AUTOMATED ATTENDANCE TAKER USING PASSIVE MAC ADDRESS PROBING

By

Low Chun Fai

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMMUNICATION AND NETWORKING

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2018

DECLARATION OF ORIGINALITY

I declare that this report entitled "**AUTOMATED ATTENDANCE TAKER USING PASSIVE MAC ADDRESS PROBING**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : _____

Date : _____

Acknowledgement

First of all, I would like to thank my supervisor, Mr. Aun Yichiet for giving me all the support and guidance throughout this project. He had given me all the guidance when I have problems during this project. This project is progressing smoothly thanks to his support and guidance.

Next, I would like to thanks my friends and family who had given all the encouragements and mental support during this project. Their continuous support and encouragements helped me to go through all the difficult times throughout the project.

Abstract

This project is titled "Automated Attendance Taker using Passive MAC Address Probing". It is an automated and unsupervised attendance taker that can complete the attendance taking process in a minimal amount of time and minimal human intervention. The MAC address of the student's smartphone is used as an indicator to proof that the student attended the class. The student will only require to turn on their Wi-Fi in their smartphone and the system is able to capture the Probe Request frames when the smartphone is searching for Wi-Fi connection. If the Probe Request frames is captured by the Wireshark, it means that the student is in the range of the classroom and attendance will be given after the validation. This automated system is able to accurately check the attendance of the students and check the continuous presence of students in the classroom to reduce the amount of proxy attendance. This automated attendance system is also able to complete the attendance taking process time effectively as the whole process can be done with a few clicks of button. Next, the automated attendance system is also unsupervised where the lecturer or tutor will only require to click a few button and the system will complete the attendance taking process automatically. The tools that will be used in this system is network analyser, Wireshark, a Main Graphical User Interface and Main Program developed in Java, Wamp Server as a web server for student registration purpose as well as MySQL as the database for storing students and mobile phone MAC address information. Wireshark is used to capture all the 802.11 frames in the network and export a text file to be input into the system. A few main algorithms are developed in this project to form a complete automated attendance taker. The algorithms are Student ID to MAC Address Binding Algorithm, MAC Address Extraction Algorithm, Cross-Checking Registered MAC with Captured MAC Algorithm, Anti MAC Spoofing Algorithm and Timestamp Validation Algorithm. Each algorithms will process the captured packets from Wireshark to determine the validity of the students' attendance. After the validation of the attendance, the attendance information will be updated into the database.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

RFID	Complementary Metal Oxide Semiconductor
IOT	Metal Oxide Semiconductor Field Effect Transistor
MAC	Integrated Circuit
SSID	Service Set Identifier

Chapter 1: Introduction

1.1 Background Information

Student attendance management system is a must in nearly all the education institutions such as colleges and universities to keep track of the attendance of the students. Attendance of the students is important as it is correlate positively to the students' performance during their studies in the university or college. However, keeping track of the attendance of the students is a very difficult task for the tutor and lecturers as there are many education institutions are still implementing the traditional attendance system where the attendance is taken by passing the attendance list around the lecture class and students will mark their attendance on the attendance list. The respective lecturer or tutor will then update the students' attendance on the database.

1.2 Motivation and Problem Statement

The problem with the traditional attendance system is that the whole process of taking attendance will be very **time consuming**. For example, there are 120 students in the classroom and each students will take around 30 seconds to mark their attendance and pass the attendance list to the next student, it will take around 1 hour to have all the students in the class to mark their attendance. Other than time consuming, the traditional method of taking attendance will also **interrupt the teaching process** in the class where students will have to pass around the attendance list. The next major problem of the traditional method of taking attendance is the attendance of the students is **not being tracked accurately**. Some students might not get to mark their attendance due to the human errors occur when passing the attendance list. If there are too many students in the class, the students might not able to get hold of the attendance list to mark their attendance. In contrast, some students might help their friend to mark their attendance when they absent from the class. With the traditional method of taking attendance, it is very hard for the lecturers or tutors to authenticate the person who signed on the attendance list due to the large amount of students in the class. Hence, the traditional attendance system is not tracking the attendance accurately and time consuming.

Nevertheless, by using the physical paper to keep track of the attendance is not reliable as there are possibilities that the physical attendance list might gone missing when students passing around the list. It is very hard to know the exact location of the attendance list due to the large amount of students in the class. The attendance list might get mixed up with the notes of the students without the knowledge of the students. Hence, the traditional attendance system is not tracking the attendance accurately and time consuming.

By looking on the scenarios above, it shows that the traditional method of taking attendance is very time consuming and not tracking the attendance accurately. It is very crucial to ensure that the attendance taking process is smooth and accurate so that the students and lecturers are able to fully utilize the time on the learning and teaching process. It is also very important to ensure the accuracy of the attendance so that the lecturers or tutors are able to know the students who attended and absent from the class. In order to solve the problems stated above, this project is being proposed to develop an automated attendance system. The automated attendance system can complete the attendance taking process in a very short time, less interference of the teaching process and is able to keep track of the attendance accurately. This project presents a pervasive, accurate and scalable **Automated Attendance Taker using Passive MAC address probing.** The proposed system is based on opportunistic algorithm to detect the presence of students using their corresponding mobile phone's MAC address within some confined coverage area of a network access point.

1.3 Project Scope

In this project, an automated attendance taker using passive MAC address probing will be develop as the final deliverable to solve the problem stated at the previous section, which is improve the efficiency and accuracy of the attendance taking process with a minimal amount of active human interaction. In order to develop this system, a wireless access point and a computer is needed in the classroom. The network analyser, Wireshark, will be needed to capture all the packets from the wireless access point as well as the students' smartphone. The computer with Wireshark installed will capture the packets and process

the packets by the main program to confirm the attendance of the student. The students' information and MAC address information will be stored in the database, which is MYSQL database.

Despite that, there might be argument where there are possibilities that some student might not have a smartphone with them. In this project, we will assume that every students in the class will have at least 1 mobile device with them such as smartphone or tablets. The accuracy and the reliability of the data collection of MAC address is also an argument where there are possibilities that student will changes their phones or brings the phone of their friend for taking attendance. In this proposed system, the MAC address to student ID mapping will be done at the first 2 weeks of the semester so the student can only take their attendance based on that MAC to student ID mapping. Student can request to change their MAC to ID mapping if there are exception cases such as student changing their smartphones.

1.4 Project Objective

The main objectives of this project is -:

- **To develop an automated and unsupervised attendance taking system to improve the efficiency of the attendance taking process.**

The automated and unsupervised attendance taking system is able to minimize the amount of time needed to complete the attendance taking process. This automated attendance taker is also able to complete the attendance taking process without any human intervention as it is an unsupervised system.

- **To design a validation algorithm to improve the accuracy of the attendance system by checking the continuous presence of students in the classroom.**

The attendance validation algorithm is able to accurately identify the presence of the students in the classroom by validating the timestamp of their MAC address in the network and thus able to improve the accuracy of the attendance as well as reducing the proxy attendance.

- **Design a MAC address to student ID binding framework for unique student's identification using mobile phone's MAC address for accurate attendance taking.**

An algorithm to bind the MAC address and student ID will be develop to identify each students in the class. This algorithm is use to capture the MAC address from the respective student and bind the MAC address with their student ID. After the mapping, the MAC-Student ID information will be saved in the database for comparing.

- **Design an algorithm to extract MAC address from SSID discovery packets**

This algorithm is designed to extract the MAC address from the SSID discovery packets from the Wireshark. After capturing the 802.11 packets, the packets will be filtered and export to the program to extract the MAC address from the packets. The MAC address will be used for identification of the students that attend the class.

- **Design an algorithm to cross-check discovered MAC to registered MAC to filter non-registered students for effective attendance taking**

  This algorithm is used to cross-check the discovered MAC address by the system with the registered MAC address in the database. The algorithm will obtain the MAC-Student ID information from the MySQL database and compared with the captured MAC address. If the captured MAC address matches the registered MAC addresses in the database, it will be used as an identification of the student's attendance.

1.5 Contribution

The innovation and contribution of this project is the development of the unsupervised attendance taker. This attendance system can contribute largely to the education institution as this system can function properly with minimal human intervention. This system can take and keep track of the student's attendance efficiently and accurately. This unsupervised attendance system can complete the whole attendance checking process automatically without interfering the teaching process in the classroom. Beside, this attendance system is also cost-effective as there are no special equipment required in this system. Only some basic equipment such as wireless access point and computer is needed for this system to work properly.

1.6 Report Organization

The report has 5 sections. For Chapter 1, the background information, motivation and problem statements, project scope and objectives as well as the project contribution will be discussed. The review of existing work and the comparison of existing work with the proposed system will be discussed in Chapter 2. Next, the system methodology and the system design will be discussed in Chapter 3. After that, system implementation and testing will be discussed in Chapter 4. Lastly, Chapter 5 will be the conclusion of the project and the discussion of limitation and future development of the system.

Chapter 2: Literature Review

2.1 Review of Existing Work

Real Time Computer Vision Algorithms in Automatic Attendance Management Systems

Shehu and Dika (2010) proposed an automatic attendance taker that used the combination of computer vision and face recognition algorithms in the process of taking attendance. This system uses the digital camera that is connected to the computer and installed in the classroom. The camera scans the whole room and captures the images of the students. The system will then extracts the faces from the image and compare with the existing student images in the database. If the extracted faces are matching the face in the database, the attendance will be successfully updated in the database.

This system completes the process of taking attendance in 3 steps, which are image capturing, face detection and face recognition. First, the images are captured and transferred on the server for processing. The camera will continuously takes pictures of the students until all the faces are successfully detected by the system. Next, the system will then detect the faces from the images using the face detection algorithms. Lastly, the system will recognize the faces extracted by the face detection algorithms and compare the faces with the existing faces stored in the database.

Strengths of this system:

- This system is able to complete the attendance checking process without interfering the teaching process in the classroom.

Weaknesses of this system:

- The accuracy of face detection algorithms is heavily depends on many external factors such as face pose, scale, position and lighting. The changes in all these factors might cause the system to not able to recognize the faces of the students and thus not able to mark the attendance of the students.

RFID Based Attendance System

Lim, Sim and Mansor (2009) proposed an attendance system that uses RFID technology. RFID is the Radio Frequency Identification. RFID uses the radio frequency wave to identify the RFID tag using the RFID reader. Each RFID tag is unique so it is very useful in authenticating the identity of the user. The RFID based attendance system is very convenient and accurate. The students can mark their attendance by placing their RFID-equipped ID card on the RFID reader and their attendance will be taken on the spot. The system is installed in the computer and the attendances taken will be stored into the database.

Nyugen and Chew (2017) also proposed a similar system which uses RFID technology in the attendance system. This system implements a number of RFID reader in the different rooms and transfers the data gathered through a Wi-Fi router to the computer. In this system, the students will touch their RFID card on the RFID reader. The RFID readers will then send the information received and transfer the information to the server through the wireless connection. The server will then processes the information and stores the data.

Strengths of this system:

- RFID-based attendance system is able to track the attendance accurately as each RFID tag is unique.

Weaknesses of this system:

- Students can mark the attendance for their friends using their RFID tag.
- Students still have to queue up to scan their RFID tag on the RFID reader in order to mark their attendance, which is time consuming.

Automatic Attendance Monitoring System using RFID and IOT using Cloud

Sharma and Aarthy (2016) proposed an attendance system that combined the RFID technology and the Internet of Things (IoT) to take the attendance of the students and process the information on the Cloud. In this system, the RFID tags will be equipped in the ID card of the students. The students will scan their RFID card on the reader and the reader will detect the unique RFID card of the students. After obtaining the data from the RFID reader, the data will be transferred through Wi-Fi adapter and stored on the Cloud. According to the authors, the concept of this system is based on Internet of Things (IoT) so that Cloud is used as the storage of the information as the information stored on the Cloud can be access anywhere and anytime.

Extra authentication also included in this system which is facial recognition. This system also added extra authentication in verifying the student's identity using the facial recognition. This system use the camera to capture the images of the students and compared with the existing images in the database. With this extra authentication, even if the students take their friend's attendance using their RFID card, the attendance will not be taken as their friend is not physically in the classroom.

Strengths of this system:

- Implementation of Cloud computing which the information can be accessed anytime and anywhere.
- Extra authentication which is facial recognition that can improve the accuracy of the attendance taken.

Weaknesses of this system:

- Students need to queue up to scan their RFID card, which is time consuming.
- The simple image comparison algorithm used in this system is not accurate as the pose, position, lighting and other factors will affect the accuracy of the facial recognition algorithms.

RFID and Pose Invariant Face Verification Based Automated Classroom Attendance System

Srivignessh and Bhaskar (2016) proposed an attendance system that implemented both RFID and Pose Invariant Face Verification. The RFID system check the students by scanning the RFID card of the students. The next verification step is done by using the face recognition algorithm. This system is similar to the system proposed by Sharma and Aarthy (2016). However, this system can verify the faces of the students in different head poses. Based on the experiments conducted by the authors, the proposed system can verifies the identity of the students about 98% correctly based on their frontal face.

The face recognition algorithm used in this system can compared the captured images with the images in the existing database in different head pose. This can be done because there are 14 photos of different head poses for each students stored in the database that are used to compare the images in different head poses.

Strengths of this system:

- Implementation of the Pose Invariant Face Verification to authenticate and verity the identity of the students. This face recognition algorithm can compare the captured images of the students with the existing images stored in the database accurately even with the different head poses, position and other external factors.

Weaknesses of this system:

- Students still have to queue up to scan their RFID tag on the RFID reader to mark their attendance, which consume a lot of time.
- The face recognition algorithm depends on the numerous images of each students to recognize the faces accurately. The huge number of images stored in the database are consuming a lot of storage space in the database.

Attendance Management System using Fingerprint Verification

Shoewu and Idowu (2012) proposed an Automated Fingerprint Attendance System (AFAS) that compare the fingerprint image of the student with the fingerprint stored in the database previously using the Automated Fingerprint Identification system (AFIS). In this system, the student with scan their fingerprint using a fingerprint reader and stored into the database along with their respective student ID. The system will then extract the unique features of the fingerprint for identification purpose. For the authentication, the student will also scan their fingerprint on the fingerprint reader and the system will use a matching algorithm to compare the current fingerprint with the existing fingerprint data in the database to authenticate the identity of the student.

Saraswat and Kumar (2010) proposed a similar system that uses fingerprint as the authentication method to check the attendance of the students. This system uses Minutiae based technique in verifying the fingerprint of the students. According to Saraswat and Kumar (2010), 'the recognition of minutiae is based on the extraction of minutiae in which binary image obtained by binarization process are submitted to fingerprint ridge thinning stage and marking of minutiae.'

Talaviya, Ramteke and Shete (2013) proposed a fingerprint based attendance system by using the ZigBee Technology. This system works like other fingerprint based attendance system where the student scan their fingerprint on the fingerprint reader. This system uses ZigBee technology in transferring the fingerprint data wirelessly to the workstation to process and stored into the database after the authentication process is done.

Strengths of this system:

- Able to uniquely authenticate the identity of each students accurately as fingerprint is unique feature for everyone that will not duplicate.

Weaknesses of this system:

- The implementation of biometric reader in every classroom is very costly and hard to conduct maintenance on all the biometric readers.

Iris Recognition Based Attendance Management System

Khatun et. al. (2015) proposed an attendance system using Iris Recognition. This system works by using web-cam to capture images of the students and send the images to the computer to process. The first image captured will be saved in the database for reference. This system will authenticate the identity of the student by capture the images of the student and compare with the existing images in the database.

This system uses MATLAB data analyzing software for iris image acquisition, iris localization, iris adjustments, iris checking, iris extraction, iris matching, data storing and authentication [4]. The system will then compare the extracted iris data and compare them with the existing data in the database to authenticate the identity of the students. If the authentication is success, the attendance will be updated into the database.

Strengths of this system:

- Able to uniquely authenticate the identity of each students accurately as iris is unique feature for everyone that will not duplicate.

Weaknesses of this system:

- The iris recognition process might affected by external factors such as poses, position, lighting and other factors.
- Capture the images of the students using the web-cam one-by-one for iris recognition, which is inefficient and time consuming.

2.2 Comparison of Existing Works with Proposed System

Existing Systems	Accuracy	Cost	Proxy Attendance	Time Consuming
RFID-Based System	High	High	✓	✓
RFID with IoT and Cloud	High	High	✗	✓
RFID with Post Invariant Face Recognition	High	High	✗	✓
Facial Recognition using Camera	Low	Low	✗	✗
Fingerprint Verification	High	High	✗	✓
Iris Recognition using Web-Cam	High	Low	✗	✓
Real Time Computer Vision Algorithms	Low	Low	✗	✗
Automated Attendance Taker using Passive MAC address probing.	High	Low	✗	✗

Table 2.1 Comparison of Existing Works with the Proposed System

The existing works are compared in terms of accuracy, cost, proxy attendance and time needed to complete the process of taking attendance. Firstly, the attendance systems that use RFID technology have a high accuracy in checking the attendance due to the uniqueness of each RFID card. However, the cost of implementing the whole RFID-based system is costly due to the high price of the RFID reader. There is also high possibility that proxy attendance could happen. It is also time consuming as the students need to queue up to scan their RFID card on the reader one-by-one. The proxy attendance problem is improved with the implementation of Face Recognition algorithm proposed by Sharma and Aarthy (2016) and Srivignessh and Bhaskar (2016).

Moreover, the facial recognition-based attendance system is lower in cost as only a camera or web-cam needed to capture the images of the students. This system also can reduce the possibility of proxy attendance as the students will need to physically in the classroom in order to take their attendance. This system also save time as this system will automatic capture the images and process the face recognition algorithm in the system without interfering the teaching process. On the other hand, the face recognition-based system is lower in accuracy as the matching algorithm of the captured images with the existing images is heavily depending on the external factors such as head poses, position, lighting and other factors. The face recognition algorithm might not function well if there are any changes in the factors stated.

Other than that, biometrics also being implemented in the attendance system such as fingerprint and iris recognition. For the fingerprint verification attendance system, it has a very high accuracy as the fingerprint is unique for every students. Nevertheless, it is costly to implement the fingerprint reader in all the classroom due to the high price of fingerprint reader. It is also time consuming as the students have to queue up to scan their fingerprint on the fingerprint sensor. Iris recognition approach is also high in accuracy and could prevent proxy attendance due to the uniqueness of the iris. It is lower in cost compared to the fingerprint system as this system only uses web-cam to capture the images of the students. However, this system also time consuming in checking the attendance as the student need to let the web-cam capture their face one-by-one.

For our proposed system, which is the **Automated Attendance Taker using Passive MAC address probing.**, can outperform other systems in terms of accuracy, cost, proxy attendance and time. The proposed system works by capturing the wireless network packets and extract all the MAC address from the packets. Each student's information is bind with their unique mobile MAC address and thus the system can take the attendance accurately and prevent proxy attendance. Only a wireless access point is needed for the system to function which is cost effective. The whole process of taking attendance is unsupervised and automatically which is time saving and not interfering the teaching process in the classroom.

Chapter 3: System Methodology and Design

3.1 Technologies Involved

NetBeans

NetBeans is an integrated development environment software that will be used to develop the main program of this project. This main program will be develop using the Java language. All the algorithms such as the Student ID-to-MAC mapping, MAC address extraction and MAC address cross-checking algorithm that are required in this projects will be develop in the NetBeans IDE. NetBeans IDE will also be used to make connection with the database, which is MySQL database, to update or retrieve the student's information from the database.

Wireshark

Wireshark is a widely-used network protocol analyzer for the users to observe and analyze the traffic in the network. Wireshark will be the main tools that will be used in this project. Wireshark is used to capture all the network packets that are going through the network. In this project, Wireshark is used to capture the 802.11 management and control frames to obtain the SSID broadcast packets in the network. The reason why Wireshark is used in this project is that Wireshark has the "monitor mode" that can be used to capture the raw 802.11 packets in the network. However, some setting needs to be done to enable the monitor mode in the Wireshark. Wireshark is not able to capture raw 802.11 packet headers in the normal mode. In order to enable "monitor mode" in Wireshark, Npcap is needed to be install instead of the default WinPcap. This is because Npcap has raw 802.11 packets capture support which enable the Wireshark to capture the raw 802.11 management packets that is needed in this project. After capturing the 802.11 packets, the Wireshark can be used to filter the specific packets needed and export as a text file to process in the main program.

Npcap

Npcap is an Nmap Project's packet sniffing library that is built for Windows. It is built from the legacy WinPcap and Libpcap libraries and with better speed, security and efficiency. Npcap is required to be installed as the Wireshark packet capture library in order to turn on the "monitor mode" in Wireshark. With this Npcap library, Wireshark is able to turn on the monitor mode and capture the raw 802.11 management frames as Npcap library has the Raw 802.11 packets support.

