NANYANG
TECHNOLOGICAL
UNIVERSITY

Report on Industrial Orientation with SingAREN

Prepared by: Tong Zi Hang

U1221544D

School of Computer Engineering (Computer Science)

Contents

Abstract

The Singapore Advanced Research and Education Networks (SingAren) aims to establish effective network connections between local research institutes and the international community. Initially a government-funded initiative, SingAren has grown to build collaborations with local universities and government entities in promoting ground-breaking innovation in the education sector.

The student's involvement in the Eduroam statistics project has allowed the student to obtain valuable programming experience with Python and JavaScript through the development of the log processing program and the data visualization tool. It allowed the student to apply his knowledge about Object-Oriented Programming and Software Engineering concepts.

The log processing program was redesigned to minimize manual interference and the data visualization tool was operational. One recommendation was made to design the data visualization to fit the dashboard user interface allowing easy viewing for the users.

Acknowledgement

The Industrial Orientation was made possible with the support and cooperation of the following people, who has enabled the student to gain more knowledge beyond the industrial aspect of the attachment. The student therefore expresses his gratitude to the following people:

- Dr Lee Bu Sung, Vice-President of SingAren and the IO supervisor of the student, for providing the opportunity to work with the organization and offering guidance to shape the student's journey through this internship.
- Ms Ong Bin Lay, the Marketing and Finance officer of SingAren, for handling all the administration matters and remuneration-related enquiries in the student's tenure as the intern.
- Mr Stanley Goh, the CITS personnel in charge of the SingAren server, for assisting in getting the Eduroam statistics program up online and handling the server-end of the implementation.

The student will also like to thank his NTU supervisor, Prof Deepu Rajan, for his support and assistance on the academic side of the Industrial Orientation programme.

List of Figures

Introduction

Purpose

The 10-week Industrial Orientation program undertaken by the student is a partial fulfillment of the requirements in pursuing the degree of Bachelor of Engineering (Computer Science) in Nanyang Technological University. This report is presented as a summary of the activities and observations gained through an internship with Singapore Advanced Research and Education Networks (SingAren).

In the first chapter, a brief history and description of the company will be explored and then followed by the objective and scope of the project.

Company Profile

SingAren is a non-profit society developed as Singapore's Research and Education Network (REN) to maintain connectivity among its member institutes in Singapore and RENs in other countries. The motivation behind SingAren's operation is to promote R&D in next-generation internet technologies, facilitate cost-effective methods of adopting advanced Internet technologies and foster collaborations between local research institutes with the global community.

SingAren began as a government-funded project established in 1997 as an effort to connect Singapore's research and education community to the international scene. It was financed by the then Telecom Authority of Singapore(TAS) and National Science and Technology Board(NSTB), the previous incarnations of the current Infocomm Development Authority(IDA) and Agency for Science, Technology and Research(A*STAR) respectively. The international connection was launched on 7 Nov 1997 by the Assistant to the USA President for Science, John Gibbons and Singapore's Minister for Education and the Chairman of National IT Committee, RAdm Teo Chee Hean. On 1 October 2003, SingAren was registered as an official Society to widen its role of serving the user community. (1)

Institution Partnerships

As the national REN, SingAren has preserved close working relationships with global Next-Generation Internet organizations such as Asia Pacific Advanced Network (APAN) and Trans-Eurasia Information Network (TEIN3). Currently, SingAren has formed partnerships with local universities such as NTU and NUS, polytechnics and government institutes like IDA in deploying the high-performance network connection and fostering innovative research in certain areas such as E-learning and cloud-based transfer services.

Task

Throughout the internship period, the student was attached to the Eduroam data analytics project. The student was tasked with the responsibility of maintaining the Eduroam usage statistics program implemented by SingAren to monitor the operation of the Eduroam service and detect any server errors that may occur during the service period. The student would learn and apply the techniques utilized in the automation of the current log processing system. The student would also learn to apply the programming knowledge and software development concepts acquired in the academic course to process the log data from the Eduroam logging system and present the data in the form of graphs. The student was required to create a visualization tool using available web-based technologies for institution users to analyse the data easily and track down problems if needed. The student would collaborate with CITS personnel attached to SingAren in order to get the Eduroam data analysis system established on the Eduroam server, enabling members to view the statistics graphs through a web browser.

Background of the Eduroam Project

In the global research environment, researchers often travel to other institutions for conventions and research collaborations. One of the essential services these researchers need in their work is Internet access. One solution is to subscribe to the roaming data plans provided by the telecom operators. However this option is expensive and unnecessary since most research institutions provide wireless Internet access. The only remaining problem lies in whether the visiting researchers have the permission to access these networks.

Eduroam is an international secure roaming service implemented for the research and education communities to address the issue of providing seamless internet access to visiting academics and research fellows to the institution. It is currently implemented in 74 countries from North and South America, Europe and Asia. Eduroam provides users from participating institutes seamless and secure Wi-Fi access to the Internet at other participating institutions around the world.

