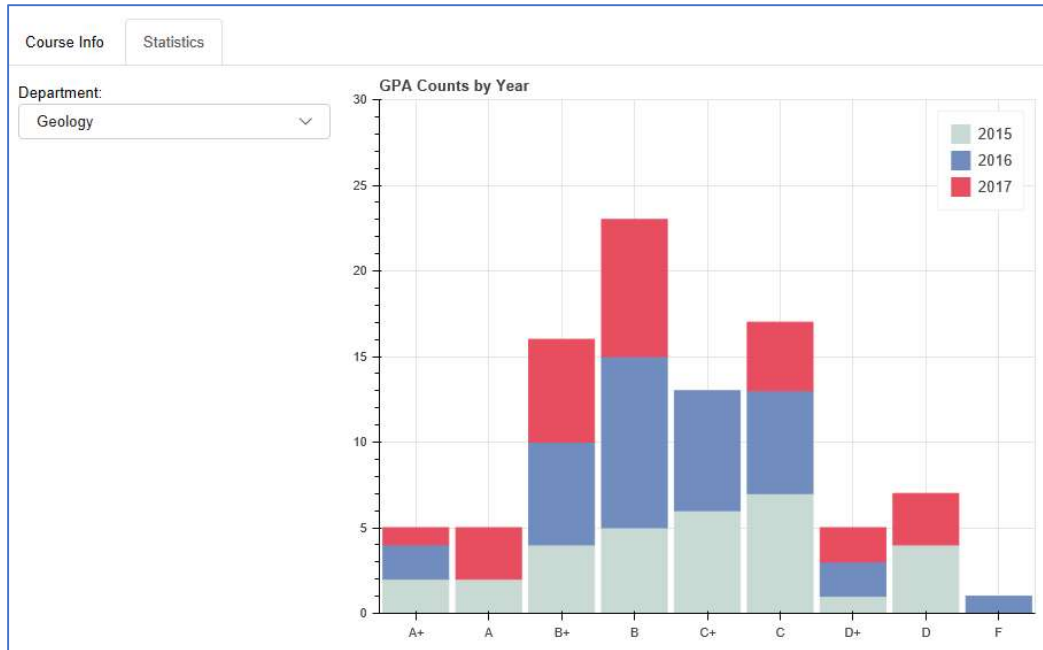
MARKING CRITERIA

- Coding Styles – layout, comments, white spaces, naming convention, variables, indentation.
- Documentation – Design + Test Plan
- Program Correctness – logic, program structure, functions with appropriate parameters
- User Interaction – how informative and accurate information is exchanged between game player and host.
- Readability counts – programs that are well structured and easy-to-follow using functions to breakdown complex problems into smaller cleaner generalized functions are preferred over a function embracing a complex logic with nested conditions and sub-functions! In other words, a design with clean architecture with high readability is the predilection for the course objectives over efficiency.
- KISS approach – Keep It Simple and Straightforward.
- Balance approach – you are not required to come up a very optimized solution. However, take a balance between readability and efficiency with good use of program constructs.

CHALLENGES

Completed by due date

DUE DATE

April 20th, 2018