MySQL Workbench

MySQL Workbench is a database program that has a user friendly graphical user interface that is working with the MySQL servers and databases. MySQL Workbench has a wide range of functionalities such as SQL Development, Data Modeling, Server Administration, Data Migration as well as MySQL Enterprise Support. The main reason that MySQL Workbench is used in this project is that it provides a graphical interface which is more user friendly to be used. MySQL Workbench will be used to store the Student-ID-to-MAC information, student's basic information and attendance of the students. The main program will connect with the MySQL database and perform update or retrieve relevant information to process the student's information and MAC addresses.

Wamp Server

Wamp server is used as a web server in the host computer. The web server will be run in the local host of the computer. The webpages will be host at the host computer and can be accessed by other users. In this system, a registration webpage will be host at the host computer using the Wamp Server.

3.2 System Methodology

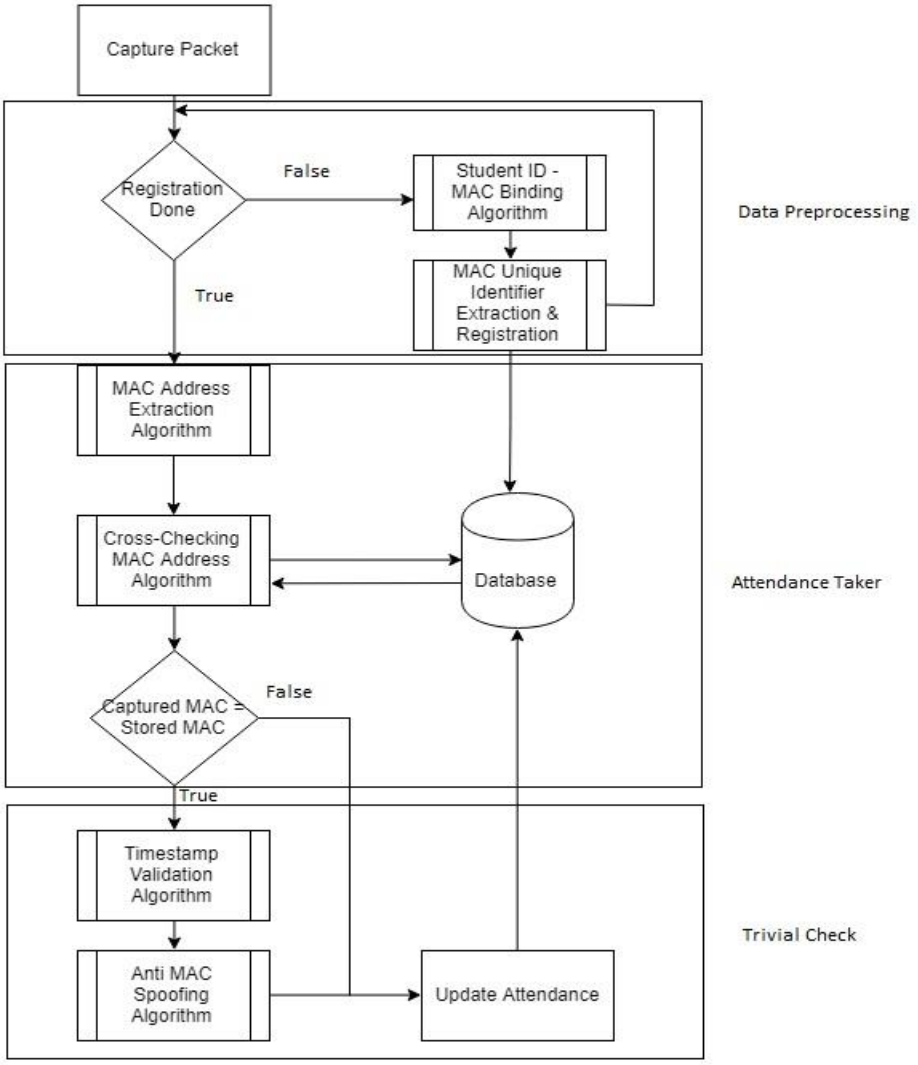


Figure 3.2 Methodology Flowchart

Figure above shows the methodology of the system. The system will begin by capturing the packets using Wireshark. After capturing the packets, the system will proceed to the Student ID – MAC address binding algorithm and the Unique Identifier Extraction and Registration if the student registration is not done. There will be a data pre-processing period for the student to register the system. If the registration is done, the system will proceed to the attendance taker that consists of the MAC address extraction algorithm and

the Cross-Checking MAC address algorithm. The MAC address extraction algorithm will extract all the MAC address from the captured packets. Next, Cross-Checking MAC Address algorithm will retrieve the registered MAC address from the database and compared the MAC Address with the MAC address extracted from the captured packets. If the captured MAC address matches the MAC address from the database, the student with that MAC address will be considered attended the class and will be proceed to further trivial checking. Otherwise, the student will be marked as absent and updated the attendance to the database. After that, the students' list will be proceed to the trivial check, which consists of the timestamp validation algorithm and the anti MAC spoofing algorithm. The timestamp validation algorithm will check whether the student's MAC address can be found in each segment of the timestamp. If the student's MAC address can be found in all the segment of the timestamp, the student will be considered in the class throughout the commencement of class. If the student's MAC address is not found in most of the timestamp segment, it means that the student is not physically in the class for a long period of time or the student leave the class before the end of the class. Hence, the student will not be given the attendance. Lastly, the system will check whether there is potential MAC spoofing attack in the network. The Anti MAC Spoofing algorithm will check the registered unique identifier in the database with the captured MAC identifier. If the identifier matches the identifier in the database, it means there is no spoofing activity to that particular MAC address. After the trivial check is done, the final attendance information of the students will be updated to the database.

3.3 System Topology

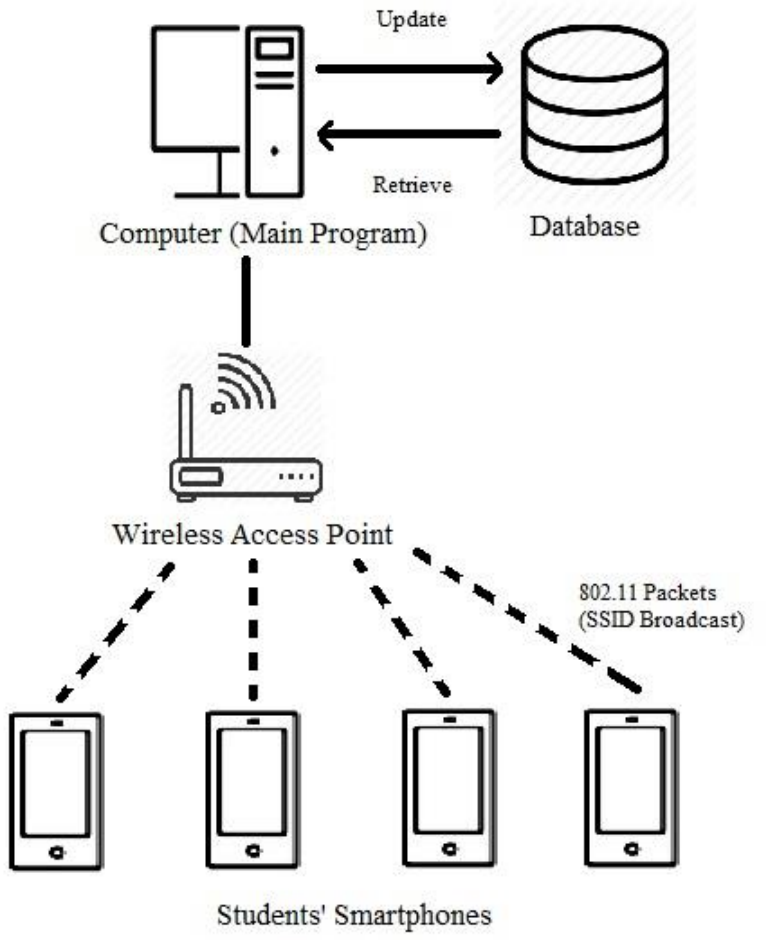


Figure 3.3 System Topology Diagram

The system will be implemented by installing a wireless access point and a computer in the classroom. The computer will be installed with the main program of this system, database and Wireshark. During the beginning of the class, the Wireshark will be turn on the start capturing the wireless packets of the network. Whenever students bring their smartphone into the class and searching for available network, their smartphone will send the Probe Request packets to look for available networks. The wireless access point will then reply the Probe Response to the smartphones. The students who attended the class will have their MAC address captured by the system and will obtain their attendance.

3.4 Wireshark

Wireshark is the main tools that is used in this program to capture the packets. In order to capture the 802.11 management packets, "monitor mode" will need to be turn on. Wireshark is required to be run as administrator to enable the "monitor mode".

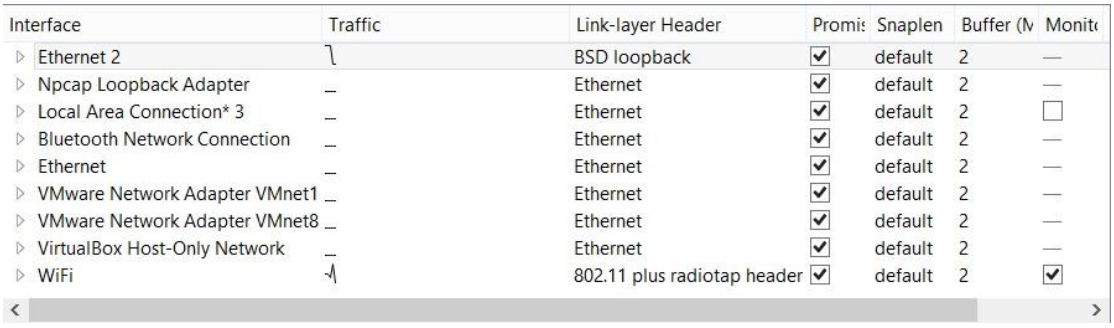


Figure 3.4.1 turning on monitor mode

After turning on the monitor mode, the Wireshark is able to capture the 802.11 wireless frames. When using Wireshark in monitor mode, the PC is not able to access to the internet as the network interface is being used as a wireless packets sniffer. Wireshark will then capture all the wireless packets near the host PC. In this project, the 802.11 management frames is the packets that are used to determine the student's attendance. Hence, a filter is required to filter out the specific packets that are needed. The packets that is needed are the Probe Request and Probe Response packets. Probe Request packet is send by the student's smartphone to obtain the nearby SSID broadcast information so that the smartphone can connect to the wireless AP. Probe Response packet is sent out by the wireless access point to respond the request from the hosts. If the student is in the range of the access point, the Wireshark will be able to capture the Probe Request packets with their MAC address inside and the Probe Response from the wireless access point to reply the request.

The following are the filters that are used to filter the Probe Request and Probe Response:-

Probe Request: **wlan.fc.type_subtype eq 0x04 && wlan.fcs.status == good**

Probe Response: wlan.fc.type_subtype eq 0x05 && wlan.fcs.status == good

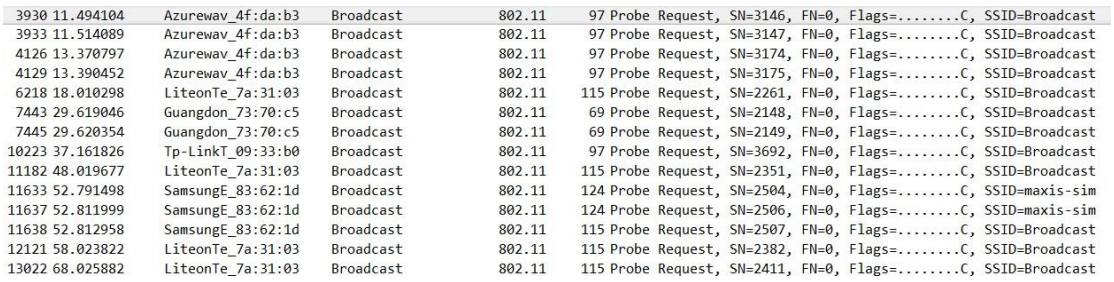
&& wlan.addr == <wireless access point MAC address>

Probe Request

"wlan.fc.type_subtype eq 0x04" specified the type of packets to be filter. 0x04 is the Probe Request. "wlan.fcs.status == good" is to check Frame Check Sequence and filter out the malformed packets.

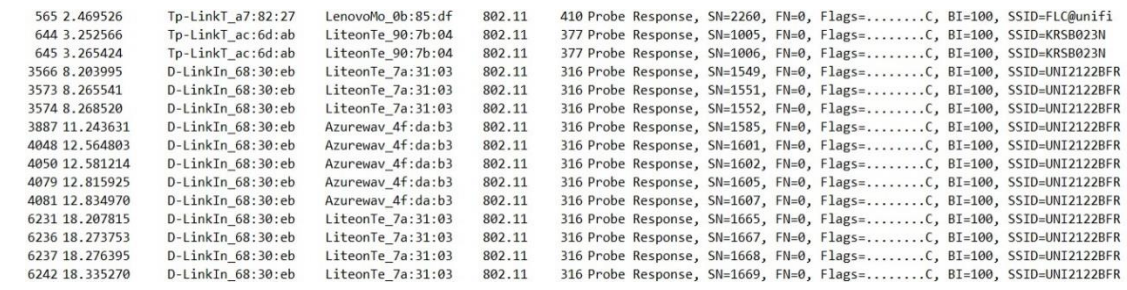
Probe Response

"wlan.fc.type_subtype eq 0x05" specified the type of packets to be filter. 0x05 is the Probe Response. "wlan.fcs.status == good" is to check Frame Check Sequence and filter out the malformed packets. The "wlan.addr == <wireless access point MAC address>" is to specify the wireless access point that sent out the response to ensure only the response from the wireless access point in the classroom is captured.



3930	11.494104	Azurewav_4f:da:b3	Broadcast	802.11	97	Probe Request, SN=3146, FN=0, Flags=........C, SSID=Broadcast
3933	11.514089	Azurewav_4f:da:b3	Broadcast	802.11	97	Probe Request, SN=3147, FN=0, Flags=........C, SSID=Broadcast
4126	13.370797	Azurewav_4f:da:b3	Broadcast	802.11	97	Probe Request, SN=3174, FN=0, Flags=........C, SSID=Broadcast
4129	13.390452	Azurewav_4f:da:b3	Broadcast	802.11	97	Probe Request, SN=3175, FN=0, Flags=........C, SSID=Broadcast
6218	18.010298	LiteonTe_7a:31:03	Broadcast	802.11	115	Probe Request, SN=2261, FN=0, Flags=........C, SSID=Broadcast
7443	29.619046	Guangdon_73:70:c5	Broadcast	802.11	69	Probe Request, SN=2148, FN=0, Flags=........C, SSID=Broadcast
7445	29.620354	Guangdon_73:70:c5	Broadcast	802.11	69	Probe Request, SN=2149, FN=0, Flags=........C, SSID=Broadcast
10223	37.161826	Tp-LinkT_09:33:b0	Broadcast	802.11	97	Probe Request, SN=3692, FN=0, Flags=........C, SSID=Broadcast
11182	48.019677	LiteonTe_7a:31:03	Broadcast	802.11	115	Probe Request, SN=2351, FN=0, Flags=........C, SSID=Broadcast
11633	52.791498	SamsungE_83:62:1d	Broadcast	802.11	124	Probe Request, SN=2504, FN=0, Flags=........C, SSID=maxis-sim
11637	52.811999	SamsungE_83:62:1d	Broadcast	802.11	124	Probe Request, SN=2506, FN=0, Flags=........C, SSID=maxis-sim
11638	52.812958	SamsungE_83:62:1d	Broadcast	802.11	115	Probe Request, SN=2507, FN=0, Flags=........C, SSID=Broadcast
12121	58.023822	LiteonTe_7a:31:03	Broadcast	802.11	115	Probe Request, SN=2382, FN=0, Flags=........C, SSID=Broadcast
13022	68.025882	LiteonTe_7a:31:03	Broadcast	802.11	115	Probe Request, SN=2411, FN=0, Flags=........C, SSID=Broadcast

Figure 3.4.2 Probe Request Frames



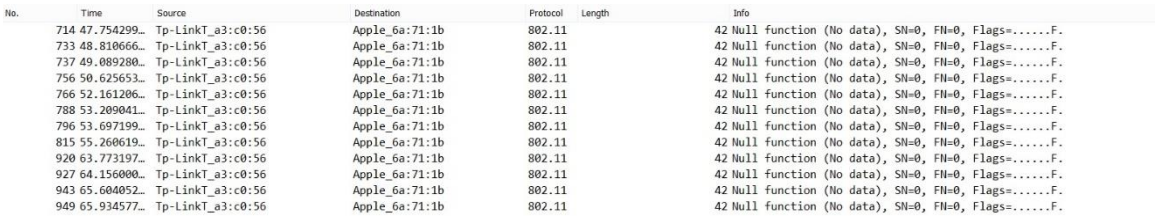
565	2.469526	Tp-LinkT_a7:82:27	LenovoMo_0b:85:df	802.11	410	Probe Response, SN=2260, FN=0, Flags=........C, BI=100, SSID=FLC@unifi
644	3.252566	Tp-LinkT_ac:6d:ab	LiteonTe_90:7b:04	802.11	377	Probe Response, SN=1005, FN=0, Flags=........C, BI=100, SSID=KRSB023N
645	3.265424	Tp-LinkT_ac:6d:ab	LiteonTe_90:7b:04	802.11	377	Probe Response, SN=1006, FN=0, Flags=........C, BI=100, SSID=KRSB023N
3566	8.203995	D-LinkIn_68:30:eb	LiteonTe_7a:31:03	802.11	316	Probe Response, SN=1549, FN=0, Flags=........C, BI=100, SSID=UNI2122BFR
3573	8.265541	D-LinkIn_68:30:eb	LiteonTe_7a:31:03	802.11	316	Probe Response, SN=1551, FN=0, Flags=........C, BI=100, SSID=UNI2122BFR
3574	8.268520	D-LinkIn_68:30:eb	LiteonTe_7a:31:03	802.11	316	Probe Response, SN=1552, FN=0, Flags=........C, BI=100, SSID=UNI2122BFR
3887	11.243631	D-LinkIn_68:30:eb	Azurewav_4f:da:b3	802.11	316	Probe Response, SN=1585, FN=0, Flags=........C, BI=100, SSID=UNI2122BFR
4048	12.564803	D-LinkIn_68:30:eb	Azurewav_4f:da:b3	802.11	316	Probe Response, SN=1601, FN=0, Flags=........C, BI=100, SSID=UNI2122BFR
4050	12.581214	D-LinkIn_68:30:eb	Azurewav_4f:da:b3	802.11	316	Probe Response, SN=1602, FN=0, Flags=........C, BI=100, SSID=UNI2122BFR
4079	12.815925	D-LinkIn_68:30:eb	Azurewav_4f:da:b3	802.11	316	Probe Response, SN=1605, FN=0, Flags=........C, BI=100, SSID=UNI2122BFR
4081	12.834970	D-LinkIn_68:30:eb	Azurewav_4f:da:b3	802.11	316	Probe Response, SN=1607, FN=0, Flags=........C, BI=100, SSID=UNI2122BFR
6231	18.207815	D-LinkIn_68:30:eb	LiteonTe_7a:31:03	802.11	316	Probe Response, SN=1665, FN=0, Flags=........C, BI=100, SSID=UNI2122BFR
6236	18.273753	D-LinkIn_68:30:eb	LiteonTe_7a:31:03	802.11	316	Probe Response, SN=1667, FN=0, Flags=........C, BI=100, SSID=UNI2122BFR
6237	18.276395	D-LinkIn_68:30:eb	LiteonTe_7a:31:03	802.11	316	Probe Response, SN=1668, FN=0, Flags=........C, BI=100, SSID=UNI2122BFR
6242	18.335270	D-LinkIn_68:30:eb	LiteonTe_7a:31:03	802.11	316	Probe Response, SN=1669, FN=0, Flags=........C, BI=100, SSID=UNI2122BFR

Figure 3.4.3 Probe Response Frames

After capturing and filtering the packets, the packets will be export as a text file to the PC. The text file will be import into the main program for MAC address extraction.

Null Data Frames

Null data frames are mostly used for power management, channel scanning and association keeping alive in the WLAN. It is has lightweight frame format (Gu, W. et al, 2008). Null data frames will be used in this system to capture the association keep alive frames of the devices that are connected to the AP in the classroom. "wlan.fc.type_subtype eq 0x0024" can be used in Wireshark to filter the Null Data Frames.