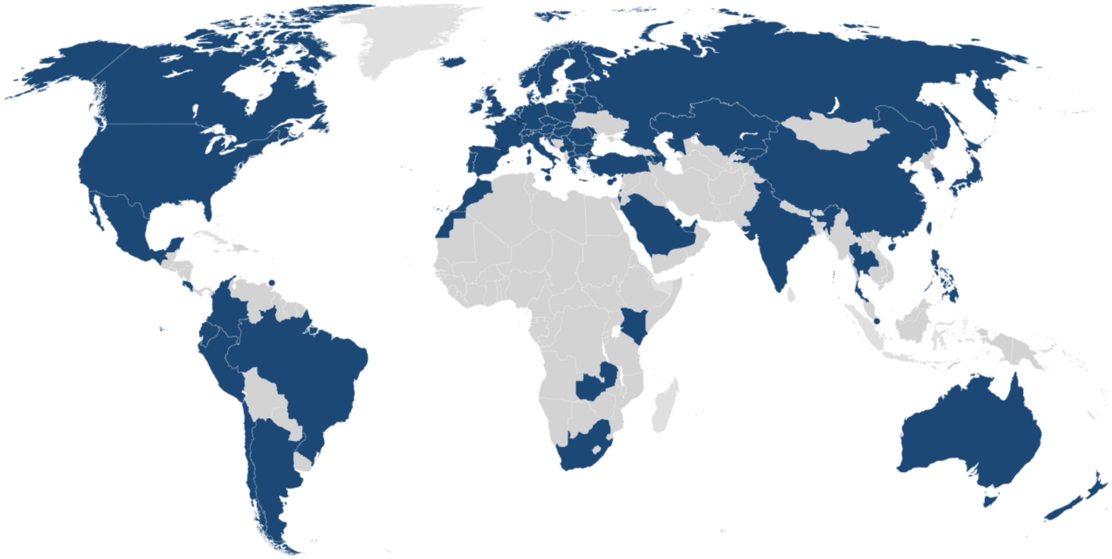


Figure 1 Map of eduroam coverage

Eduroam utilises a hierarchical RADIUS server infrastructure and the IEEE 802.1x standard networking protocol to provide roaming Internet service across research and education networks across the world. (2) The RADIUS server hierarchy is involved in forwarding user credentials to the home institution of the user for validation and verification purposes. The IEEE 802.1x technology standard defines the end-to-end encryption and authentication protocol over the wireless connection offered by Eduroam.

Authentication and authorisation of access is performed under the following arrangement, where users are authenticated at their home institutions which act as Identity Providers (IdPs) and the authorisation for user network access is handled by the host institution providing the access points as a Service Provider(SP). When the user first requests authentication at a SP, the SP looks for the realm of the user that indicates the IdP associated with the user. The realm can be found in the form of userid@IdP.TLD, where IdP refers to the domain name of the home institution and TLD indicates the top-level domain in which the institution belongs to. For instance, the National University of Singapore (NUS) belongs to the top-level .sg domain. Each institution has its own RADIUS server connected to a local user database. With the information from the user realm, the RADIUS server can route that request to the following suitable RADIUS server until the correct Identity Provider is reached. At the Identity Provider, the user authentication details can be validated with the user data in the local database and transport the validation result back to the SP. User access authorisation is then conducted by the SP and network access is granted to the user by allocating a specific VLAN for visitors.

The secure transmission of user authentication details across the RADIUS server infrastructure is maintained by the IEEE 802.1x standard, which utilises the Extensible Authentication Protocol (EAP). The function of EAP is to encapsulate the authentication data such that only the user and its IDP are aware of the actual authentication process, and other users will be unable to hijack the connection after successful verification. WPA2-Enterprise, a security mechanism for wireless networks, is used for encrypting the authentication data to prevent eavesdropping for the user credentials.

Organisation of Eduroam Infrastructure

The following diagram is an illustration of the hierarchical RADIUS server infrastructure adopted by Eduroam. It is divided into 3 levels, namely the Confederation level(ETLRs), the Federation level(FLRs) and lastly, the Institutional level(IdPs and SPs). (3)

