Question

Although there are many places for self-study on campus, many students choose to go to libraries, especially during the revision week. Therefore, when deciding where to do self-study, students probably wish to know the situations in differents libraries and in different sections of one particular library (whether it is open, how many people are there, which library is the nearest, etc), and may also want some advice.

Proposed Solution

This bot is designed to address the above-mentioned problem.

The algorithm is explained in pseudocode, for the first half, and in natural language, for the second half, because it is really difficult to draw a flowchart to cover everything and we would also have struggled with the page layout if we chose to do so.

Pseudocode for the first part

```
procedure BotFirstPart
read message
lib:={"“LWN Lib”, “HSS Lib”, “BIZ Lib”, “ADM Lib”, “CHN Lib”}
lib_p:={“LWN lib”, “HSS lib”, “BIZ lib”, “ADM lib”, “CHN lib”}
if the message is not sent during lib hours:
        if the message is sent on Sunday:
                print "It's Sunday dude, all libraries are closed.\nGo relax and have
                fun!\nヽ(✿゜▽゜)ノ"
        else if the message is sent during 0am to 8am:
                print "So hard-working dude! Libraries are not opened yet.\n╮('～`〃 )╭"
        else:
                Print "Oops, all libraries have closed.\n( ˘･з･)"
else:
label1:
        if the message is text:
                if the button “Library Current Status Inquiry” is clicked:
                        print “Please Choose Your Library”
                        display keyboard buttons “Nearest Lib”, “LWN Lib”, “BIZ Lib”,
                        “HSS Lib”, “CHN Lib”, “ADM Lib”
                        if the button “Nearest Lib” is clicked:
                                ask for the coordinate of the user
                                goto label2
                        goto label1
                else if the button “Library Status Prediction” is clicked:
                        print “Where do you want to go?”
                        display keyboard buttons “LWN lib”, “BIZ lib”, “HSS lib”, “CHN
                        lib”, “ADM lib”
                        goto label1
```

```
else if the clicked button has the same text as one of the members of
lib:
        n:=the index of the found string in lib
        Print "Please Choose Your Floor and Section"
        if n=0:
                display keyboard buttons "5F Quiet Zone", "4F", "3F", "2F"
                buff:="lwn"
        else if n=1:
                display keyboard buttons "Inside", "Outside"
                buff:="hss"
        else if n=2:
                display keyboard buttons "B4 Quiet Zone", "B3", "B2&B1"
                buff:="biz"
        else if n=3:
                display keyboard buttons "Section A", "Section B"
                buff:="adm"
        else:
                display keyboard buttons "Section A", "Section B", "Section
                C"
                buff:="chn"
else if the clicked button has the same text as one of the members of
lib_p:
        n:=the index of the found string in lib_p
        Print "Please Choose Your Floor and Section"
        if n=0:
                display keyboard buttons "5F Quiet Zone", "4F", "3F", "2F"
                buff:="lwnp"
        else if n=1:
                display keyboard buttons "Inside", "Outside"
                buff:="hssp"
        else if n=2:
                display keyboard buttons "B4 Quiet Zone", "B3", "B2&B1"
                buff:="bizp"
        else if n=3:
                display keyboard buttons "Section A", "Section B"
                buff:="admp"
        else:
                display keyboard buttons "Section A", "Section B", "Section
                C"
                buff:="chnp"
else:
        Print "Welcome to NTU library assistant,\nwhat can I do for
        you?"
        display keyboard buttons "Library Current Status Inquiry",
        "Library Status Prediction"
        goto label1
```

```
label2:
if the message is location:
        n:=0
        dis:=9999999
        if (x-1.347757)^2+(y-103.680900)^2<dis:
                dis:=(x-1.347757)^2+(y-103.680900)^2
                n:=0
        if (x-1.344308)^2+(y-103.682256)^2<dis:
                dis:=(x-1.344308)^2+(y-103.682256)^2
                n:=1
        if (x-1.346526)^2+(y-103.680051)^2<dis:
                dis:=(x-1.346526)^2+(y-103.680051)^2
                n:=2
        if (x-1.349541^)2+(y-103.683809)^2<dis:
                dis:=(x-1.349541^)2+(y-103.683809)^2<dis
                n:=3
        if (x-1.343871)^2+(y-103.682369)^2<dis:
                dis:=(x-1.343871)^2+(y-103.682369)^2
                n:=4
        Print "Please Choose Your Floor and Section"
                if n=0:
                        display keyboard buttons "5F Quiet Zone", "4F", "3F", "2F"
                        buff:="lwn"
                else if n=1:
                        display keyboard buttons "Inside", "Outside"
                        buff:="hss"
                else if n=2:
                        display keyboard buttons "B4 Quiet Zone", "B3", "B2&B1"
                        buff:="biz"
                else if n=3:
                        display keyboard buttons "Section A", "Section B"
                        buff:="adm"
                else:
                        display keyboard buttons "Section A", "Section B", "Section
                        C"
                        buff:="chn"
```

Explanation for the second part

For the next part, we believe it would be more clear to be explained in words rather than pseudocodes and we also have not got all the data we want yet. Firstly, the algorithm checks the length of the string buff and we would get 3 or 4 in different situations. 3 indicates the algorithm should return the current status in the request session, whereas 4 requires it to estimate the number of people may be there in the next hour. For the current status, we would have a constant array storing the number of seats in each section of every library. The algorithm counts all the devices using

school network service and takes this as an estimation of people studying in that particular section. In this step, the algorithm also pushes the number and the time to the database. The algorithm subtracts the estimated number of occupied seats from the total seat number ,output the result, and give different suggestions when the result is less than 10 or not less than 10. For the estimation part, the algorithm goes back to the database, finds out the last p sets of data (p is a number designated by the developers and currently it is 12), extract the numbers from these data (the times will be ignored by the algorithm), and computes the average of them. This average number would be the estimated number output to the user as a reference. Besides all these, the algorithm would also update its database every 5 minutes.

Constraints and limitations

1. The opening time of the libraries differs in semester period and vacation period. Also, some of them may have extended hours. The program does not take the above items into consideration.

2. While finding the nearest lirary, straightaway distance between the two positions is computed, which can sometimes be useless in some real life cases.

3. In the estimation function, the program takes the number of devices connected with the school network service as that of people studying in that particular section. This step can cause deviation, since someone there may not use the Internet.

Conclusions

This bot provides its users with useful information about the libraries in NTU, specifically the opening time, the exact location, the current and past-time status of selectable sections. Additionally, users can get the predicted number of seats available in whichever place they want efficiently. So this program will be a great convenience for those who wants to arrange his self-study better.