No.	Time	Source	Destination	Protocol	Length	Info
714	47.754299…	Tp-LinkT_a3:c0:56	Apple_6a:71:1b	802.11	42	Null function (No data), SN=0, FN=0, Flags=.....F.
733	48.810666…	Tp-LinkT_a3:c0:56	Apple_6a:71:1b	802.11	42	Null function (No data), SN=0, FN=0, Flags=.....F.
737	49.089280…	Tp-LinkT_a3:c0:56	Apple_6a:71:1b	802.11	42	Null function (No data), SN=0, FN=0, Flags=.....F.
756	50.625653…	Tp-LinkT_a3:c0:56	Apple_6a:71:1b	802.11	42	Null function (No data), SN=0, FN=0, Flags=.....F.
766	52.161206…	Tp-LinkT_a3:c0:56	Apple_6a:71:1b	802.11	42	Null function (No data), SN=0, FN=0, Flags=.....F.
788	53.209041…	Tp-LinkT_a3:c0:56	Apple_6a:71:1b	802.11	42	Null function (No data), SN=0, FN=0, Flags=.....F.
796	53.697199…	Tp-LinkT_a3:c0:56	Apple_6a:71:1b	802.11	42	Null function (No data), SN=0, FN=0, Flags=.....F.
815	55.260619…	Tp-LinkT_a3:c0:56	Apple_6a:71:1b	802.11	42	Null function (No data), SN=0, FN=0, Flags=.....F.
920	63.773197…	Tp-LinkT_a3:c0:56	Apple_6a:71:1b	802.11	42	Null function (No data), SN=0, FN=0, Flags=.....F.
927	64.156000…	Tp-LinkT_a3:c0:56	Apple_6a:71:1b	802.11	42	Null function (No data), SN=0, FN=0, Flags=.....F.
943	65.604052…	Tp-LinkT_a3:c0:56	Apple_6a:71:1b	802.11	42	Null function (No data), SN=0, FN=0, Flags=.....F.
949	65.934577…	Tp-LinkT_a3:c0:56	Apple_6a:71:1b	802.11	42	Null function (No data), SN=0, FN=0, Flags=.....F.

Figure 3.4.4 Null Data Frames

3.5 Main Program

There will be a few algorithms that will be included in the main program. The main program is developed in Java programming language.

3.5.1 Student ID to MAC Address Binding Algorithm

This algorithm is use to bind the student ID of the specific student with their respective MAC address of their smartphone. This algorithm will be used in the first class of the course. This algorithm can obtain the student ID-to-MAC mapping and stored in the database for future reference. This algorithm can also act as a registration of students in the course. In the beginning of the semester, we will launch this algorithm to collect the information about the student and bind them with their respective MAC address of their smartphone. This period of time will be our data preprocessing period.

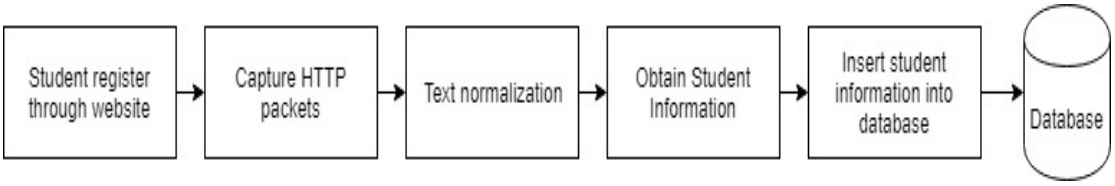


Figure 3.5.1.1 Student ID to MAC Binding Algorithm

In this algorithm, the students will have to connect their device to the wireless access point in the classroom so that the students are able to access the registration webpage hosted at the computer in the classroom. The webpage is a simple webpage that required the name, student ID, course and email address of the student. In the host computer, WAMP server will be used to host the webpage at the local host of the host computer. The students can access the IP address of the host computer to access the webpage. When the student enter their information and submit, the Wireshark will capture the HTTP packets with POST method. All the relevant information such as name, student ID, course and email will then available in the packets. Additionally, the MAC address of the students' smartphone will also available in the same packet. Hence, the algorithm is able to extract all the information and bind the Student ID with the MAC address of the smartphone that is available in the same HTTP packet captured by the Wireshark.

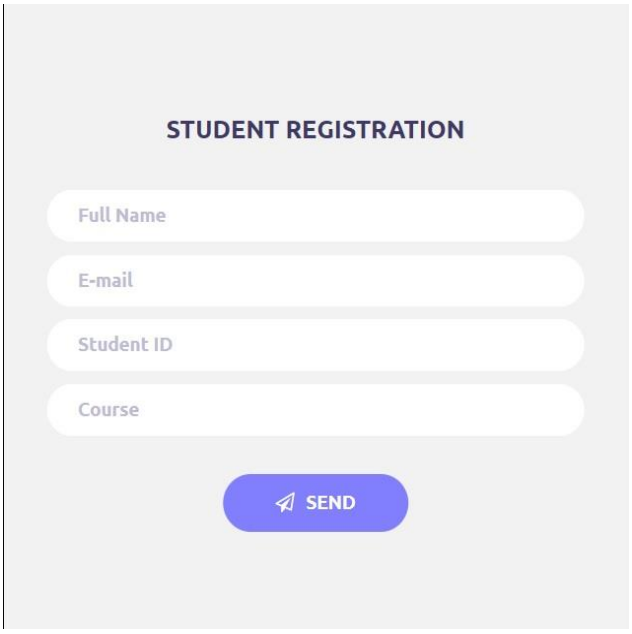


Figure 3.5.1.2 Registration Webpage

The above figure shows the interface of the webpage. It is a simple webpage that enable the students to enter their full name, email, student ID and Course.

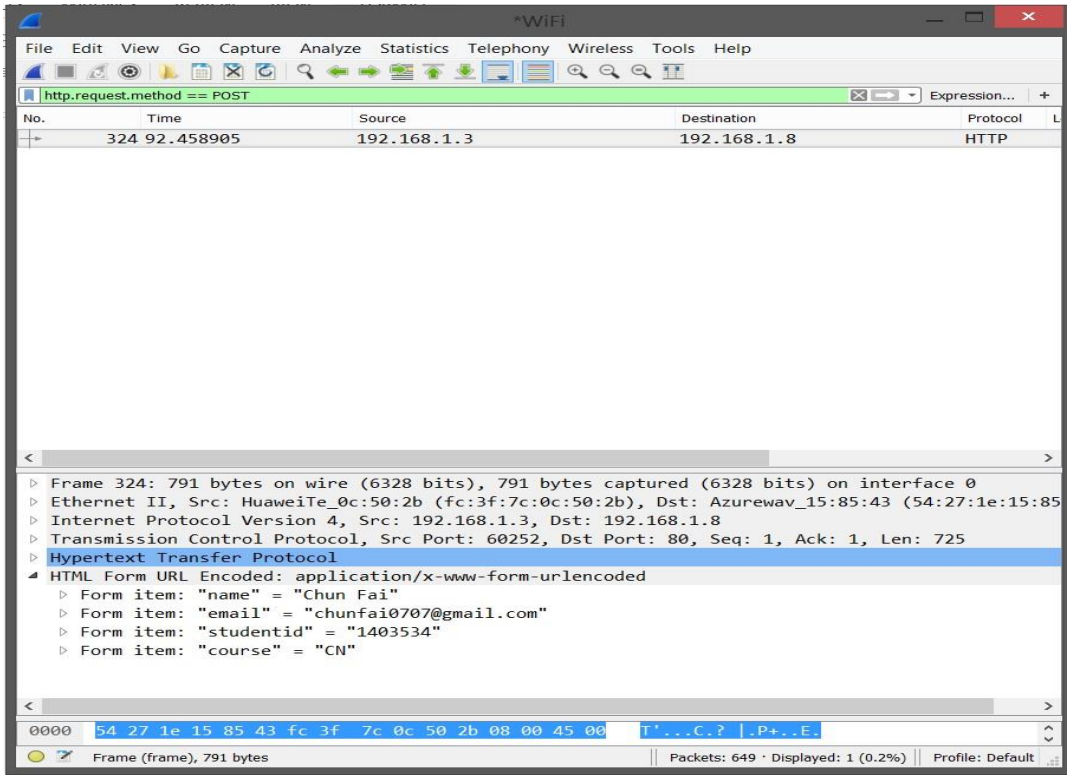


Figure 3.5.1.3 HTTP packet capture

After the student fill in all the relevant information in the webpage and click “Send” button, Wireshark will then capture the HTTP packets submitted by the particular student. As shown in the figure above, all the information in displayed in the packets such as the name, email, student ID and course. The MAC address of the source is also available in the packets and hence the algorithm will extract all the information from the text file exported from Wireshark and bind the Student ID with their respective MAC Address.

```
public static ArrayList<Student> studentToMacBinding(String file) throws FileNotFoundException, IOException {

    FileReader fileName = new FileReader(file);
    BufferedReader br = new BufferedReader(fileName);

    String line = "";
    String credential = "Ethernet";
    String str = "";

    ArrayList<Student> list = new ArrayList<>();
    ArrayList<String> nameList = new ArrayList<>();
    ArrayList<Integer> idList = new ArrayList<>();
    ArrayList<String> courseList = new ArrayList<>();
    ArrayList<String> macList = new ArrayList<>();
    ArrayList<String> emailList = new ArrayList<>();

    while (br.ready()) {

        line = br.readLine();

        if (line.length() > 0) {

            str = line.substring(0, 8);

            if (str.compareTo(credential) == 0) {
                macList.add(line.substring(line.indexOf("(") + 1, line.indexOf(")")));
            } else if (line.substring(16, 20).compareTo("name") == 0) {
                nameList.add(line.substring(line.indexOf("=") + 3, line.lastIndexOf("\"")));
            } else if (line.substring(16, 21).compareTo("email") == 0) {
                emailList.add(line.substring(line.indexOf("=") + 3, line.lastIndexOf("\"")));
            } else if (line.substring(16, 25).compareTo("studentid") == 0) {
                idList.add(Integer.parseInt(line.substring(line.indexOf("=") + 3, line.lastIndexOf("\""))));
            } else if (line.substring(16, 22).compareTo("course") == 0) {
                courseList.add(line.substring(line.indexOf("=") + 3, line.lastIndexOf("\"")));
            }
        }
    }

    for (int i = 0; i < idList.size(); i++) {
        Student student = new Student();

        student.setName(nameList.get(i));
        student.setCourse(courseList.get(i));
        student.setEmail(emailList.get(i));
        student.setMac(macList.get(i));
        student.setStudentID(idList.get(i));

        list.add(student);
    }

    return list;
}
```

Figure 3.5.1.4 Source code of Student ID to MAC Binding Algorithm

Based on the source code of the algorithm above, the algorithm will first open the text file exported from Wireshark. A few different list is created to store the information extracted from the text file. The algorithm will then look for the word “Ethernet” and extract the MAC address and store the MAC address into the MAC address list. Next, the algorithm will then look for the students’ information by searching for the keyword such as “name”, “email”, “course” and “studentid”. The extracted information will be stored into different list according to their respective types. After all the information is extracted from the text file, the algorithm will then start allocate the information

into the Student object. The Student object will then have their name, student ID, course, email address and their MAC address. The Student object will then be inserted into the Student list.

```
public static void insertStudent(ArrayList<Student> list) {
    int count = 0;
    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/finalyearproject", "root", "mysql");

        for (int i = 0; i < list.size(); i++) {
            PreparedStatement stmt = con.prepareStatement("INSERT INTO student VALUES (?,?,?,?,?)");
            stmt.setInt(1, list.get(i).getStudentID());
            stmt.setString(2, list.get(i).getName());
            stmt.setString(3, list.get(i).getEmail());
            stmt.setString(4, list.get(i).getCourse());
            stmt.setString(5, list.get(i).getMac());
            stmt.executeUpdate();
            count++;

        }
        System.out.println(count + " records inserted");
        con.close();

    } catch (Exception e) {
        System.out.println(e);
    }

}
```

Figure 3.5.1.5 Insert student into database

Next, the Student object list will be passed to the insertStudent function to insert all the student and their information into the database. The algorithm will first get connection with the database and each Student object in the list will be taken out and insert the information in the object into the database.

studentid	name	email	course	mac
1403534	Low Chun Fai	chunfai0707@gmail.com	CN	fc:3f:7c:0c:50:2b
1406675	Jasmine Wong	wong@gmail.com	CN	54:27:1e:15:85:43
1502787	Lai Tzi Yan	cherrv97@gmail.com	AC	24:e3:14:6a:71:1b
1506493	Simon Lee	lee@gmail.com	CT	1c:7b:21:57:20:9f
1602584	Beniamin	ben@gmail.com	CS	38:2c:4a:4f:40:ab

Figure 3.5.1.6 Student information inserted into database

The information inserted into the database can be seen in the MySQL workbench. After executing this algorithm, the database of student ID to MAC address binding can be done dynamically without having to bind each student ID and MAC address of their smartphone manually. This algorithm also can serve as a registration process for the student for that particular class.

3.5.2 MAC Address Extraction Algorithm

After extracting the required packets from Wireshark, the text file will be pass into the main program and the MAC Address Extraction Algorithm will be used to extract the MAC address from the text file through text mining. The algorithm will read the text file and find for the specific keyword of the line, and extract the MAC address from the specific line. The algorithm will also eliminate the redundancy of the MAC address and form a list of unique MAC address.

Figure 3.5.2.1 MAC Address Extraction Algorithm

Below are the code of the algorithm:-

```
public static ArrayList<String> getSourceMac(String file) throws IOException {

    ArrayList<String> sourceMac;
    try (FileReader fileName = new FileReader(file)) {
        BufferedReader br = new BufferedReader(fileName);
        String line = "";
        String credential = "Source";
        String str = "";
        sourceMac = new ArrayList<>();
        while (br.ready()) {

            line = br.readLine();

            if (line.length() > 0) {
                str = line.substring(4, 10);

                if (str.compareTo(credential) == 0) {
                    sourceMac.add(line.substring(line.indexOf("(") + 1, line.indexOf(")")));
                }
            }
        }
        br.close();
    }
    return sourceMac;
}
```

Figure 3.5.2.2 Get Source Mac

getSourceMac() Method

This method is used to extract the source MAC address from the packets. It will traverse the text file and look for the keyword then extract the MAC address. All the MAC address will be stored into an ArrayList.

```
public static ArrayList<String> getDestMac(String file) throws IOException {

    ArrayList<String> destMac;
    try (FileReader fileName = new FileReader(file)) {
        BufferedReader br = new BufferedReader(fileName);
        String line = "";
        String credential = "Receiver";
        String str = "";
        destMac = new ArrayList<>();
        while (br.ready()) {

            line = br.readLine();

            if (line.length() > 0) {
                str = line.substring(4, 12);

                if (str.compareTo(credential) == 0) {
                    destMac.add(line.substring(line.indexOf("(") + 1, line.indexOf(")")));
                }
            }
        }
        br.close();
    }
    return destMac;
}
```

Figure 3.5.2.3 Get Destination MAC

getDestMac() Method

This method is used to extract the destination MAC address from the packets. It will traverse the text file and look for the keyword then extract the MAC address. The destination MAC Address will be stored in an ArrayList.

```
public ArrayList<String> hash(ArrayList<String> list) {
    HashSet hs = new HashSet();
    hs.addAll(list);
    list.clear();
    list.addAll(hs);
    return list;
}
```

Figure 3.5.2.4 Hash Function

hash() Method

This method is used to filter out redundant MAC address from the capture. It will return a list of unique MAC address to the program.

```
public static List<MacAddress> hashMac(List<MacAddress> list) {
    HashSet hs = new HashSet();
    hs.addAll(list);
    list.clear();
    list.addAll(hs);
    return list;
}
```

Figure 3.5.2.5 hashMac Function

The hashMac function is used to filter out redundant MAC address objects. It will return a list of unique MAC address object.

3.5.3 Cross-Checking Extracted MAC with Registered MAC Algorithm

This algorithm will be used to cross-check the extracted MAC address from the program with the registered MAC address in the database. The program will first establish a connection with the database. After connecting to the database, the program will retrieve the MAC address from the database and compare with the extracted MAC address from the program. If the MAC address from the program matches the MAC address from the database, the student will be counted as attended. If the MAC address in the database is not found in the extracted MAC address, the students will be count as absent.

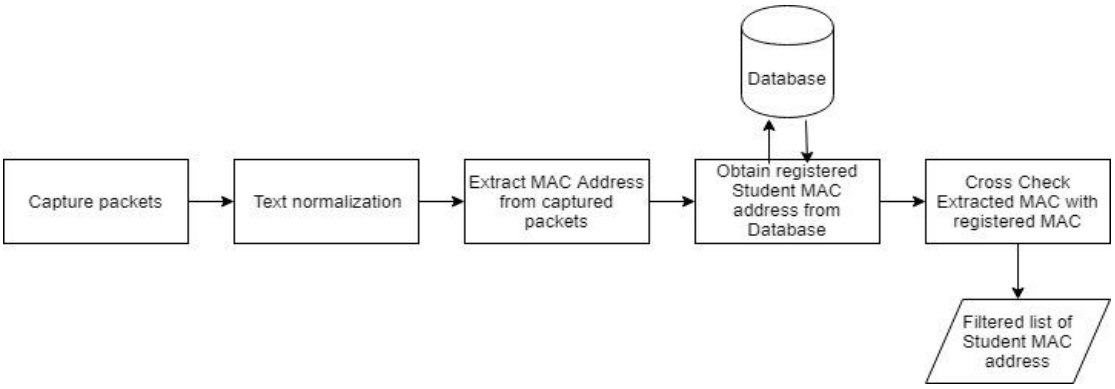


Figure 3.5.3.1 Cross Checking Algorithm

```
public ArrayList<Student> getStudent() throws SQLException {
    ArrayList<Student> list = new ArrayList<>();

    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/finalyearproject", "root", "mysql");
        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery("select * from student");
        while (rs.next()) {
            Student stu = new Student(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getString(4));
            list.add(stu);
        }
        con.close();

    } catch (ClassNotFoundException e) {
        System.out.println(e);
    }

    return list;
}
```

Figure 3.5.3.2 Retrieve Information from database

This method is used to establish connection with the database and retrieve the student's information including their MAC address to the program.

```
public static ArrayList<Student> TimestampValidation(String file) throws IOException, SQLException {

    int count = 0;
    ArrayList<List<String>> timestamp = new ArrayList<>();
    ArrayList<Integer> time = new ArrayList<>();
    time.add(0, 0);

    for (int i = 0; i < getTimeStamp(file).size(); i++) {
        time.add(i + 1, getTimeStamp(file).get(i));
    }

    for (int i = 0; i < getTimeStamp(file).size(); i++) {
        List<String> list = getSourceMac(file).subList(count, count + time.get(i + 1));
        count = count + (time.get(i + 1));
        hash(list);
        timestamp.add(list);
    }

    ArrayList<Student> stulist = getStudent();

    for (int i = 0; i < stulist.size(); i++) {
        for (int j = 0; j < timestamp.size(); j++) {
            for (int k = 0; k < timestamp.get(j).size(); k++) {
                if (stulist.get(i).getMac().equals(timestamp.get(j).get(k))) {
                    stulist.get(i).setTimeCount(stulist.get(i).getTimeCount() + 1);
                }
            }
        }
    }

    return stulist;
}
```

Figure 3.5.3.3 Timestamp Validation

The cross-checking algorithm is implemented inside the Timestamp Validation function. The function will extract the MAC address from the captured packets and obtain another list of Student with their MAC address from the database. The program will then compared the captured packets with the registered MAC address from the database and only the registered MAC address is added into another list. This can eliminate the unregistered MAC address captured by the Wireshark.

3.5.4 Anti MAC Spoofing Algorithm

As the student monitoring system is heavily depending on the MAC address of the mobile phone, it is very essential to have an algorithm to mitigate the attack such as MAC spoofing attack. This algorithm will use some unique identifier from the mobile device to generate a set of fingerprint for that particular device. The algorithm will extract the identifier from the captured packets and generate a set of fingerprint of the device and store into the database for further reference. At the following class, the algorithm will then extract the captured identifier and compared to the identifier from the database to check whether there is any MAC spoofing attack in the network.