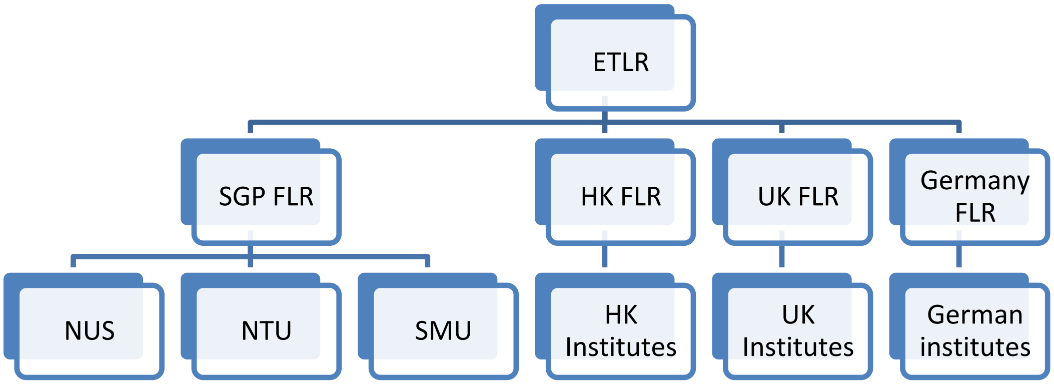


Figure 2 Hierarchy of Eduroam servers

The confederation top-level RADIUS (TLR) servers are authoritative top-level domains coordinating the operation of eduroam servers within the regions under their jurisdiction. Each TLR has a list of connected country domains serving the appropriate National Research and Education Networks(NRENs). The TLRs accept requests for federation domains under their authority and then forward them to the respective federation RADIUS server. Requests for domains not under their authority would be forwarded to the proper confederation server. Currently, there are two confederation-level TLRs based in Europe known as ETLRs that handle requests from the European NRENs and also NRENs from other parts of the world.

The federation-level RADIUS (FLR) servers are responsible for handling requests between the IDPs/SPs and the upstream confederation TLRs. Each FLR possess a list of connected IdPs and SPs, and they redirect requests to the proper SP based on the user realm. Requests with destinations not under their authority would be forwarded upstream to the TLR. The FLR is operated by the NREN of the territory to manage connections from its member institutes. For instance, the FLR for Singapore is maintained by SingAren. (4)

The bottom-level servers consist of IdPs and SPs participating in eduroam. They are responsible for the actual authentication of their own users through the validation of the user's credentials with their internal database. They are also tasked with forwarding access requests from visiting users to the associated federation server.

The hierarchical structure of RADIUS servers hence ensures that the transport of user authentication details can be conducted effectively between the SPs and the IdPs across different federations of networks.

Eduroam Statistics Project

The Eduroam statistics project was set up by the SingAren staff in 2013 as an effort to allow member institutes to track their eduroam usage statistics and detect any problems that may arise in the operation of the Eduroam system. Currently, the Eduroam statistics project consists of a conversion program running on the eduroam server and a rudimentary graphing of the usage statistics for each institution. The Eduroam statistics system was currently operated manually by the SingAren staff without automation.

Software Requirement Specifications

The following software requirements were established by the student under the guidance of the company supervisor and will be adhered to during the internship project.

1. Data visualisation for the amount of authenticated users for each Institute of Higher Learning (IHL) is to be expressed in the form of graphs.
    a. Daily graph: displays the number of unique users for each day in the current month.
    b. Monthly graph: displays the number of unique users for each month in the current year.
    c. Yearly graph: displays the number of unique users for each year.
2. Data visualisation for the amount of individual authentication requests received by the eduroam server are to be expressed in the form of graphs
    a. Hourly graph: displays the number of accepted/rejected requests in the previous day at hourly intervals.
    b. Daily graph: displays the number of accepted/rejected requests for each day in the current month.
    c. Monthly graph: displays the number of accepted/rejected requests for each month in the current year.
3. The authenticated users are defined in 3 categories:
    a. LocalUsers: denotes the staff and students from a certain IHL using eduroam overseas or in other IHLs.
    b. Visitors: denotes foreign exchange personnel using eduroam in that specific IHL.
    c. Rejected: denotes all rejected users from the certain IHL or overseas.
4. The log file processing program is to be refined to increase modifiability and maintainability by reducing the amount of code needed for adding new IHLs to the system or the addition of new features.

5. The data visualisation should be viewed on a javascript-enabled html webpage within the statistics portion of the SingAren website.

6. The graphs should clearly define the data values such that users can deduce the usage statistics easily.

7. The log extraction program is to be written in Python, while the data visualisation tool is coded in Javascript and HTML.

8. The specified environment for the log file processing system is Ubuntu Linux running Python v2.7.3 and Apache Server.

9. The Python scripts are to be executed on a daily basis with a task scheduler in the server,

Project Overview

The Eduroam statistics project is divided into two sections, namely the logfile extraction program and the data visualisation tool. The log file extraction program extracts and processes the data from the eduroam log file obtained from the Eduroam server. The data visualisation tool utilises the extracted data from the previous program to display the usage statistics in the form of graphs on the SingAren website for eduroam members to inspect the usage data.

Part 1: Log File Extraction

The log file extraction program, named as “convert.py”, takes in the Eduroam log file of the previous day as input, and returns the UniqueUsers files for each IHL and the extracted data containing the number of users using the Eduroam system as outputs. The contents of the log file is first extracted and saved into a list of log entries for processing in the convert.py script. The UniqueUsers files which contain the lists of accepted/rejected unique users using eduroam for each IHL during the particular month/year, are also extracted and saved into the respective variables.