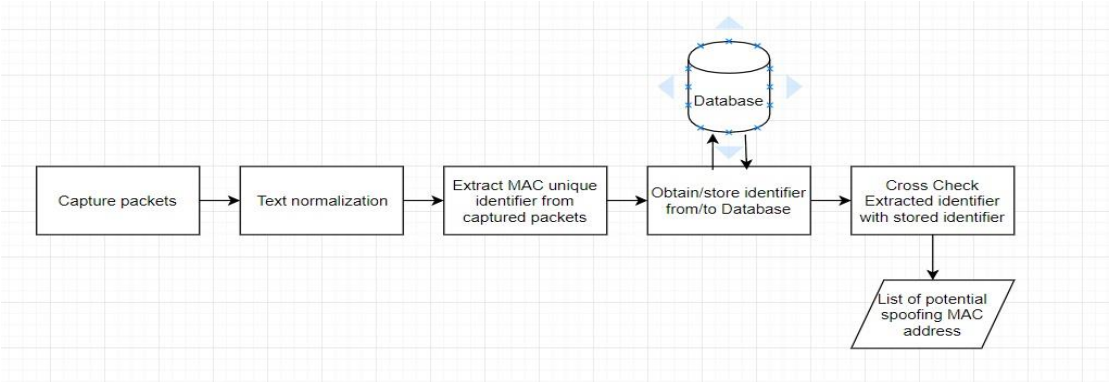


Figure 3.5.4.1 Anti MAC Spoofing Algorithm

According to Idland (2011), there are several ways to fingerprint a device. Some vendor specific extensions can be used as a source for fingerprinting a device. Some of the implicit identifiers that can be used to create a fingerprint for the device are Supported rates, Listen interval and Vendor Specific. In our system, we will use the Supported Rates, Extended Supported Rates and HT Capabilities Info as our implicit identifier to create a fingerprint for the students' mobile devices.

```
public static ArrayList<String> antiMacSpoofing(String file) throws IOException, SQLException {

    ArrayList<String> list = new ArrayList<>();

    ArrayList<MacAddress> capture = getIdentifier(file);
    ArrayList<MacAddress> identifier = getIdentifierFromDB();
    hashMac(capture);

    for (int i = 0; i < identifier.size(); i++) {
        for (int j = 0; j < capture.size(); j++) {
            if (capture.get(j).getSupportedRate() != (null) && capture.get(j).getExtendedRate() != (null) && capture.get(j) != (null)) {
                if (identifier.get(i).getMac().equals(capture.get(j).getMac())) {

                    if (identifier.get(i).getSupportedRate().equals(capture.get(j).getSupportedRate())) {
                        capture.get(j).setSpoofCheck(capture.get(j).getSpoofCheck() + 1);
                    }
                    if (identifier.get(i).getExtendedRate().equals(capture.get(j).getExtendedRate())) {
                        capture.get(j).setSpoofCheck(capture.get(j).getSpoofCheck() + 1);
                    }

                    if (identifier.get(i).getHtCapability() != null) {
                        if (identifier.get(i).getHtCapability().equals(capture.get(j).getHtCapability())) {
                            capture.get(j).setSpoofCheck(capture.get(j).getSpoofCheck() + 1);
                        }

                    } else if (identifier.get(i).getHtCapability() == null) {
                        if (capture.get(j).getHtCapability() == null) {
                            capture.get(j).setSpoofCheck(capture.get(j).getSpoofCheck() + 1);
                        }
                    }
                }
            }
        }
    }

    for (int i = 0; i < capture.size(); i++) {
        if (capture.get(i).getSpoofCheck() == 0) {
            list.add(capture.get(i).getMac());
        }
    }

    hash(list);
    return list;
}
```

Figure 3.5.4.2 Code of Anti MAC Spoofing Algorithm

The Anti MAC Spoofing algorithm will create 2 list of MAC address which is the captured MAC address with their identifier and the registered MAC address with their identifier. The algorithm will compare both MAC address in the both list to check whether there are any different in term of the unique identifier. When abnormalities or difference detected by comparing the unique identifier, it means that there are potential MAC spoofing attack happening in the network. If there are abnormalities detected by the algorithm, the affected MAC address will be added into the potential spoofing list.

```
public static ArrayList<MacAddress> getIdentifier(String file) throws FileNotFoundException, IOException {

    FileReader fileName = new FileReader(file);
    BufferedReader br = new BufferedReader(fileName);

    String line = "";
    String credential = "Source";
    String str = "";

    ArrayList<MacAddress> list = new ArrayList<>();

    int count = 0;

    while (br.ready()) {

        line = br.readLine();

        if (line.length() > 0) {

            str = line.substring(4, 10);

            if (str.compareTo(credential) == 0) {
                MacAddress mac = new MacAddress();
                mac.setMac(line.substring(line.indexOf("(") + 1, line.indexOf(")")));
                list.add(mac);
                count++;

            } else if (line.length() > 30 && line.substring(13, 22).compareTo("Supported") == 0) {
                String support = line.substring(line.indexOf("Rates") + 6, line.lastIndexOf(","));
                list.get(count - 1).setSupportedRate(support);

            } else if (line.length() > 30 && line.substring(13, 31).compareTo("Extended Supported") == 0) {
                String extended = line.substring(line.indexOf("Rates") + 6, line.lastIndexOf(","));
                list.get(count - 1).setExtendedRate(extended);

            } else if (line.length() > 30 && line.substring(12, 27).compareTo("HT Capabilities") == 0) {
                String ht = line.substring(line.indexOf(":") + 2, line.length());
                list.get(count - 1).setHtCapability(ht);

            }
        }
    }
    return list;
}
```

Figure 3.5.4.3 Get Identifier

The method getIdentifier will extract the identifier from the captured packets. The method will extract the MAC address and their unique identifier and then store into a list. The unique identifier that will be extracted is the supported rate, extended supported rate and also the HT Capabilities information.

```
public static void insertIdendifier(ArrayList<MacAddress> list) throws SQLException {
    int count = 0;

    ArrayList<Student> stu = getStudent();
    ArrayList<MacAddress> macList = new ArrayList<>();

    for (int i = 0; i < stu.size(); i++) {
        MacAddress mac = new MacAddress();
        mac.setMac(stu.get(i).getMac());
        for (int j = 0; j < list.size(); j++) {
            if (mac.getMac().equals(list.get(j).getMac())) {
                mac.setSupportedRate(list.get(j).getSupportedRate());
                mac.setExtendedRate(list.get(j).getExtendedRate());
                mac.setHtCapability(list.get(j).getHtCapability());
                break;
            }
        }
        macList.add(mac);
    }

    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/finalyearproject", "root", "mysql");

        for (int i = 0; i < macList.size(); i++) {
            PreparedStatement stmt = con.prepareStatement("INSERT INTO identifier VALUES (?,?,?,?)");

            stmt.setString(1, macList.get(i).getMac());
            stmt.setString(2, macList.get(i).getSupportedRate());
            stmt.setString(3, macList.get(i).getExtendedRate());
            stmt.setString(4, macList.get(i).getHtCapability());
            stmt.executeUpdate();
            count++;

        }
        con.close();

    } catch (Exception e) {
        System.out.println(e);
    }
}
```

Figure 3.5.4.4 Insert Identifier into Database

The insertIdentifier method will take in the list of MAC address with their respective identifier. By using the Cross-Checking algorithm, the method will obtain the list of MAC address registered in the student database and filtered out the registered MAC address from the MAC address list. The filtered MAC address will be added into another list. After obtaining the filtered MAC address list, the method will then establish connection with the database the insert the MAC address with their unique identifier into the database.

```
public static ArrayList<MacAddress> getIdentifierFromDB() throws SQLException {
    ArrayList<MacAddress> list = new ArrayList<>();

    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/finalyearproject", "root", "mysql");
        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery("select * from identifier");
        while (rs.next()) {
            MacAddress mac = new MacAddress(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4));
            list.add(mac);
        }
        con.close();

    } catch (ClassNotFoundException e) {
        System.out.println(e);
    }

    return list;
}
```

Figure 3.5.4.5 Retrieve Identifier from Database

The getIdentifierFromDB method is to retrieve the registered MAC with their identifier from the database. This method is used when the algorithm needs to check the potential spoofing by comparing the registered identifier with the captured identifier.

3.5.5 Timestamp Validation Algorithm

This algorithm will be used to improve the accuracy of the system where the algorithm will check whether the student is in the classroom throughout the commencement of the class. This algorithm divide the captured packets into a few segment according to the timestamp of the packets. The packets will be divided into a 10 minutes segment. For example, if the captured duration is 60 minutes, the algorithm will divide the packets into 6 segments, 10 minutes for each segment. The main purpose of this algorithm is to act as a checkpoint in order to check whether the student is in the classroom based on each segment of the packets. If the students is in the classroom for the whole commencement of the class, the MAC address of that student will be visible in every segment of the packets. Hence, the attendance will be considered valid. On the other hand, some student leave the class halfway before the end of the class, their MAC address will not be seen in the last few segment of the packets as they are not physically in the range of the classroom. As a result, their attendance is not considered as a valid attendance.

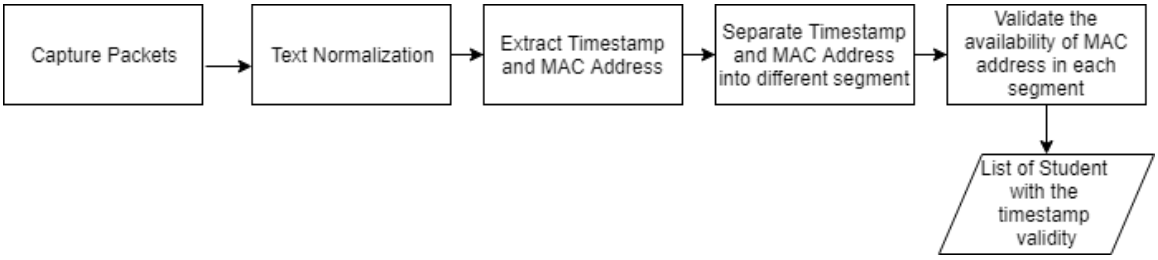


Figure 3.5.5.1 Timestamp Validation Algorithm

As shown in the above flowchart, the algorithm will input the captured packets and extract the timestamp and MAC address from the text file. The timestamp and MAC address is stored into a separate list. In this case, the MAC address redundancy is not eliminated as it needs to tally with the timestamp extracted from the text file. Next, the algorithm will separate the timestamp and the MAC address into the 10 minutes segment. The algorithm will then validate and check whether the MAC address is in each segment and lastly the percentage of validity will be calculated. The percentage of validity needs to be over 70 % to be considered a valid attendance.

```
public static ArrayList<Integer> getTimeStamp(String file) throws FileNotFoundException, IOException {

    FileReader fileName = new FileReader(file);
    BufferedReader br = new BufferedReader(fileName);

    String line = "";
    String credential = "[Time since";
    String str = "";

    ArrayList<Integer> timestamp = new ArrayList<>();
    ArrayList<Double> time = new ArrayList<>();
    ArrayList<List<Double>> timelist = new ArrayList<>();

    while (br.ready()) {

        line = br.readLine();

        if (line.length() > 0) {
            str = line.substring(4, 15);

            if (str.compareTo(credential) == 0) {
                String s = line.substring(line.indexOf(":") + 1, line.indexOf("seconds") - 1);
                Double d = Double.parseDouble(s);
                time.add(d);
            }
        }
    }

    for (int i = 0; i < ((time.get(time.size() - 1)) / 600); i++) {
        ArrayList<Double> time1 = new ArrayList<>();

        for (int j = 0; j < time.size() - 1; j++) {
            if (time.get(j) >= (600.00 * i) && time.get(j) <= (600.00 * (i + 1))) {
                time1.add(time.get(j));
            }
        }

        timelist.add(time1);
    }

    for (int i = 0; i < timelist.size(); i++) {
        timestamp.add(timelist.get(i).size());
    }

    return timestamp;

}
```

Figure 3.5.5.2 Get Timestamp

Based on the algorithm above, the system will first open the text file extracted from the Wireshark and look for the specific keyword to extract the information from the specific line. Similar with the previous algorithm, this algorithm also created a few list to store the timestamp. Next, the algorithm will divide the timestamp into different segment, each segment will be 600 seconds, which is 10 minutes. A list that calculate the number of timestamp in each segment is also included in the function. The list that contains the total number of timestamp in each segment will be returned in this function.

```
public static ArrayList<Student> TimestampValidation(String file) throws IOException, SQLException {

    int count = 0;
    ArrayList<List<String>> timestamp = new ArrayList<>();
    ArrayList<Integer> time = new ArrayList<>();
    time.add(0, 0);

    for (int i = 0; i < getTimeStamp(file).size(); i++) {
        time.add(i + 1, getTimeStamp(file).get(i));
    }

    for (int i = 0; i < getTimeStamp(file).size(); i++) {
        List<String> list = getSourceMac(file).subList(count, count + time.get(i + 1));
        count = count + (time.get(i + 1));
        hash(list);
        timestamp.add(list);
    }

    ArrayList<Student> stulist = getStudent();

    for (int i = 0; i < stulist.size(); i++) {
        for (int j = 0; j < timestamp.size(); j++) {
            for (int k = 0; k < timestamp.get(j).size(); k++) {
                if (stulist.get(i).getMac().equals(timestamp.get(j).get(k))) {
                    stulist.get(i).setTimeCount(stulist.get(i).getTimeCount() + 1);
                }
            }
        }
    }

    return stulist;
}
```

Figure 3.5.5.3 Timestamp Validation

In this algorithm, the program will apply the getTimeStamp function to get the list of MAC address and compared with the student MAC address in the database. The program will check whether the students' MAC address is available in each 10 minute segment. If the students' MAC is found in the segment, the number of count will be increases by 1. At the end of this algorithm, the function will return a list of Student with the total number of count to indicate the percentage of availability in the classroom and further determine the validation of the attendance.

3.5.6 Attendance Validation Algorithm

The attendance validation algorithm is used to validate the attendance of the students by using both the Timestamp Validation algorithm and also the Anti MAC Spoofing Algorithm. This algorithm will be used at the end of the system. The algorithm will check for the timestamp validation of the student's MAC address and run the next check to detect any potential spoofing attack by that particular MAC address. If the MAC address passed both of the validation algorithm, the student will be granted a validated attendance. Otherwise, the student will be marked as "Absent" if their MAC address failed the validation test. After the validation process, the attendance will be updated into the database.

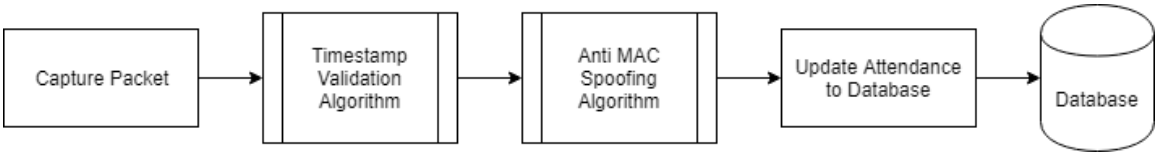


Figure 3.5.6.1 Attendance Validation Algorithm

```
public static ArrayList<Student> validateAttendance(ArrayList<Student> list, ArrayList<String> spoof, String file) throws IOException {

    String attendance = "";
    double percentage = 0.00;
    ArrayList<Student> stulist = new ArrayList<>();

    for (int i = 0; i < list.size(); i++) {

        percentage = (list.get(i).getTimeCount() / ((double) getTimeStamp(file).size())) * (100);
        if (percentage >= 70) {
            attendance = "Attended";
        } else {
            attendance = "Absent";
        }

        for (int j = 0; j < spoof.size(); j++) {
            if (list.get(i).getMac().equals(spoof.get(j))) {
                attendance = "Potential Spoofing Detected! Attendance is Void.";
                break;
            }
        }

        for (Iterator<Student> it = list.iterator(); it.hasNext();) {
            Student list1 = it.next();
            Student stu = new Student();
            stu.setName(list.get(i).getName());
            stu.setStudentID(list.get(i).getStudentID());
            stu.setCourse(list.get(i).getCourse());
            stu.setAttendance(attendance);
            stulist.add(stu);
        }
    }

    return stulist;
}
```

Figure 3.5.6.2 Validation of Attendance

In the validateAttendance algorithm, the algorithm will take in 2 list which is the list of Student from the Timestamp Validation algorithm and the other list from the Anti MAC Spoofing Algorithm. First, the system will extract the timestamp validation information from the list from Timestamp Validation algorithm. If the Timestamp Validation of the student is over 70%, the student will be given an "Attended" to indicate that they attended the class and is in the class throughout the class. Otherwise, the student will be given an "Absent" if their Timestamp Validation is less than 70% as the system does not capture any packets from that particular student for a certain time interval. Next, the algorithm will proceed to the next validation, which is the Anti Spoofing algorithm. The algorithm will check whether the MAC address of the students in the list matches the potential spoofing MAC address list from the Anti MAC Spoofing algorithm. If they are matching in the MAC address, the MAC address will be marked as potentially spoofed and attendance will not be given to the student. After both validation is completed, the information will be updated into the list of Student and pass to the next algorithm.

```
public static void updateAttendance(ArrayList<Student> stulist) throws SQLException, ClassNotFoundException {

    ArrayList<Student> list = getAttendanceList();

    double percentage;

    for (int i = 0; i < list.size(); i++) {
        list.get(i).setTotalclass(list.get(i).getTotalclass() + 1);

        for (int j = 0; j < stulist.size(); j++) {
            if (list.get(i).getStudentID() == stulist.get(j).getStudentID()) {
                if (stulist.get(j).getAttendance().equals("Attended")) {
                    list.get(i).setCount(list.get(i).getCount() + 1);
                    break;
                }
            }
        }
    }

    for (int i = 0; i < list.size(); i++) {
        percentage = ((double) list.get(i).getCount() / (double) (list.get(i).getTotalclass()) * 100);
        list.get(i).setPercentage(percentage);
    }

    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/finalyearproject", "root", "mysql");

        for (int i = 0; i < list.size(); i++) {
            PreparedStatement stmt = con.prepareStatement("UPDATE attendance SET attendancepercent=?, attendance=?, totalclass=? WHERE studentid=?");
            stmt.setDouble(1, list.get(i).getPercentage());
            stmt.setInt(2, list.get(i).getCount());
            stmt.setInt(3, list.get(i).getTotalclass());
            stmt.setInt(4, list.get(i).getStudentID());
            stmt.executeUpdate();

        }

    } catch (ClassNotFoundException | SQLException e) {
        System.out.println(e);
    }

}
```

Figure 3.5.6.3 Update Attendance to Database

For the update Attendance algorithm, the algorithm will take in the list of Student from the Validate Attendance algorithm. The algorithm will update the total number of class into the database every time the algorithm is invoked. Next, the algorithm will check the attendance information in the student list. If the student has an “Attended”, their total number of class attended will be increase. Otherwise, the total number of class attended will not be increases if they have an “Absent” or “Potentially Spoofed”. After that, the algorithm will calculate the total percentage of the attendance by dividing the total attended class with the total number of class. Lastly, the information will be updated into the database.

3.6 Main Graphic User Interface

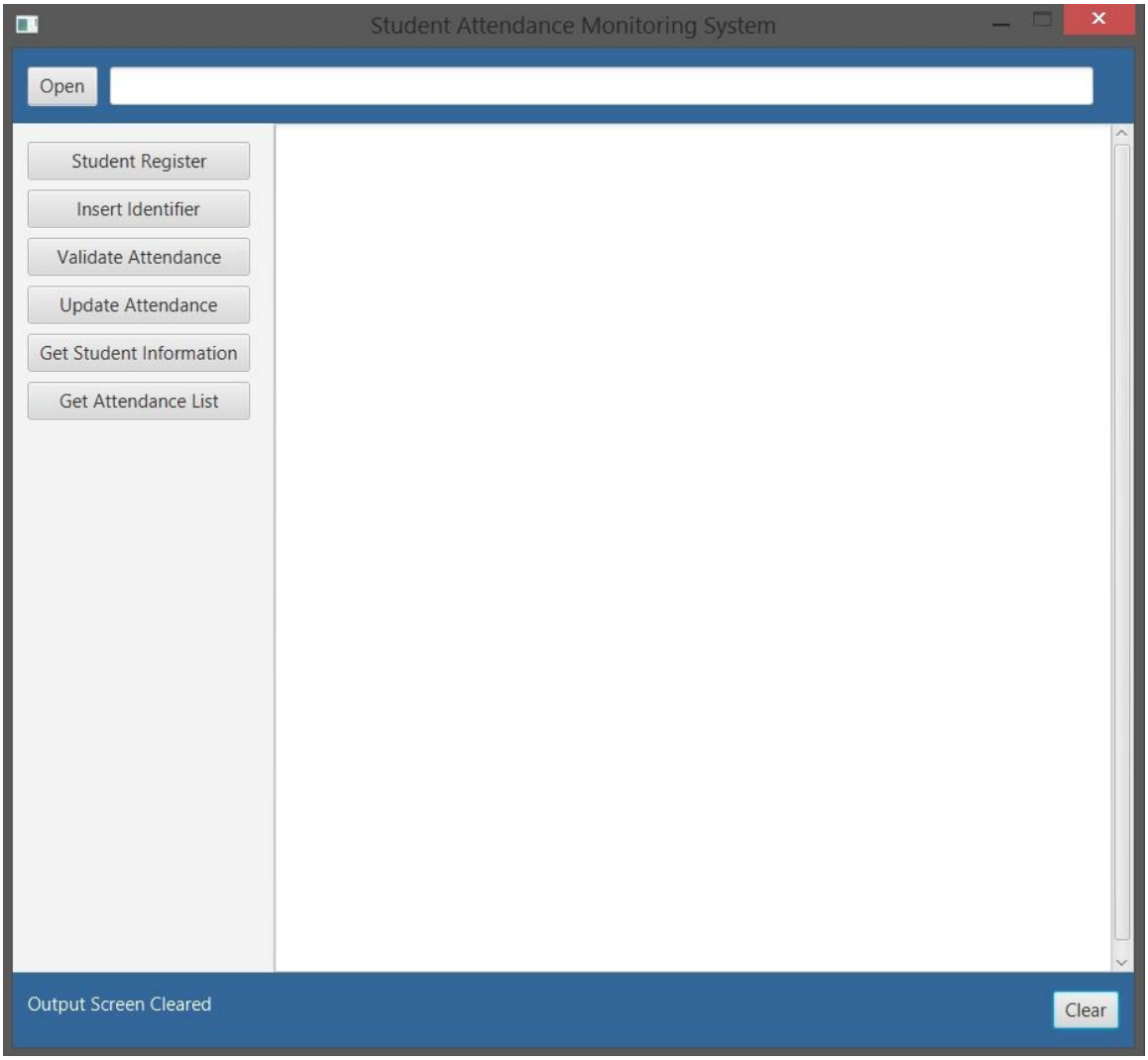


Figure 3.6.1 GUI of the System

Above diagram shows the Main Graphic User Interface of the system. There are a few button which are “Open”, “Student Register”, “Insert Identifier”, “Validate Attendance”, “Update Attendance”, “Get Student Information”, Get Attendance List” and “Clear”. Each button will invoke the specific algorithm from the Main Program and the output will be displayed at the blank area of the system. The function of each button will be discussed in the next section.

```
@Override
public void start(Stage primaryStage) throws IOException, SQLException {

    primaryStage.setTitle("Student Attendance Monitoring System");
    FileChooser fileChooser = new FileChooser();

    Button openButton = new Button("Open");
    Button registerButton = new Button("Student Register");
    Button retrieveStudent = new Button("Get Student Information");
    Button validate = new Button("Validate Attendance");
    Button clear = new Button("Clear");
    Button identifier = new Button("Insert Identifier");
    Button update = new Button("Update Attendance");
    Button getAttendance = new Button("Get Attendance List");

    retrieveStudent.setMaxWidth(200);
    registerButton.setMaxWidth(200);
    validate.setMaxWidth(200);
    identifier.setMaxWidth(200);
    update.setMaxWidth(200);
    getAttendance.setMaxWidth(200);

    Text progress = new Text("");
    progress.setFill(Color.WHITE);
    progress.prefWidth(800);
    TextField textField = new TextField();
    textField.setPrefWidth(800);

    TextArea textArea = TextAreaBuilder.create()
            .prefWidth(400)
            .prefHeight(700)
            .wrapText(true)
            .build();
    textArea.setEditable(false);

    ScrollPane scrollPane = new ScrollPane();
    scrollPane.setContent(textArea);
    scrollPane.setFitToWidth(true);
    scrollPane.setPrefWidth(700);
    scrollPane.setPrefHeight(700);

    HBox hbox = new HBox();
    hbox.setPadding(new Insets(15, 12, 15, 12));
    hbox.setSpacing(10);
    hbox.setStyle("-fx-background-color: #336699;");
    hbox.getChildren().addAll(openButton, textField);
```

```
HBox hbox2 = new HBox();
hbox2.setPadding(new Insets(15, 12, 15, 12));
hbox2.setSpacing(8);

final Pane spacer = new Pane();
HBox.setHgrow(spacer, Priority.ALWAYS);
spacer.setMinSize(10, 1);
hbox2.setStyle("-fx-background-color: #336699;");
hbox2.getChildren().addAll(progress, spacer, clear);

VBox vbox2 = new VBox();
vbox2.setPadding(new Insets(15, 12, 15, 12));
vbox2.setSpacing(8);
vbox2.getChildren().addAll(registerButton, identifier, validate, update, retrieveStudent, getAttendance);
vbox2.setFillWidth(true);

BorderPane border = new BorderPane();
border.setTop(hbox);
border.setBottom(hbox2);
border.setLeft(vbox2);
border.setRight(scrollPane);
border.setCenter(null);

primaryStage.setScene(new Scene(border, 900, 800));
primaryStage.setResizable(false);
primaryStage.show();
```

Figure 3.6.2 Code of GUI

Above are the code of the GUI interface. The GUI is developed using JavaFX. First, the title of the system is set to "Student Attendance Monitoring System. A few button is declared which are the Open button, Student Register button, Get Student Information button, Insert Identifier button, Update Attendance, Get Attendance List button, Validate Attendance button as well as the Clear button. The 3 button on the left is set to the max width of 200px. A progress indicator is also created to show the progress of the system. Next, a text area is created to display all the information on the GUI when the button is clicked.

3.6.1 Open Button

```
openButton.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(final ActionEvent e) {
        File file = fileChooser.showOpenDialog(primaryStage);
        if (file != null) {
            fileLocation = file.getAbsolutePath();
            textField.setText(fileLocation);
        }
    }
});
```

Figure 3.6.1.1 Code of Open Button

The open button will prompt a windows for the user to select the input file they want to input to the system. The chosen file will have their absolute path extracted and input as a String into the algorithm in the Main Program.

3.6.2 Clear Button

```
clear.setOnAction(new EventHandler<ActionEvent>() {

    @Override
    public void handle(final ActionEvent e) {
        textArea.setText("");
        progress.setText("Output Screen Cleared");
    }
});
```

Figure 3.6.2.1 Code of Clear Button

The clear button will set the text area into empty and hence clear the output screen when clicked.

3.6.3 Student Register Button

```
registerButton.setOnAction(new EventHandler<ActionEvent>() {
    ArrayList<Student> stu = new ArrayList<>();

    @Override
    public void handle(ActionEvent event) {
        textArea.setText("");
        if (fileLocation.isEmpty()) {
            progress.setText("Please Select an input file.");
        } else {

            try {
                stu = MainProgram.studentToMacBinding(fileLocation);
            } catch (IOException ex) {
                progress.setText("Input Output Error");
            }
        }
        try{
        MainProgram.insertStudent(stu);

        } catch (Exception ex){
            progress.setText(ex.toString());
        }


    }

});
```

Figure 3.6.3.1 Code of Student Register Button

The student registration button will use the studentToMacBinding algorithm in the Main Program to bind the student ID with their respective MAC address by extracting all the information from the input file. After extracting the information, the program will call the insertStudent method to insert the information into the database. The student information will showed on the output screen as well as the total number of data inserted into the database.

3.6.4 Insert Identifier Button

```
identifier.setOnAction(new EventHandler<ActionEvent>() {
    ArrayList<MacAddress> mac = new ArrayList<>();
    ArrayList<MacAddress> mac2 = new ArrayList<>();

    @Override
    public void handle(ActionEvent event) {
        textArea.setText("");
        if (fileLocation.isEmpty()) {
            progress.setText("Please Select an input file.");
        } else {
            try {
                try {
                    mac = MainProgram.getIdentifier(fileLocation);
                } catch (IOException ex) {
                }

                MainProgram.insertIdentifier(mac);

                mac2 = MainProgram.getIdentifierFromDB();

                for (int i = 0; i < mac2.size(); i++) {
                    textArea.setText(textArea.getText() + "Mac Address: " + mac2.get(i).getMac() + "\n"
                            + "Supported Rate: " + mac2.get(i).getSupportedRate() + "\n"
                            + "Extended Rate: " + mac2.get(i).getExtendedRate() + "\n"
                            + "HT Capabilities: " + mac2.get(i).getHtCapability() + "\n\n"
                    );

                    progress.setText("Attendance Validation Done!");
                }

                progress.setText("Mac Uniqie identifier inserted successfully.");
            } catch (SQLException ex) {
                progress.setText("SQL error.");
            }
        }
    }
```

Figure 3.6.4.1 Code of Insert Identifier Button

The Insert Identifier button will invoke the getIdentifier method from the Main Program. The method will extract all the unique identifier of the MAC address from the input file. The identifier and the MAC address will be stored inside a list. Next, the program will pass the list of MAC address to the insertIdentifier method to insert the identifier into the database. Lastly, the program will call the getIdentifierFromDB method to retrieve all the information from database and display them to the screen.

3.6.5 Validate Attendance Button

```
validate.setOnAction(new EventHandler<ActionEvent>() {

    @Override
    public void handle(ActionEvent event) {
        textArea.setText("");
        if (fileLocation.isEmpty()) {
            progress.setText("Please Select an input file.");
        } else {

            try {
                progress.setText("Attendance Validation in Progress...!");
                textArea.setText("");
                ArrayList<Student> list = MainProgram.TimestampValidation(fileLocation);
                ArrayList<String> spoof = MainProgram.antiMacSpoofing(fileLocation);
                ArrayList<Student> stulist = MainProgram.validateAttendance(list, spoof, fileLocation);

                for (int i = 0; i < stulist.size(); i++) {
                    textArea.setText(textArea.getText() + "Name: " + stulist.get(i).getName() + "\n"
                            + "Course: " + stulist.get(i).getCourse() + "\n"
                            + "Mac: " + stulist.get(i).getMac() + "\n"
                            + "Timestamp: " + df.format(stulist.get(i).getTimestamp()) + " %" + "\n"
                            + "Attendance: " + stulist.get(i).getAttendance() + "\n\n"
                    );

                    progress.setText("Attendance Validation Done!");
                }

            } catch (IOException ex) {
                progress.setText("Input/Outpur Error.");
            } catch (SQLException ex) {
                progress.setText("SQL error.");
            }

        }
    }

});
```

Figure 3.6.5.1 Code of Validate Attendance Button

By clicking the Validate Attendance button, the system will call the Validate Attendance algorithm where 2 validation which is Timestamp Validation and antiMacSpoofing algorithm will be called. The system will run the validation of the extracted information from the input file and the student information and their attendance validity will be shown on the output area.

3.6.6 Update Attendance Button

```
update.setOnAction(new EventHandler<ActionEvent>() {

    @Override
    public void handle(final ActionEvent e) {
        textArea.setText("");
        if (fileLocation.isEmpty()) {
            progress.setText("Please Select an input file.");
        } else {
            try {
                ArrayList<Student> list = MainProgram.TimestampValidation(fileLocation);
                ArrayList<String> spoof = MainProgram.antiMacSpoofing(fileLocation);
                ArrayList<Student> stulist = MainProgram.validateAttendance(list, spoof, fileLocation);

                MainProgram.updateAttendance(stulist);

                ArrayList<Student> attendance = MainProgram.getAttendanceList();

                for (int i = 0; i < attendance.size(); i++) {
                    textArea.setText(textArea.getText() + "Student ID: " + attendance.get(i).getStudentID() + "\n"
                            + "Name: " + attendance.get(i).getName() + "\n"
                            + "Course: " + attendance.get(i).getCourse() + "\n"
                            + "Attendance: " + attendance.get(i).getPercentage() + " % \n\n"
                    );
                }
                progress.setText("Attendance is updated into the database!");
            } catch (SQLException ex) {
                progress.setText("SQL error.");
            } catch (ClassNotFoundException | IOException ex) {
                progress.setText("Input/Output error.");
            }
        }
    }
});
```

Figure 3.6.6.1 Code of Update Attendance Button

After the validation of the attendance, the update attendance button can be used to update the attendance to the database. The program will first run the attendance validation algorithm and pass the student list to the updateAttendance algorithm. The attendance of the student will then updated into the database. Lastly, the program will call the getAttendanceList method to retrieve all the latest attendance information of the students to the screen.

3.6.7 Get Student Information Button

```
retrieveStudent.setOnAction(new EventHandler<ActionEvent>() {

    ArrayList<Student> list = MainProgram.getStudent();

    @Override
    public void handle(final ActionEvent e) {
        textArea.setText("");
        for (int i = 0; i < list.size(); i++) {
            textArea.setText(textArea.getText() + "Name: " + list.get(i).getName() + "\n"
                    + "Course: " + list.get(i).getCourse() + "\n"
                    + "Student ID: " + list.get(i).getStudentID() + "\n"
                    + "Email: " + list.get(i).getEmail() + "\n"
                    + "Mac: " + list.get(i).getMac() + "\n\n"
            );

            progress.setText("Student Information from Databse");
        }
    }
});
```

Figure 3.6.7.1 Code of Get Student Information Button

The Get Student Information will use the getStudent () method in the Main Program to get a list of Student from the database. Upon clicking the Get Student Information Button, the system will extract the information of the students and display them onto the output screen in the GUI.

3.6.8 Get Attendance List Button

```
getAttendance.setOnAction(new EventHandler<ActionEvent>() {

    ArrayList<Student> list = MainProgram.getAttendanceList();

    @Override
    public void handle(final ActionEvent e) {
        textArea.setText("");
        for (int i = 0; i < list.size(); i++) {
            textArea.setText(textArea.getText() + "Name: " + list.get(i).getName() + "\n"
                    + "Course: " + list.get(i).getCourse() + "\n"
                    + "Student ID: " + list.get(i).getStudentID() + "\n"
                    + "Class Attended: " + list.get(i).getCount() + "\n"
                    + "Total Class: " + list.get(i).getTotalclass() + "\n"
                    + "Attendance: " + list.get(i).getPercentage() + " % \n\n"
            );
            progress.setText("Attendance Information from Databse");
        }
    }
});
```

Figure 3.6.8.1 Code of Get Attendance List Button

The get attendance list button will call the getAttendanceList method to retrieve all the students' attendance information from the database and display to the output screen of the program.

3.7 MacAddress Class

```
public class MacAddress {

    private String mac;
    private String supportedRate;
    private String extendedRate;
    private String htCapability;
    private int spoofCheck;

    public MacAddress() {
        this.mac = null;
        this.supportedRate = null;
        this.extendedRate = null;
        this.htCapability = null;
        this.spoofCheck = 0;
    }

    public MacAddress(String mac, String supportedRate, String extendedRate, String htCapability) {
        this.mac = mac;
        this.supportedRate = supportedRate;
        this.extendedRate = extendedRate;
        this.htCapability = htCapability;
    }
```

Figure 3.7.1 Constructor

Above figure shows the constructor of the MacAddress object that consists of MAC address, supported rate, extended supported rate, HT capabilities info and the counter used in anti MAC spoofing algorithm.

```
    public String getSupportedRate() {
        return supportedRate;
    }

    public String getExtendedRate() {
        return extendedRate;
    }

    public String getHtCapability() {
        return htCapability;
    }

    public String getMac() {
        return mac;
    }

    public int getSpoofCheck() {
        return spoofCheck;
    }

    public void setSpoofCheck(int spoofCheck) {
        this.spoofCheck = spoofCheck;
    }
```

```
    public void setSpoofCheck(int spoofCheck) {
        this.spoofCheck = spoofCheck;
    }

    public void setSupportedRate(String supportedRate) {
        this.supportedRate = supportedRate;
    }

    public void setExtendedRate(String extendedRate) {
        this.extendedRate = extendedRate;
    }

    public void setHtCapability(String htCapability) {
        this.htCapability = htCapability;
    }

    public void setMac(String mac) {
        this.mac = mac;
    }
```

Above figure shows the getter and setter of the MacAddress object.

```
@Override
public int hashCode() {
    int hash = 5;
    hash = 17 * hash + Objects.hashCode(this.mac);
    hash = 17 * hash + Objects.hashCode(this.supportedRate);
    hash = 17 * hash + Objects.hashCode(this.extendedRate);
    hash = 17 * hash + Objects.hashCode(this.vendor);
    hash = 17 * hash + this.spoofCheck;
    return hash;
}

@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final MacAddress other = (MacAddress) obj;
    if (this.spoofCheck != other.spoofCheck) {
        return false;
    }
    if (!Objects.equals(this.mac, other.mac)) {
        return false;
    }
    if (!Objects.equals(this.supportedRate, other.supportedRate)) {
        return false;
    }
    if (!Objects.equals(this.extendedRate, other.extendedRate)) {
        return false;
    }
    if (!Objects.equals(this.htCapability, other.htCapability)) {
        return false;
    }
    return true;
}
```

Figure 3.7.2 Overriding hashCode () and equals ()

The hashCode () and equals () is override in order to compare the MacAddress object and eliminate the redundancy of duplicate objects in the list.

3.8 Student Class

```
public class Student {

    private String name;
    private int studentID;
    private String course;
    private String mac;
    private String email;
    private double timestamp;
    private String attendance;
    private int timecount;
    private int count;
    private int totalclass;
    private double percentage;

    public Student() {
    }

    ;

    public Student(int studentID, String name, String email, String course, String mac) {
        this.name = name;
        this.email = email;
        this.studentID = studentID;
        this.course = course;
        this.mac = mac;
        this.timecount = 0;
    }

    public Student(int studentID, String name, String course, String email, int count, int totalclass, double percentage) {
        this.studentID = studentID;
        this.name = name;
        this.email = email;
        this.course = course;
        this.count = count;
        this.totalclass = totalclass;
        this.percentage = percentage;
    }
```

Figure 3.8.1 Constructor

The above figure shows the constructor of the Student object. The constructor consists of student's name, student ID, course, MAC address, email address, timestamp percentage in Timestamp Algorithm, attendance, total number of class attended, total class of the subject and the attendance in percentage.

```
public String getName() {
    return name;
}

public int getCount() {
    return count;
}

public int getTotalclass() {
    return totalclass;
}

public double getPercentage() {
    return percentage;
}

public int getStudentID() {
    return studentID;
}

public String getCourse() {
    return course;
}

public String getMac() {
    return mac;
}

public String getEmail() {
    return email;
}

public int getTimeCount() {
    return timecount;
}
```

```
public String getName() {
    return name;
}

public int getCount() {
    return count;
}

public int getTotalclass() {
    return totalclass;
}

public double getPercentage() {
    return percentage;
}

public int getStudentID() {
    return studentID;
}

public String getCourse() {
    return course;
}

public String getMac() {
    return mac;
}

public String getEmail() {
    return email;
}

public int getTimeCount() {
    return timecount;
}
```

Figure 3.8.2 Getter and Setter

Above figure shows the getter and setter for the Student object.

3.9 MySQL Database

MySQL is the database that will be used in the project. It is used to store the student's information, the MAC address information and the attendance information. 3 tables will be created in the database. The student database is created to stores the name, student ID, course, email and their MAC address of their smartphone. The next table is the attendance table which is created to store the student information and their total attendance for the class. The last table is the identifier table which is created to store the MAC address and their respective unique identifier for anti MAC spoofing purpose.

3.9.1 Student Table

studentid	name	email	course	mac
1304567	Terrence Ang	terrence@gmail.com	CT	24:df:6a:c4:85:bc
1402309	Debbie Soo	debbie@gmail.com	IA	e8:94:f6:22:2f:ef
1403534	Low Chun Fai	chunfai0707@gmail.com	CN	fc:3f:7c:0c:50:2b
1508263	Kent Ma	kent@hotmail.com	IB	38:2c:4a:4f:40:ab
1605513	Au Yeong Ming	minaming98@hotmail.com.my	IB	28:c2:dd:43:06:c9
1605578	Jasmine Chang	iasminelol@gmail.com	FN	1c:7b:21:57:20:9f
1706023	Charmine Lee	charmine99@gmail.com	IA	24:e3:14:6a:71:1b

Figure 3.9.1 Student Table

Above diagram shows the structure of the student table in the database. Student ID is the primary of the table. Student name, email, course and their MAC address will be stored in this table. The data in this table is generated dynamically by using the Student ID to MAC binding algorithm in the Main Program when the student register using the webpage provided in the system.