Next, the script sorts each entry according to whether the user access is accepted or rejected. If access is rejected for the specific user, the script takes note of the source of the user access request, and then adds the user to RejectUniqueUsers files of the user’s home institution if it is not present in the file previously. The number of unique users for the certain IHL will also be collated. If access is accepted for the user, the script takes note of the entry and exit points of the user access request, and then adds the user to UniqueUsers files of the user’s home institution if it is not in the file before the running of the script. The number of unique users for the certain IHL will be collected, together with the number of unique users from overseas using eduroam at each IHL. The extraction process will end after all the log entries have been checked by the script.

The extracted data will then be written into a daily results text file to summarise the usage statistics obtained in the operation. The extracted data will also be saved into a database as storage such that the visualisation tool can use it as a data source for displaying the graphs.

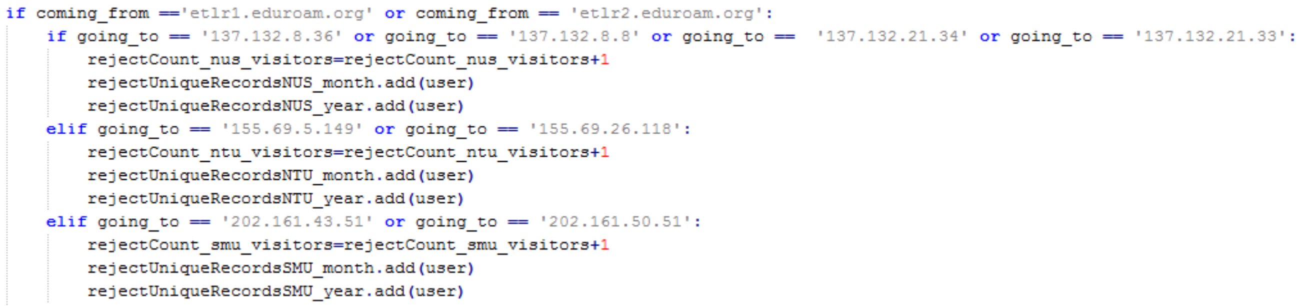
Part 2: Data Visualisation Tool

The data visualisation tool is represented by CreateHTML.py, which produces the HTML files for each IHL present in the eduroam system. The HTML files act as a container for structuring the graphs in a neat fashion. The graphs themselves are created using a javascript charting library and they utilise the database as a data source for displaying. The HTML files are then put on the web server for display as part of the statistics webpages of the SingAren website.

Revamp of Log Extraction Component

Problem

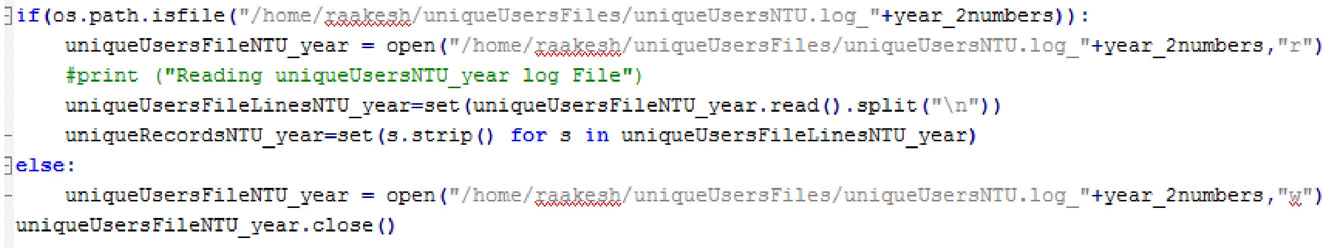
The existing Python code of the Log file extraction program was written in a procedural format, such that there are instances of spaghetti code whereby there are many if-statements for log file data extraction for each IHL without any proper segmentation. At the same time, the details of each IHL are hard-coded in the program, making maintenance difficult as the whole program has to be taken offline to perform changes. An example of this issue can be seen in the code excerpt below.



```
if coming_from =='etlr1.eduroam.org' or coming_from == 'etlr2.eduroam.org':
    if going_to == '137.132.8.36' or going_to == '137.132.8.8' or going_to ==  '137.132.21.34' or going_to == '137.132.21.33':
        rejectCount_nus_visitors=rejectCount_nus_visitors+1
        rejectUniqueRecordsNUS_month.add(user)
        rejectUniqueRecordsNUS_year.add(user)
    elif going_to == '155.69.5.149' or going_to == '155.69.26.118':
        rejectCount_ntu_visitors=rejectCount_ntu_visitors+1
        rejectUniqueRecordsNTU_month.add(user)
        rejectUniqueRecordsNTU_year.add(user)
    elif going_to == '202.161.43.51' or going_to == '202.161.50.51':
        rejectCount_smu_visitors=rejectCount_smu_visitors+1
        rejectUniqueRecordsSMU_month.add(user)
        rejectUniqueRecordsSMU_year.add(user)
```