3.9.2 Attendance Table

studentid	name	course	email	attendance	totalclass	attendancepercent
1304567	Terrence Ang	CT	terrence@gmail.com	1	4	25.00
1402309	Debbie Soo	IA	debbie@gmail.com	3	4	75.00
1403534	Low Chun Fai	CN	chunfai0707@gmail.com	1	4	25.00
1508263	Kent Ma	IB	kent@hotmail.com	4	4	100.00
1605513	Au Yeong MIng Ming	IB	minaming98@hotmail.com.my	1	4	25.00
1605578	Jasmine Chang	FN	iasminelol@gmail.com	2	4	50.00
1706023	Charmine Lee	IA	charmine99@gmail.com	1	4	25.00

Figure 3.9.2 Attendance Table

The attendance table is used to store the attendance information of the students. The student ID, name, course, email, their attendance count, total number of class as well as their attendance in percentage is stored in this table. The attendance percentage is calculated by dividing the attendance count with the total number of class.

3.9.3 Identifier Table

mac	supportrate	extended	ht_capability
1c:7b:21:57:20:9f	1. 2. 5.5. 11	6. 9. 12. 18. 24. 36. 48. 54	0x012c
24:df:6a:c4:85:bc	1. 2. 5.5. 11	6. 9. 12. 18. 24. 36. 48. 54	0x112d
24:e3:14:6a:71:1b	1. 2. 5.5. 11	6. 9. 12. 18. 24. 36. 48. 54	0x0020
28:c2:dd:43:06:c9	1(B). 2(B). 5.5(B). 11(B). 6. 9. 12. 18	24. 36. 48. 54	0x016e
38:2c:4a:4f:40:ab	1(B). 2(B). 5.5(B). 11(B). 9. 18. 36. 54	6. 12. 24. 48	0x01ee
e8:94:f6:22:2f:ef	1(B). 2(B). 5.5(B). 11(B). 9. 18. 36. 54	6. 12. 24. 48	0x016e
fc:3f:7c:0c:50:2b	1(B). 2(B). 5.5(B). 6. 9. 11(B). 12. 18	24. 36. 48. 54	NULL

Figure 3.9.3 Attendance Table

For the identifier table, the MAC address unique identifier is stored in this table to create a MAC fingerprint to uniquely identify each device. The identifier stored in the table are supported rate, extended supported rate and HT capabilities info. This table will be generated dynamically through the insertIdentifier method in the Main Program where the unique identifier of each MAC address will be extracted and inserted to the database.

3.9.4 Entity Relationship Diagram

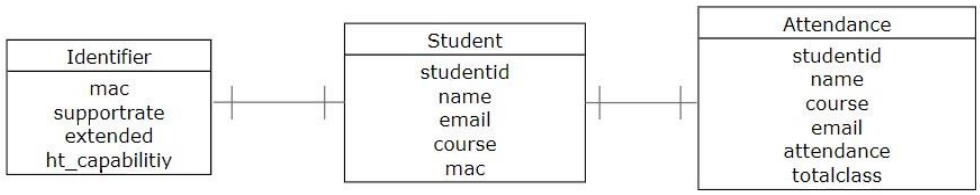


Figure 3.9.4 Entity Relationship Diagram

Based on the ERD diagram of the database. Each table has a one-to-one relation with each other. Each MAC address of the student will has only 1 record of unique identifier. Each student will has 1 record of their attendance information.

Chapter 4: System Implementation and Testing

4.1 Student Registration

During the data preprocessing period, which is the first 2 weeks of the semester, the students are able to register themselves to the system. The students can register themselves into their system with their student information as well as their smartphone's MAC address. The student ID to MAC address binding algorithm will be used to extract the student information, bind the student ID with the MAC address as well as insert all the information into the database. To implement the system, the lecturer or tutor will have to first run the Wireshark to capture all the packets. For the Student ID to MAC address binding, the Wireshark is required to run without the monitor mode at the local host of the computer. The computer will serve as a server to process all the information and update to the database.

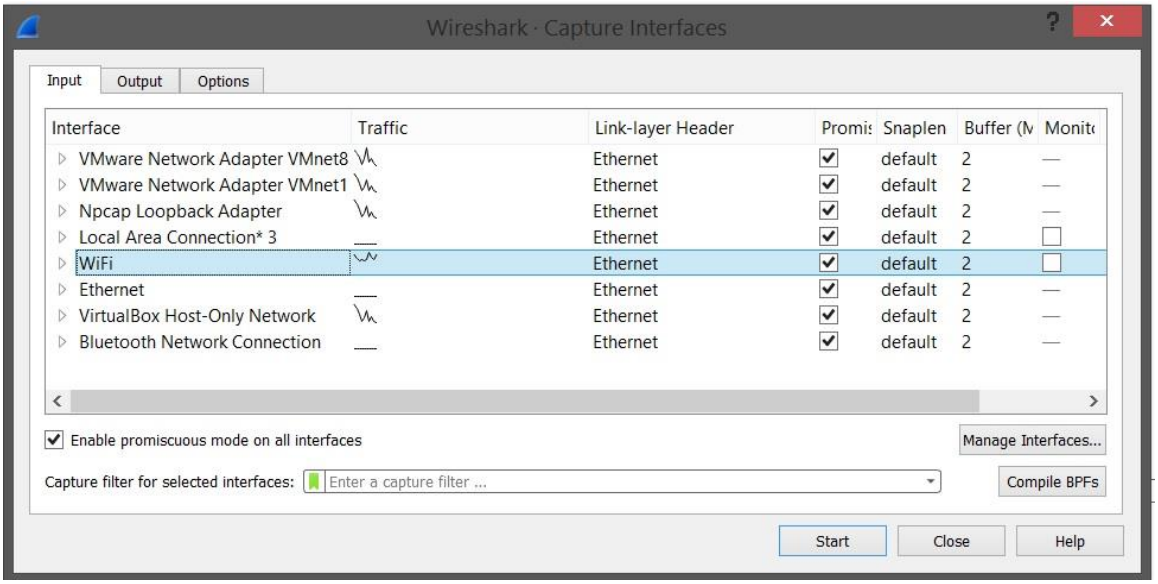


Figure 4.1.1 Turn off Monitor Mode

By capturing the local network traffic, the Wireshark is able to capture all the traffic going into the host computer. The student will connect to the local area network and access the computer to access the registration webpage. The student will access the webpage and fill in their relevant information and submit the information through the webpage that used the POST method.

Upon the submission of the information, Wireshark is able to capture all the traffic from the smartphone. After the capture, the lecturer of the tutor will need to filtered out the HTTP packets using the command “http.request.method == POST” and export the packets as a text file. The text file will be input into the system for processing.

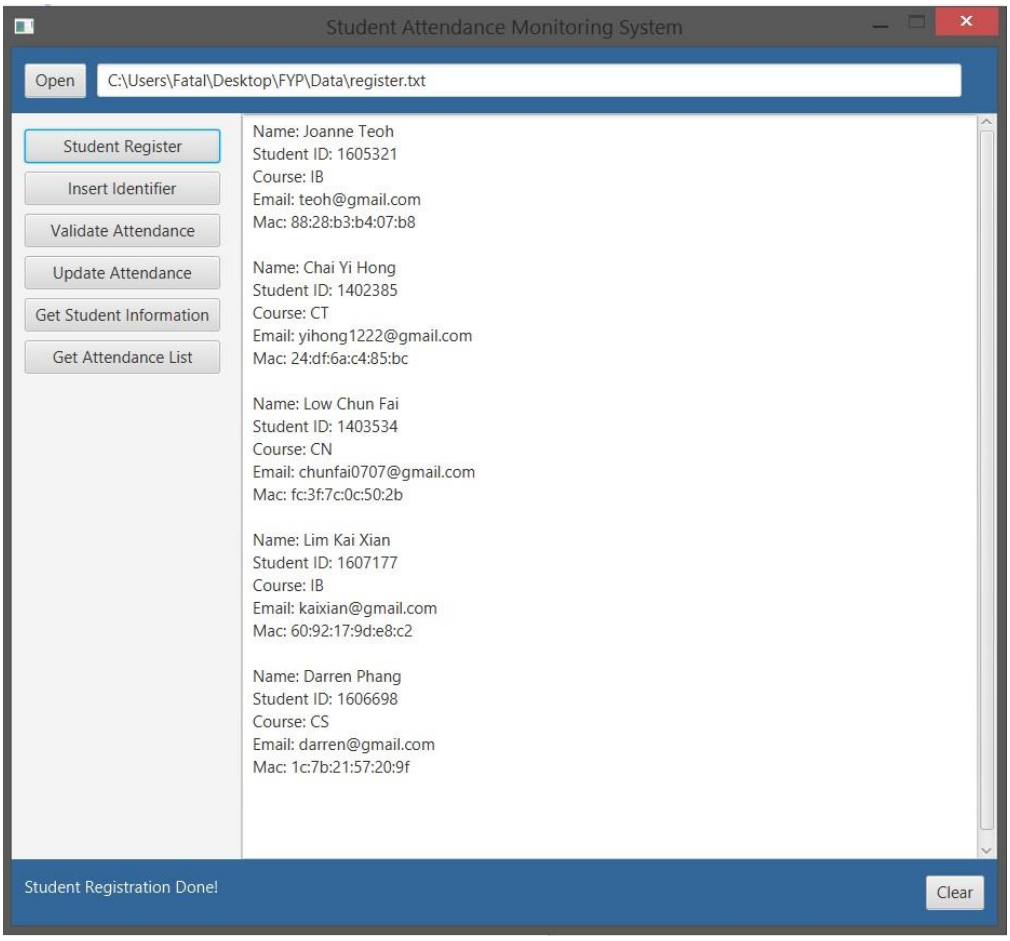


Figure 4.1.2 Output of Student Registration

Next, the text file can be chosen by clicking the open button in the system. The student registration process will be execute by clicking the Student Register button in the system. As shown in the above figure, the sample capture with 5 students is successfully registered into the database. Upon successful registration, the information of the students will also be displayed in the output screen of the system.

studentid	name	email	course	mac
1402385	Chai Yi Hong	vihong1222@gmail.com	CT	24:df:6a:c4:85:bc
1403534	Low Chun Fai	chunfai0707@gmail.com	CN	fc:3f:7c:0c:50:2b
1605321	Joanne Teoh	teoh@gmail.com	IB	88:28:b3:b4:07:b8
1606698	Darren Phang	darren@gmail.com	CS	1c:7b:21:57:20:9f
1607177	Lim Kai Xian	kaixian@gmail.com	IB	60:92:17:9d:e8:c2
NULL	NULL	NULL	NULL	NULL

Figure 4.1.3 Student Table in Database

Above figure shows the student information in the student table in the database. The student information is successfully inserted into the database after clicking the student register button the system.

studentid	name	course	email	attendance	totalclass	attendancepercent
1402385	Chai Yi Hong	CT	vihong1222@gmail.com	1	1	100.00
1403534	Low Chun Fai	CN	chunfai0707@gmail.com	1	1	100.00
1605321	Joanne Teoh	IB	teoh@gmail.com	1	1	100.00
1606698	Darren Phang	CS	darren@gmail.com	1	1	100.00
1607177	Lim Kai Xian	IB	kaixian@gmail.com	1	1	100.00

Figure 4.1.4 Attendance Table in Database

Other than the student table, the information of the student also inserted into the attendance table in the database. Initially, each student who completed the registration process will be given an attendance for attending the class to register themselves to the system.

4.2 Unique Identifier Registration

During the data preprocessing period, the unique identifier of the MAC address is also required to be registered into the database to act as a fingerprint for the MAC address of the students' smartphone. To capture the unique identifier of the MAC address, a separate Wireshark is required to be run simultaneously in monitor mode with the other Wireshark that is running without monitor mode for student registration. This is because the unique identifier is available in the 802.11 frames which can be captured by enabling the monitor mode.

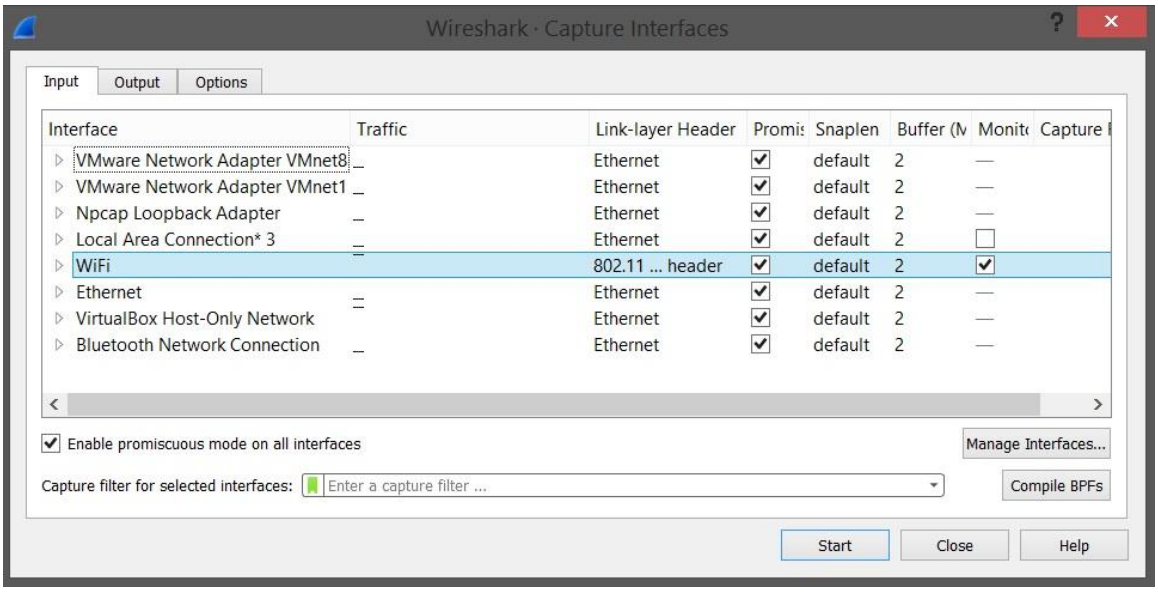


Figure 4.2.1 Turn on Monitor Mode

The 802.11 frames that will be used for unique identifier extraction is the Probe Request frames that contains the supported rate, extended supported rate and the HT capabilities info. Same as previous process, the text file is needed to be export from the Wireshark with all field expanded and input into the system.

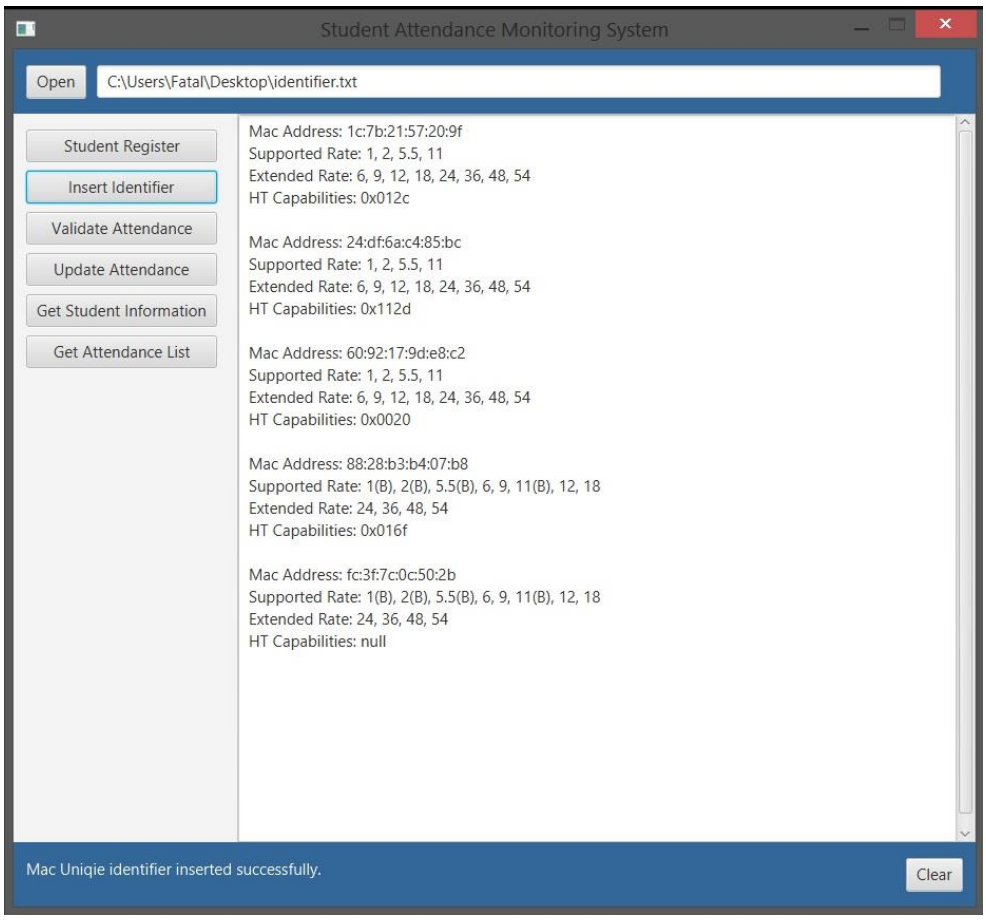


Figure 4.2.2 Output of Unique Identifier Registration

By referring to the above figure, the text file is input through the open button and the unique identifier of the MAC address is inserted successfully into the database by clicking the Insert Identifier button. The unique identifier information of the MAC address is shown in the output area of the system.

4.3 Attendance Taking and Validation

After the registration period, the system will proceed to the attendance taking and validation algorithm. For the attendance taking and validation process, the lecturer or tutor are required to run the Wireshark in monitor mode at the beginning of the class. The Wireshark will run for the whole duration of the class. The Wireshark will capture all the wireless packets in the range of the classroom. The MAC address of the students' smartphone will be captured by the Wireshark if they are in the range of the classroom. If the MAC address of the students can be found in the captured packets, it will served as a proof that the student attended the class. The attendance taker is built with the combination of the few algorithms in the Main Program such as MAC extraction algorithm, Cross-Checking captured MAC with registered MAC algorithm, Timestamp Validation algorithm as well as the Anti MAC Spoofing algorithm. Similar with the previous process, the lecturer of tutor are required to extract the Wireshark packets with all field expanded into a text file and input to the system. By clicking the validate attendance button, the system will go through all the checking and validation algorithm of the system. The system will extract all the MAC address from the input text file and compared the captured MAC addresses with the MAC addresses in the database. The Cross-Checking algorithm will also act as a filter to filter out all the irrelevant packets captured by the Wireshark. If the registered MAC address is not found in the captured MAC address, the student will be marked as absent by the system. Otherwise, the system will forward the list of MAC address for the validation algorithm which is the Timestamp validation and the Anti MAC Spoofing algorithm. If the MAC address pass both the validation algorithm, the student will be given the attendance.

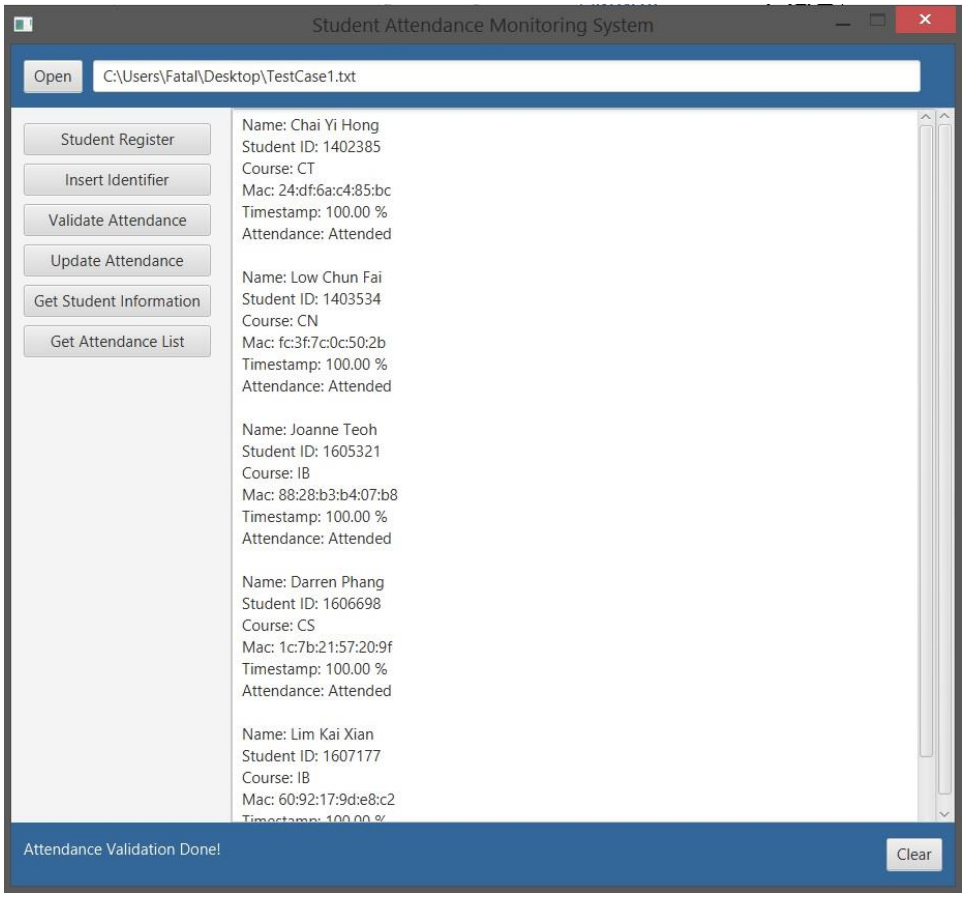


Figure 4.3.1 Output of Timestamp Validation

Above figure shows the output screen after clicking the validate attendance. The process of validation will only take a few seconds to complete. The output screen will display all the information of the student with the result of the validation. The above scenario shows that all the students are physically in the class for whole process of the class and no spoofing activity is detected for the students' MAC address.