Figure 3 Part of old logExtract code

Multiple instances of repeated logic are detected since the code has to perform the same actions for each IHL. For example, the code for opening the uniqueUserFiles for every IHL shown below is replicated frequently for each IHL and time period, namely monthly and yearly.



```
if(os.path.isfile("/home/raakesh/uniqueUsersFiles/uniqueUsersNTU.log_"+year_2numbers)):
    uniqueUsersFileNTU_year = open("/home/raakesh/uniqueUsersFiles/uniqueUsersNTU.log_"+year_2numbers,"r")
    #print ("Reading uniqueUsersNTU_year log File")
    uniqueUsersFileLinesNTU_year=set(uniqueUsersFileNTU_year.read().split("\n"))
    uniqueRecordsNTU_year=set(s.strip() for s in uniqueUsersFileLinesNTU_year)
else:
    uniqueUsersFileNTU_year = open("/home/raakesh/uniqueUsersFiles/uniqueUsersNTU.log_"+year_2numbers,"w")
uniqueUsersFileNTU_year.close()
```

Figure 4 Opening the uniqueUsersFile of a IHL

There is heavy reliance of global variables in the code and proper structuring of code is non-existent with the lack of functions. This makes the code hard to test for validity as there are several different portions in the conversion program such that it would be difficult to pinpoint the exact location of the error. Adding new functionalities would also be tedious due to the various dependencies between each component of the programs.

In conclusion, the existing code is unable to handle the expected changes such as the addition of new IHLs to the statistics system without needing a significant amount of code to be edited accordingly.

Design & Implementation

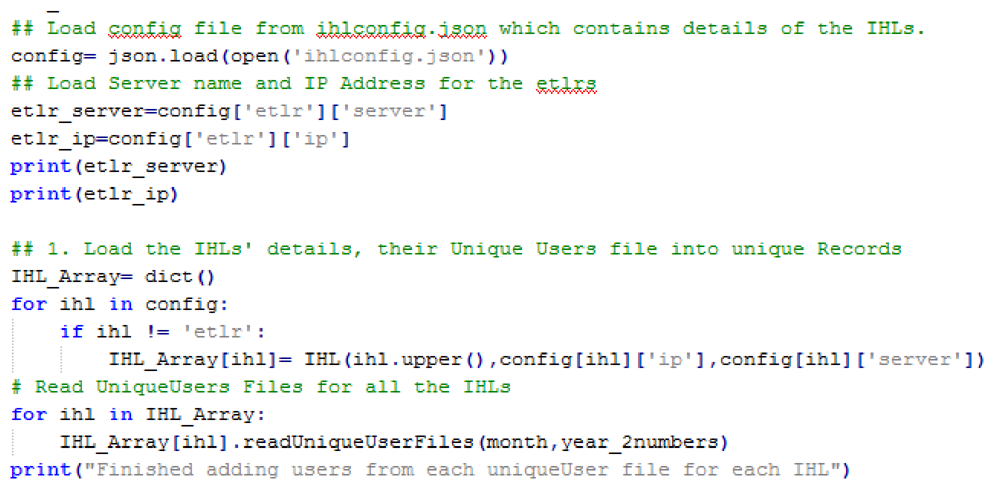
In lieu of the new requirements and the problems of the legacy code, there is a need for redesigning the structure of the program in order to fulfil these conditions.

The conversion program is divided into 4 components: main(), logExtract(), results() and saveCSV(). The main() module is in charge of coordinating the different components and direct the control flow of the program. The logExtract() module extracts the required data from the contents of the log file into local variables and the results() saves the extracted data in a daily results text file. The saveCSV() module stores the same extracted data into CSV files that would be used for the data source of the displayed graphs.

IHL-specific operations such as the reading and writing of uniqueUserFiles are consolidated under the class IHL. The IHL class also contains variables to assist in the gathering of related statistics through the log extraction. This design simplifies the process of repeated file I/O operations and the sorting of data into the respective IHLs.