4.4 Updating Attendance into Database

The attendance of the students is not yet update into the database in the validation process. The lecturer or tutor will need to click the Update Attendance in the system to update the attendance into the database.

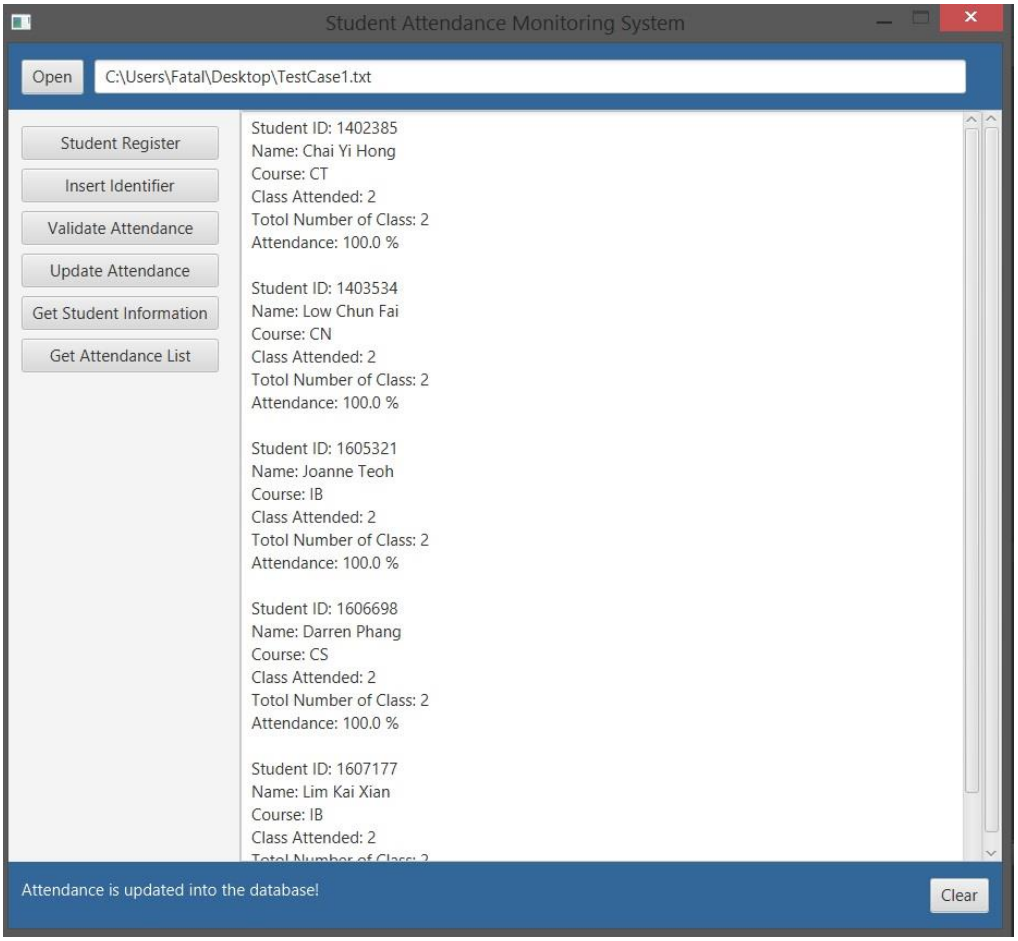


Figure 4.4.1 Output of Update Attendance

Upon clicking the Update Attendance, the system will first retrieve the attendance information from the database and update the attendance information according to the validation of the attendance. After updating the attendance, the updated attendance will be shown in the output screen.

studentid	name	course	email	attendance	totalclass	attendancepercent
1402385	Chai Yi Hong	CT	vihong1222@gmail.com	2	2	100.00
1403534	Low Chun Fai	CN	chunfai0707@gmail.com	2	2	100.00
1605321	Joanne Teoh	IB	teoh@gmail.com	2	2	100.00
1606698	Darren Phang	CS	darren@gmail.com	2	2	100.00
1607177	Lim Kai Xian	IB	kaixian@gmail.com	2	2	100.00

Figure 4.4.2 Attendance Table after Update Attendance

As shown in the above figure, the attendance is successfully updated into the database upon clicking the Update Attendance button in the system. As the scenario is that all the students attended the class, the total number of class attended of each student will be increase by one. The total class attended is divided by the total number of class conducted and the percentage is calculated.

4.5 Retrieve Student and Attendance Information

In the system, the lecturer of tutor is able to retrieve the student information or their attendance information by clicking the Get Student Information button or the Get Attendance List button. The system will retrieve the information from the database and display them into the output screen.

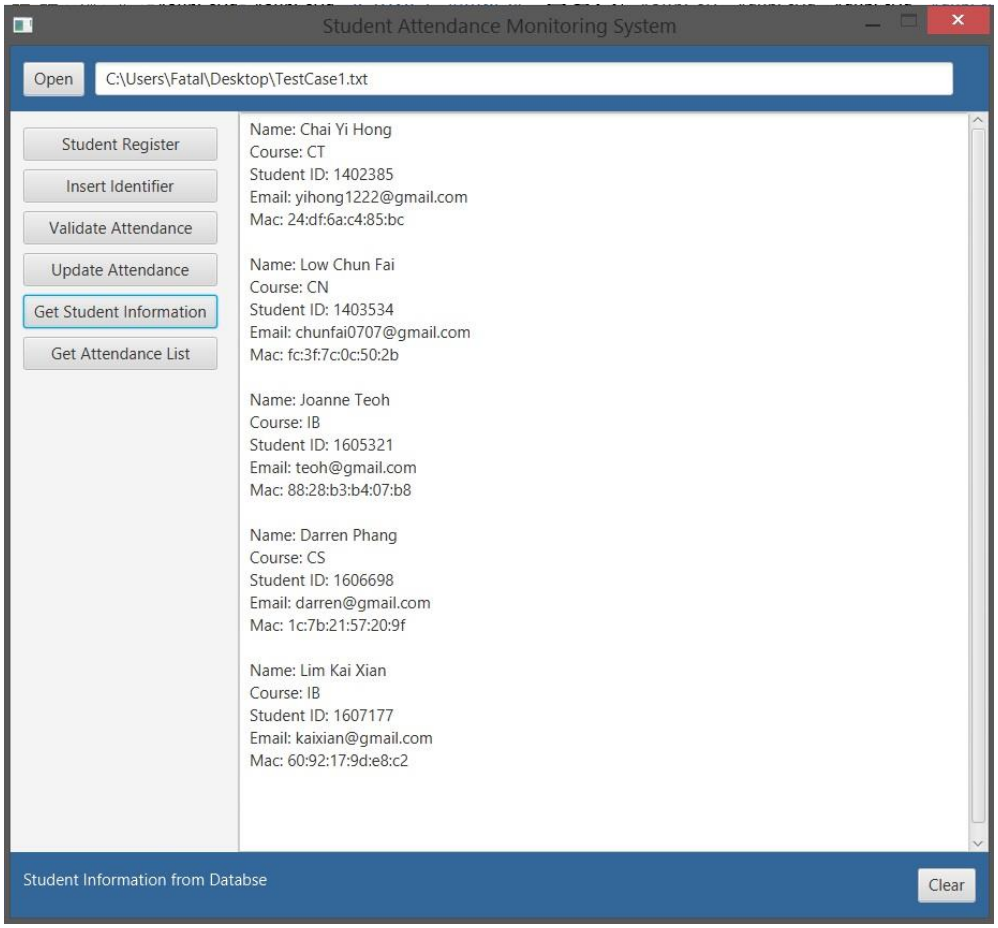


Figure 4.5.1 Student Information from Database

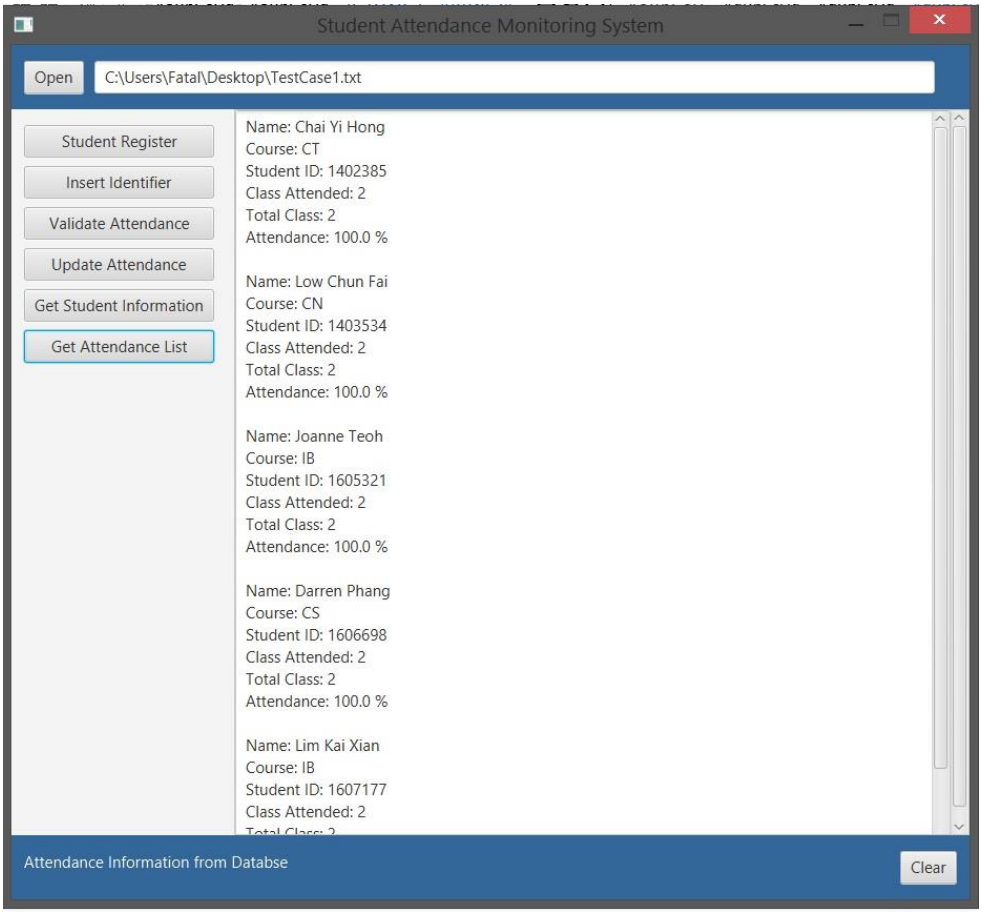


Figure 4.5.2 Attendance Information from Database

4.6 System Testing

After the completion of the system, some testing is needed to be done in order to ensure the system is working according to how it is developed. For our testing, we are using the White Box Testing technique to test our system. White Box Testing is also known as Clear Box Testing or Open Box Testing, is a testing method where the tester understand the design and structure of the system.(Software Testing Fundamental, 2018). The tester will select the different inputs for the system and determine and evaluate the outputs. Since the main purpose of this test is to determine the accuracy of our algorithms, this testing technique is suitable to test our attendance system. Some different scenarios will be used as the input to test and simulate the system.

Test/Simulation Parameter

In our test, we will use 5 mobile phones to act as 5 students in the class. The student information, the attendance information and the unique identifier information is updated into the database before the test. The information of the parameter used in this test is shown below:

Student 1: **Darren Phang**, Smart Phone Model: **Sony Xperia Z1** MAC: **1c:7b:21:57:20:9f**

Student 2: **Joanne Teoh**, Smart Phone Model: **Huawei Honor 5X** MAC: **88:28:b3:b4:07:b8**

Student 3: **Low Chun Fai**, Smart Phone Model**: Huawei P9 Lite** MAC: **fc:3f:7c:0c:50:2b**

Student 4: **Chai Yi Hong**, Smart Phone Model: **Huawei Honor 7** MAC: **24:df:6a:c4:85:bc**

Student 5: **Lim Kai Xian**, Smart Phone Model: Apple **IPhone 5s** MAC: **60:92:17:9d:e8:c2**

Test Duration: 30 minutes

Duration of each segment in Timestamp Validation algorithm: 5 minutes

Each test will last for 30 minutes and each segment will be 5 minutes. There will be total of 6 segments in the Timestamp Validation algorithm. The system will check the availability of students' MAC address in each of the segment to validate the attendance.

Scenario 1 – All Student Attended

In the first scenario, all the students attended the class and they are all physically in the classroom throughout the whole class. No MAC spoofing activity in the network.

Test ID	Description	Expected Result	Actual Result
1	All the students in attended the class and physically in the class throughout the class.	All student get 100% in timestamp validation	All student get 100% in timestamp validation

Table 4.6.1 Test Scenario 1

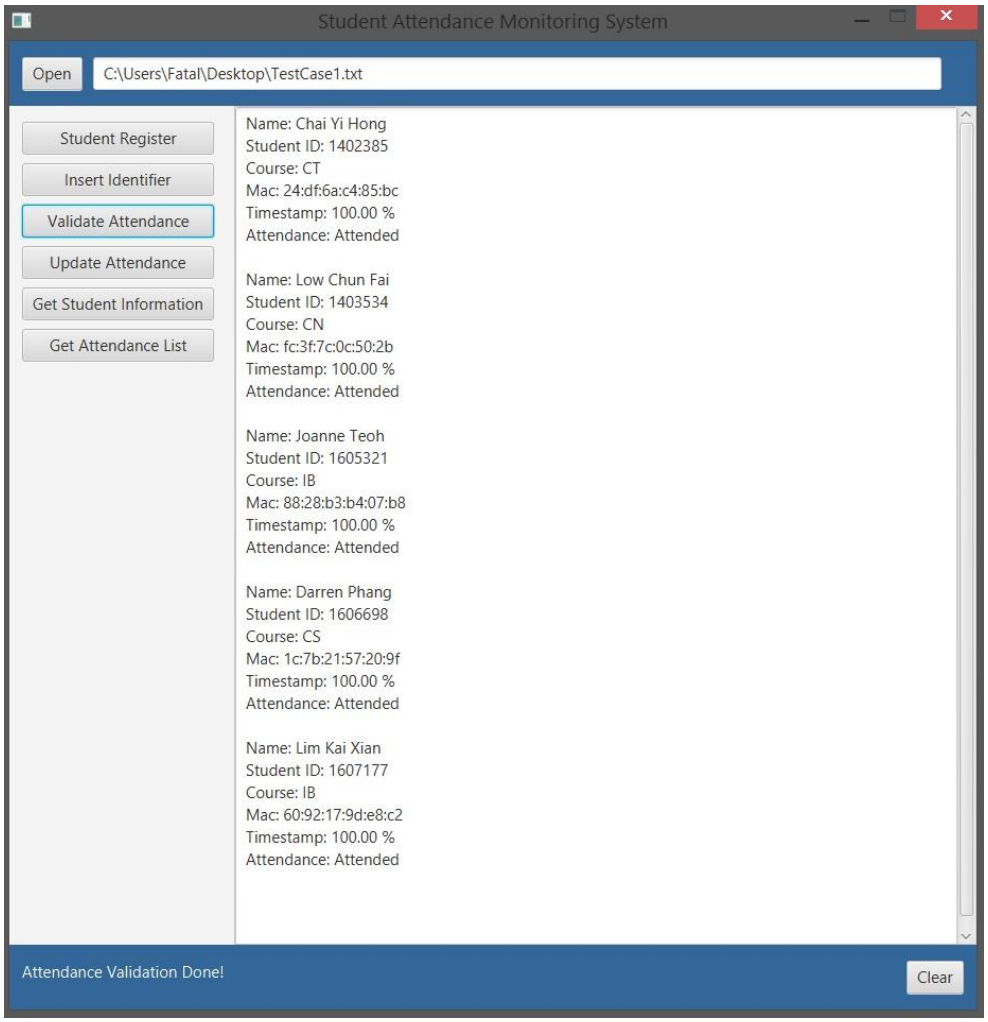


Figure 4.6.1 Result of Scenario 1

Scenario 2 – Student 1 Absent

In the second scenario, student 1, Darren Phang is absent to the class and all the other students attended the class and physically inside the classroom throughout the class. No MAC spoofing activity in the network.

Test ID	Description	Expected Result	Actual Result
2	Student 1, Darren Phang absent from the class.	All student get 100% in timestamp validation except Student 1	All student get 100% in timestamp validation except Student 1

Table 4.6.2 Test Scenario 2

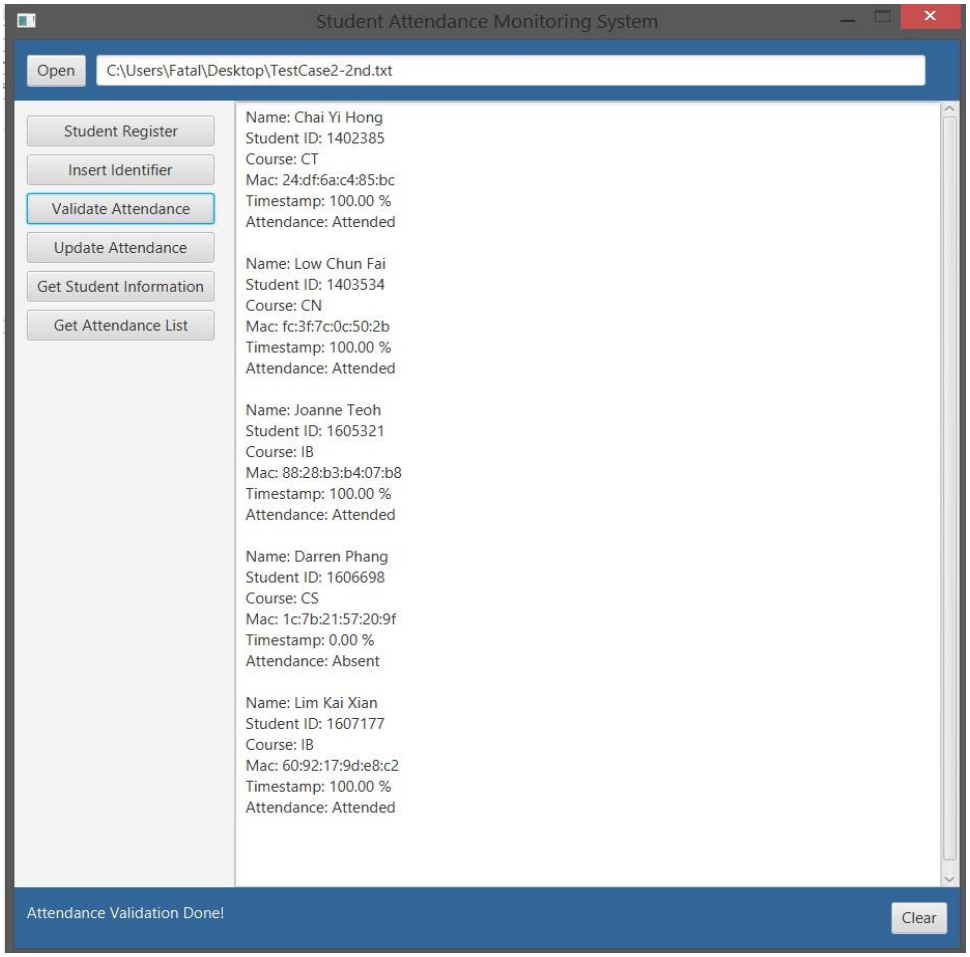


Figure 4.6.2 Result of Scenario 2

Scenario 3 – Student 2 Leaves the Class after 15 minutes

In the third scenario, student 2, Joanne Teoh is absent to the class and all the other students attended the class and physically inside the classroom throughout the class. No MAC spoofing activity in the network.