Main() module

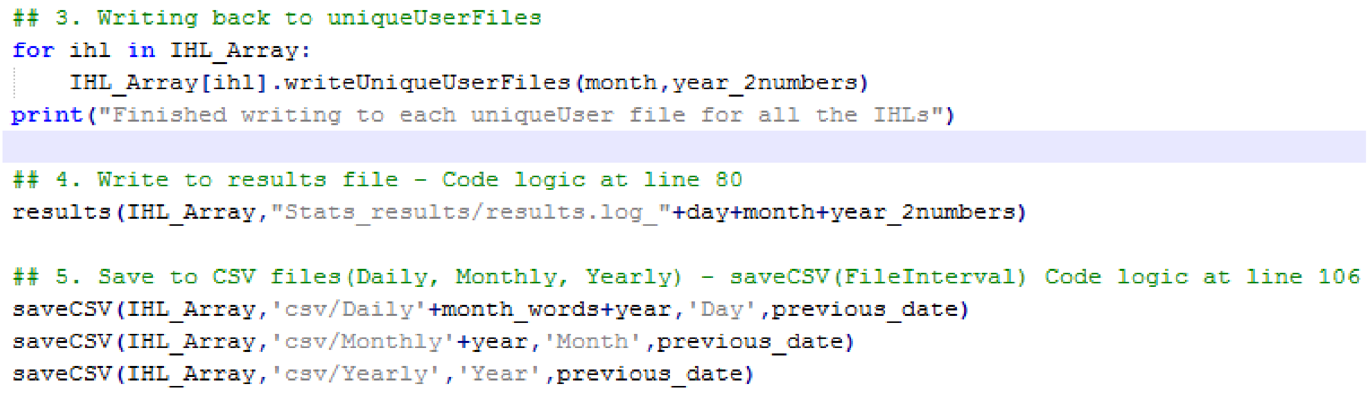
The main() module first opens the previous day's log file and stores its contents into a list variable logData. It also loads the configuration file ihlconfig.json which contains the server names and ip addresses of every IHL and the ETLR servers which indicates overseas users. These details are inserted dynamically into the log extraction system as shown in the code excerpt below.



```
## Load config file from ihlconfig.json which contains details of the IHLs.
config= json.load(open('ihlconfig.json'))
## Load Server name and IP Address for the etlrs
etlr_server=config['etlr']['server']
etlr_ip=config['etlr']['ip']
print(etlr_server)
print(etlr_ip)

## 1. Load the IHLs' details, their Unique Users file into unique Records
IHL_Array= dict()
for ihl in config:
    if ihl != 'etlr':
        IHL_Array[ihl]= IHL(ihl.upper(),config[ihl]['ip'],config[ihl]['server'])
# Read UniqueUsers Files for all the IHLs
for ihl in IHL_Array:
    IHL_Array[ihl].readUniqueUserFiles(month,year_2numbers)
print("Finished adding users from each uniqueUser file for each IHL")
```

Figure 5 Dynamic instantiation of the IHLs

The module then calls logExtract() to process logData and IHL_Array to contain the extracted statistics. Next, it proceeds to write all the unique user files for all the IHLs. Finally, it proceeds to call the functions results() and saveCSV() to save the data in a portable format as illustrated below.



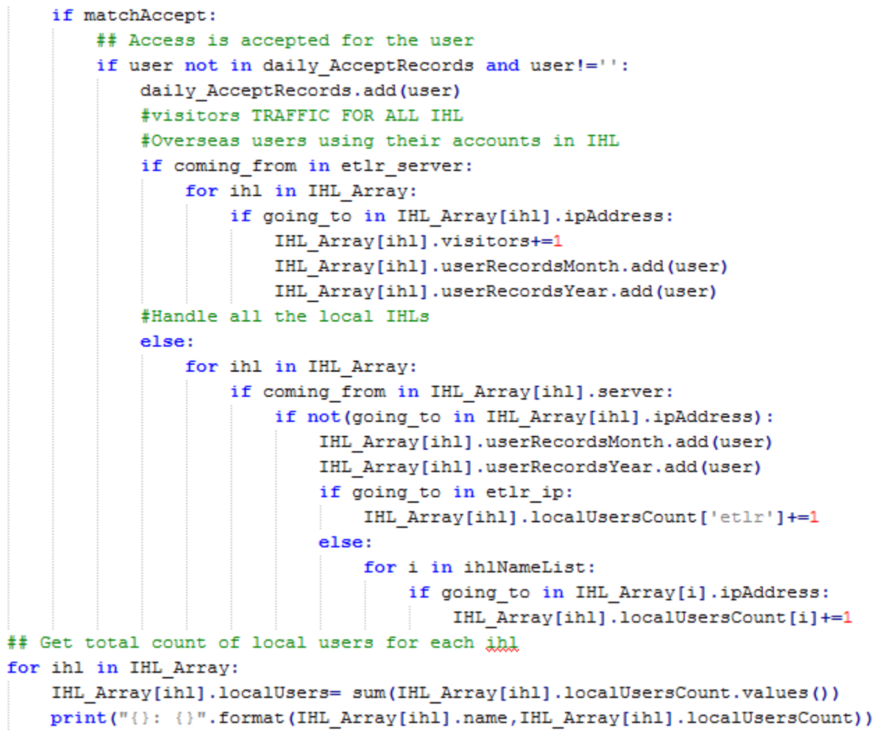
```
## 3. Writing back to uniqueUserFiles
for ihl in IHL_Array:
    IHL_Array[ihl].writeUniqueUserFiles(month,year_2numbers)
print("Finished writing to each uniqueUser file for all the IHLs")

## 4. Write to results file - Code logic at line 80
results(IHL_Array,"Stats_results/results.log_"+day+month+year_2numbers)

## 5. Save to CSV files(Daily, Monthly, Yearly) - saveCSV(FileInterval) Code logic at line 106
saveCSV(IHL_Array,'csv/Daily'+month_words+year,'Day',previous_date)
saveCSV(IHL_Array,'csv/Monthly'+year,'Month',previous_date)
saveCSV(IHL_Array,'csv/Yearly','Year',previous_date)
```

Figure 6 Writing of uniqueUserFiles and saving the results

This method of structuring eases the process of tracking the code logic during debugging sessions.

LogExtract()

The module logExtract() takes in the arguments logData, IHL_Array and the details of the ETLRs. It uses the Python in-built regex library to sort the entries in logData for accepted and rejected entries. The source and destination of each authentication request is then extracted and added to the statistics. The process can be seen in the following code.



```
    if matchAccept:
        ## Access is accepted for the user
        if user not in daily_AcceptRecords and user!='':
            daily_AcceptRecords.add(user)
            #visitors TRAFFIC FOR ALL IHL
            #Overseas users using their accounts in IHL
            if coming_from in etlr_server:
                for ihl in IHL_Array:
                    if going_to in IHL_Array[ihl].ipAddress:
                        IHL_Array[ihl].visitors+=1
                        IHL_Array[ihl].userRecordsMonth.add(user)
                        IHL_Array[ihl].userRecordsYear.add(user)
            #Handle all the local IHLs
            else:
                for ihl in IHL_Array:
                    if coming_from in IHL_Array[ihl].server:
                        if not(going_to in IHL_Array[ihl].ipAddress):
                            IHL_Array[ihl].userRecordsMonth.add(user)
                            IHL_Array[ihl].userRecordsYear.add(user)
                            if going_to in etlr_ip:
                                IHL_Array[ihl].localUsersCount['etlr']+=1
                            else:
                                for i in ihlNameList:
                                    if going_to in IHL_Array[i].ipAddress:
                                        IHL_Array[ihl].localUsersCount[i]+=1
## Get total count of local users for each ihl
for ihl in IHL_Array:
    IHL_Array[ihl].localUsers= sum(IHL_Array[ihl].localUsersCount.values())
    print("{}: {}".format(IHL_Array[ihl].name,IHL_Array[ihl].localUsersCount))
```

Figure 7 Accepted portion of logExtract module

One key improvement in the log extraction component is the usage of the dictionary variable “localUsersCount” to indicate the users from the specific IHL who are accessing eduroam at the other IHLs and overseas. E.g. IHL_Array[ihl].localUsersCount[‘etlr’] refers to the users who using eduroam overseas. The variable “localUsersCount” reduces the amount of variables needed in total to store the data and hence makes the code less prone to errors.

Results()

The results() module writes the summary of extracted data obtained from the log file for the day. It uses IHL_Array as an argument to save the results into an text file for references purpose. The contents of the results file are as follows:

a. Total number of localUsers from each IHL who are abroad and in other IHLs.
b. Total number of localUsers from each IHL in total.
c. Total number of visitors to each IHL.
d. Total number of unique/rejectUnique users to each IHL for the month.
e. Total number of unique/rejectUnique users to each IHL for the year.
f. Total number of rejected accesses from each IHL for the day.

SaveCSV()

The saveCSV() module is a crucial component of the log processing system as it bridges the gap between the Python-based log file extraction program and the web-based data visualisation tool. Since it is a new component, more time was needed to ensure the seamless operation of the module with testing using mock data. Three CSV files were used for the three graphs of different intervals, daily, monthly and yearly, and hence the module is called three times by the main() method with different parameters.

The format in which the data is arranged was also important since the charting library used had a strict requirement in terms of the organisation of the data. The saveCSV() module utilises the Python in-built csv library to handle the reading and writing of the CSV files. Since there is a difference in the definition of newlines for Windows and Linux systems which affect file I/O, I decided to let Python handle the differences automatically through its usage of universal newlines and not explicitly define the newline manually.

Data Visualisation Component

Problem

The old data visualisation tool implemented by the previous engineer, upon initial inspection, was not operational since the statistics has not been updated since January 2015. The problem was found to be in the change of authentication details for the Google account associated with the statistics program, and the online Google spreadsheets are not updated. Since the charting library for the old visualisation tool, Google Charts, was dependant on the online spreadsheets as a data source, therefore the graphs on the statistics were not up-to-date.

In addition, the Google Data API used in the legacy code has been deemed deprecated and it may not work in the future. This meant that the Google Charts API was not a viable solution with the present situation. Hence, a new charting library and a new data source have to be used.

The existing code also delegates one html creation python script for each IHL, which is inefficient since the logic behind the scripts is the same for all IHLs and the approach is not scalable if many IHLs get involved in the Eduroam statistics.

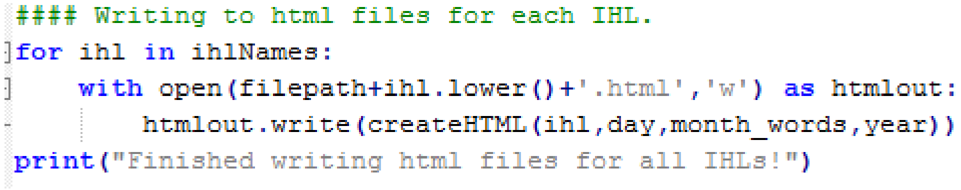
Design & Implementation

The data visualisation process has to be redesigned from scratch in order to create accurate graphs on the webpages. In addition to the saveCSV() module in convert.py, an external charting library was used in place of the preceding Google Chart API.

Dimple.js, an external Javascript charting library by Align Alytics (5), was used for creating the graphs on the statistics web pages at the SingAren website. Described as an object-oriented API designed for business analytics, Dimple.js allows the creation of flexible axis-based graphs with minimal code using d3js, a popular Javascript library that performs DOM manipulation driven by data. The advantage of using Dimple rather than d3js by itself lies within the gentler learning curve and the various features offered by Dimple such as chart templates compared to the low-level approach of pure d3js. Furthermore, Dimple also allows exposure of core d3js components for customization such as adding the title to the chart.

The creation of the daily, monthly and yearly charts follow a similar method. First, the CSV file is obtained through a HTTP GET request encapsulated by d3js' parser method (6). Next, the data is filtered according to IHL such that the correct data will be displayed for the IHL's statistics page. The graph is then created by adding the code determining the type of x and y axis, and the series of data that is needed to be displayed.

Only one html generation script, CreateHTML.py, is needed for creating the web pages of all the IHLs. The script also depends on ihlconfig.json to determine the names and the amount of web pages needed to be created. The html code is encoded within a multi-line string and written to the respective files as illustrated in the following snippet.



```
#### Writing to html files for each IHL.
for ihl in ihlNames:
    with open(filepath+ihl.lower()+'.html','w') as htmlout:
        htmlout.write(createHTML(ihl,day,month_words,year))
print("Finished writing html files for all IHLs!")
```

Figure 8 Writing of html files for all IHLs

Additional Feature

An additional feature was also implemented to the Eduroam statistics system. In order to track the amount of individual authentication requests that the eduroam server at SingAren is handling, a separate Python script, convertSingaren.py, was written for extracting these details from the log file and saved into a separate CSV file named ServerLoad2015. This was done by creating a class ServerLoad in the script and let the main() function instantiate that class to perform the specific operations. The intention of creating this ServerLoad class was to allow the integration of the new functionality into the main conversion script when it is needed.

A separate CreateHTML python script was also written to generate the hourly, daily and monthly charts for the total eduroam statistics web page.

Conclusion

Throughout the whole internship period, the student has successfully revamped the log file processing system and deployed the data visualisation tool on the SingAren website, applying the object-oriented approach and software engineering concepts learnt in the course of study. Certain amendments were made to the code logic such as the calculation of visitors to each IHL and the addition of a new IHL to the Eduroam statistics.

The eduroam statistics project was showcased over a sharing session with the SingAren staff and representatives from the institutions involved in Eduroam on 21 July. The student gave an overview of the back-end implementation of the statistics program and demonstrated the analysis of the graphs on the SingAren statistics web pages. Several comments were given in regards of the look and feel of the data visualisations. For instance, Prof Lawrence, a member of the SingAren committee, suggested a “dashboard” user interface concept as a method to allow users to view all the essential data quickly and only present more details when the user selects the specific chart.

The sharing session also allowed the student to understand more about the implementation of Eduroam in local institutions and the challenges faced by IT teams as they attempt to set up multiple servers across the campus. For instance, the SUTD team had to contend with shifting its servers to its current campus while deploying eduroam. The ASTAR team had to create their user accounts manually due to the differences in network protocol.

As a conclusion, the student had gained valuable experience during his internship period with SingAren in terms of both technical and non-technical aspects of the IT industry.

Bibliography

1. **Singapore Advanced Research & Education Networks.** About Us- SingAren. *SingAren Web site.* [Online] July 31, 2014. [Cited: July 15, 2015.] http://www.singaren.net.sg/history.php.

2. **GEANT Association.** eduroam in a nutshell. [Online] August 12, 2014. https://wiki.geant.org/display/H2eduroam/eduroam+in+a+nutshell.

3. **M. Milinović, Srce / CARNet, Stefan Winter, RESTENA and members of the SA3 T2 group.** eduroam Policy Service Definition. [Online] July 26, 2012. https://www.eduroam.org/downloads/docs/GN3-12-192_eduroam-policy-service-definition_ver28_26072012.pdf.

4. **Singapore Advanced Research & Education Networks.** eduroam Project - Welcome to eduroam in Singapore. *SingAren Web Site.* [Online] July 31, 2014. http://www.singaren.net.sg/Eduroam.php.

5. **PMSI-AlignAlytics.** DimpleJS. *GitHub.* [Online] https://github.com/PMSI-AlignAlytics/dimple.

6. **Bostock, Mike.** d3 CSV. *GitHub.* [Online] December 14, 2014. https://github.com/mbostock/d3/wiki/CSV.