Test ID	Description	Expected Result	Actual Result
3	Student 2, Joanne Teoh leaves the class after 15 minutes of the class	All student get 100% in timestamp validation. Student 2 get 50%	All student get 100% in timestamp validation. Student 2 get 50%

Table 4.6.3 Test Scenario 3

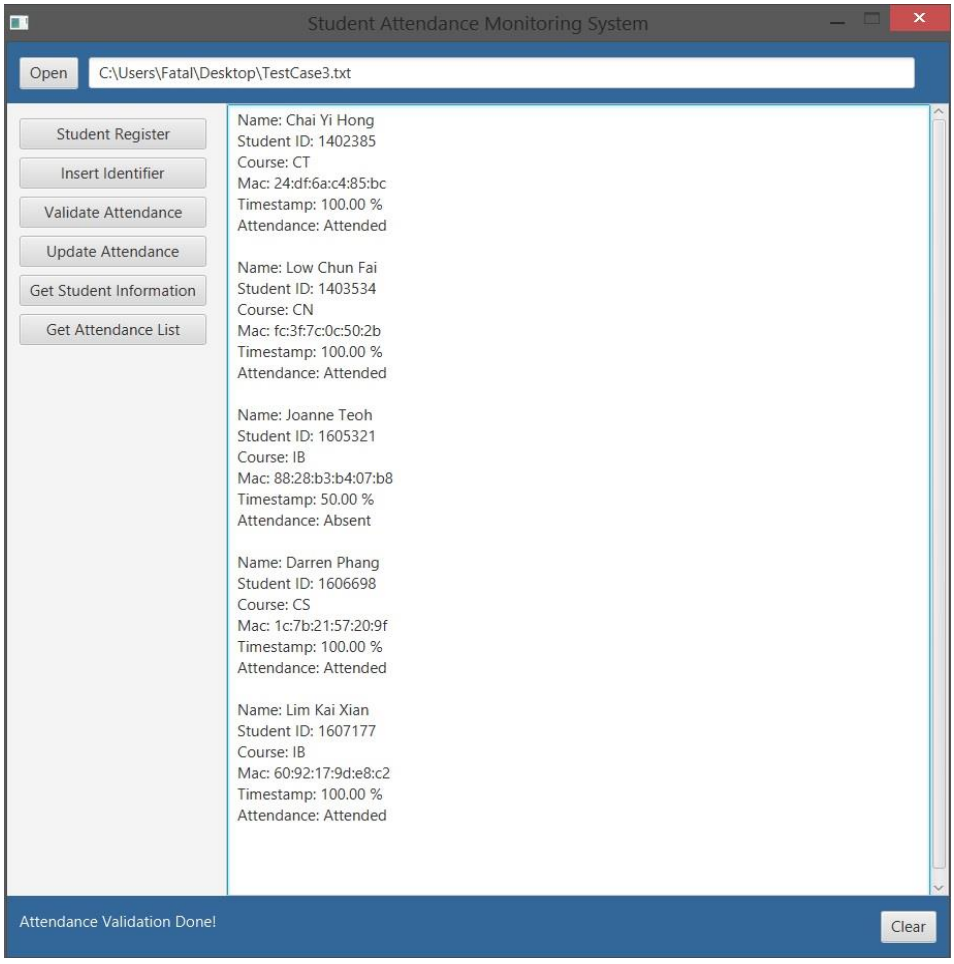


Figure 4.6.3 Result of Scenario 3

Scenario 4 – All Student Absent

In the fourth scenario, all the students are absent from the class. No MAC spoofing activity in the network.

Test ID	Description	Expected Result	Actual Result
4	All Student absent from the class	All student get 0 % in timestamp validation	All student get 0% in timestamp validation

Table 4.6.4 Test Scenario 4

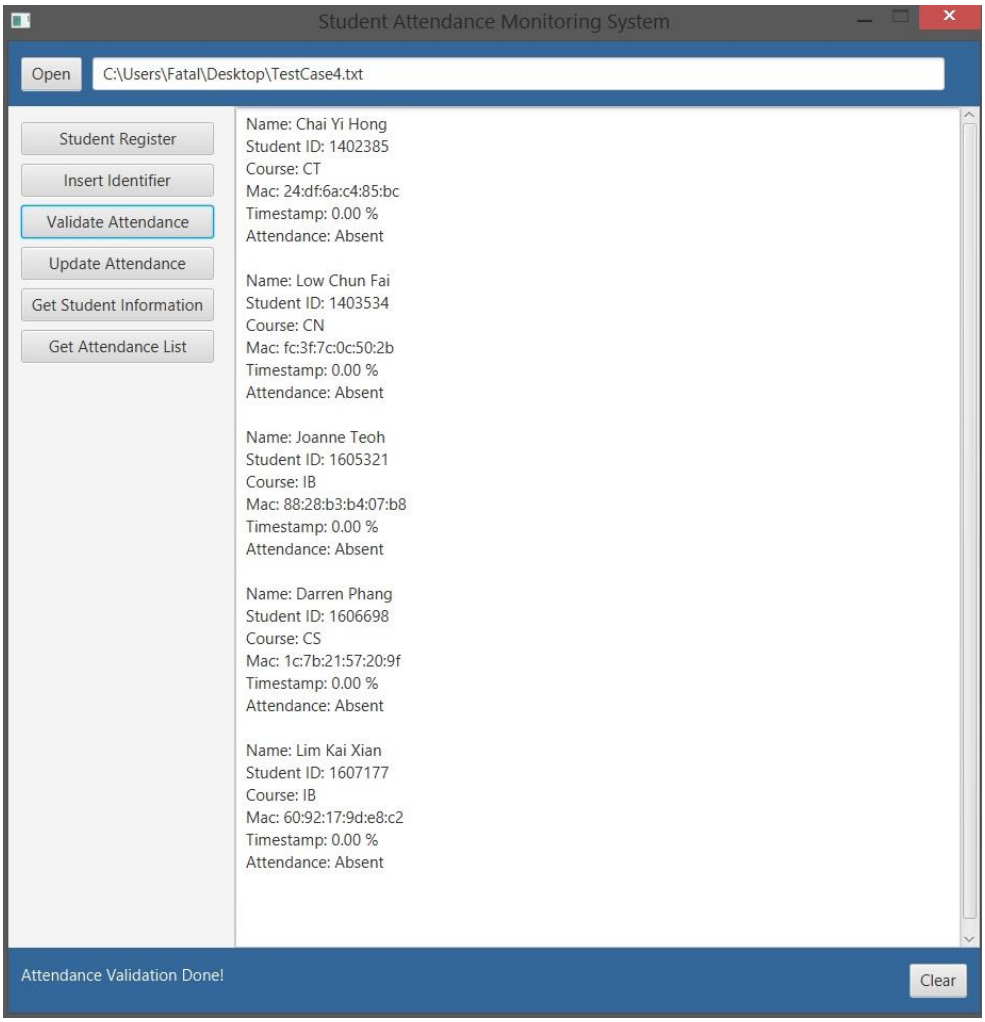


Figure 4.6.4 Result of Scenario 4

Scenario 5 – Student 2 Leaves class after 15 minutes, Student 3 leaves class after 25 minutes

In the fifth scenario, Student 2, Joanne Teoh, leaves the class after 15 minutes of the class. Student 3, Low Chun Fai, leaves the class after 25 minutes of the class. No MAC spoofing activity in the network.

Test ID	Description	Expected Result	Actual Result
5	Student 2 leaves the class after 15 minutes. Student 3 leaves the class after 25 minutes.	All student get 100 % in timestamp validation. Student 2 get 50%. Student 3 get 83.33 %	All student get 100% in timestamp validation Student 2 get 50%. Student 3 get 83.33%. Student 1 get 66.67%

Table 4.6.5 Test Scenario 5

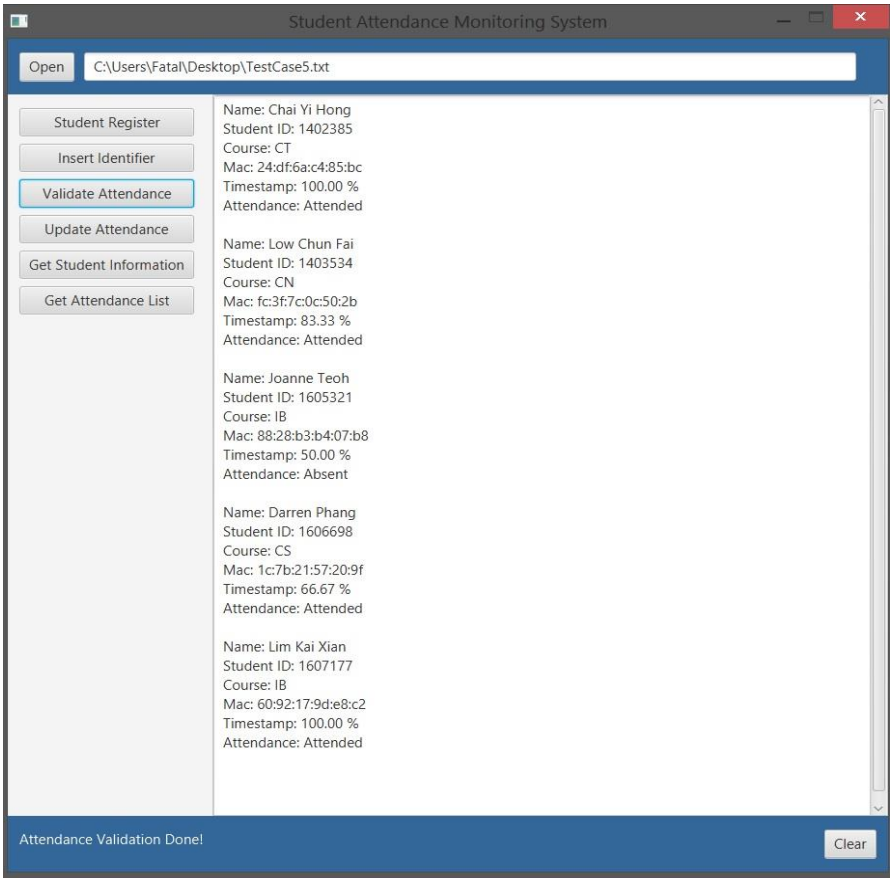


Figure 4.6.5 Result of Scenario 5

Scenario 6 – Student 4 leaves at 5 minutes, comes back at 10 minutes
Student 2 leaves at 12 minutes, comes back at 25 minutes

In the sixth scenario, Student 2, Joanne Teoh, leaves the class after 15 minutes of the class. Student 3, Low Chun Fai, leaves the class after 25 minutes of the class. No MAC spoofing activity in the network.

Test ID	Description	Expected Result	Actual Result
6	Student 4 leaves at 5 minutes, comes back at 10 minutes. Student 2 leaves at 12 minutes, comes back at 25 minutes.	All student get 100 % in timestamp validation. Student 4 get 83.33%. Student 2 get 67.67 %	All student get 100 % in timestamp validation. Student 4 get 83.33%. Student 2 get 67.67 %

Table 4.6.6 Test Scenario 6

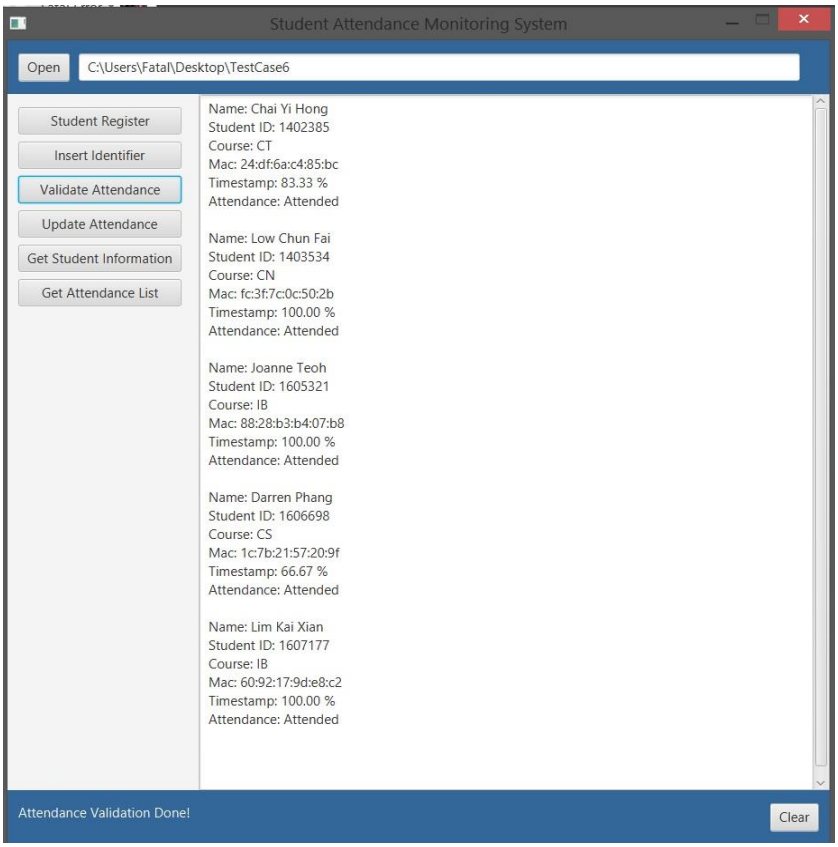


Figure 4.6.6 Result of Scenario 6

Scenario 7 – MAC Spoofing Attack using MAC address of Student 1

In the seventh scenario, Student 1, Darren Phang, is absent from the class and some student spoof his MAC address to get attendance. The MAC spoofing is done using a laptop with a USB Wi-Fi Adapter.

Test ID	Description	Expected Result	Actual Result
7	Student 1 absent from class. Other student spoof his MAC address in the class.	All student get 100 % in timestamp validation. Student 1's attendance void due to potential spoofing attack	All student get 100 % in timestamp validation. Student 1's attendance void due to potential spoofing attack

Table 4.6.7 Test Scenario 7

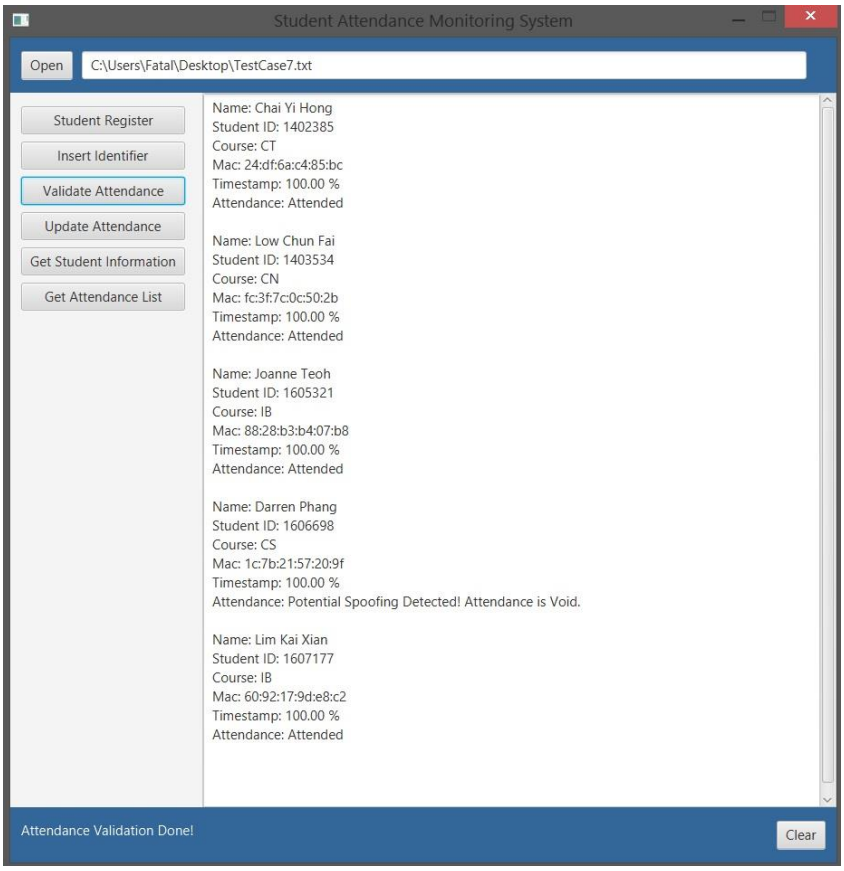


Figure 4.6.7 Result of Scenario 7

Summary of Test Result

Test ID	Description	Expected Result (Timestamp Validation)	Actual Result (Timestamp Validation)	Validity of Result
1	All the students in attended the class.	All student get 100 %	All student get 100 %	**Valid**
2	Student 1 Absent from class	Student 1 get 0%	Student 1 get 0%	**Valid**
3	Student 2 leaves the class after 15 minutes of the class	Student 2 get 50%	Student 2 get 50%	**Valid**
4	All Student absent from the class	All student get 0 %	All student get 0 %	**Valid**
5	Student 2 leaves the class after 15 minutes. Student 3 leaves the class after 25 minutes	Student 2 get 50%. Student 3 get 83.33 %	Student 2 get 50%. Student 3 get 83.33%. Student 1 get 66.67%	**Not Valid**
6	Student 4 leaves at 5 minutes, comes back at 10 minutes. Student 2 leaves at 12 minutes, comes back at 25 minutes.	Student 4 get 83.33%. Student 2 get 67.67 %	Student 4 get 83.33%. Student 2 get 66.67 %	**Valid**
7	Student 1 absent from class. Other student spoof his MAC address in the class	Student 1's attendance void due to potential spoofing attack	Student 1's attendance void due to potential spoofing attack	**Valid**

Table 4.6.8 Summary of Result

Result Discussion

Based on the result obtained from the test, we can see that the accuracy of the system is quite high. The system pass six out of seven test conducted. The result in Test 5 shows that there are irregularity where Student 1 get a 66.67% in the Timestamp Validation algorithm even though he attended fully to the class and hence it is not a valid result for the attendance. The reason that cause the invalid result is that there are possibilities that the smartphone of that particular student is not sending any Probe Request for a long period of time due to inactivity or other possible scenario that cause the unavailability of the student's MAC address in that segment of time. By looking at this case, we can see that the probing behaviour of the smartphone is one of the important element to investigate in order to improve the accuracy of the system. Looking at the overall result, it shows that the system is able to check the attendance of the student accurately.

As we can see from the result, our attendance system is able to accurate determine the validity of the attendance with our algorithms. By comparing to other existing attendance system, our system is able to reduce the amount of proxy attendance with our validation algorithms. The student is not able to sign their attendance and leave the class early with our attendance system. Hence the attendance of the students is accurately obtained with our system. Next, our attendance system is unsupervised and automated where the system is able to complete the attendance taking process with a minimal human intervention. The student will only require to go in the classroom with their mobile Wi-Fi turned on and the lecturer will only require to start the Wireshark and input the file into the system at the end of the class. Our system also can complete the attendance taking process in a short amount of time as the attendance taking process can be done only with a few clicks of button. Thus, our system outperform other system such as RFID biometric system where the student will still need to line up and check their attendance using their biometrics traits or RFID cards, which is still not time effective.

Chapter 5: Conclusion

5.1 Project Review and Conclusion

In conclusion, the final deliverable in this project is an automated attendance taker using passive MAC probing that can improve the attendance taking process. The automated attendance system required minimal human intervention and the whole process of attendance taking will only require a minimal amount of time. With the implementation of the automated attendance system, the attendance taking process will be time efficient and the education institutes are able to keep track the student's attendance accurately and effectively.

The motivation of this project is to solve the problem with the current attendance system, where the attendance taking process requires many human interaction, time consuming and not tracking the attendance accurately. The automated attendance system can contribute largely to the education institution as the attendance tracking process is very essential for every education institution. This is due to the performance of the student in their academic studies is closely related to their attendance in the class. The system consists of a few algorithms which are Student ID to MAC address binding algorithm, MAC address extraction algorithm, Timestamp Validation algorithm, Anti MAC Spoofing algorithm and Cross-Checking MAC address algorithm.

The main objectives of the project, which is developing an automated attendance system is achieve by developing and integrating all the algorithms together in this system. With the completion of our objectives, the automated system is able to solve the problem with our current attendance system as stated above. Our system is able to outperform other attendance system in terms of accuracy, cost, less proxy attendance and time efficiency because the whole attendance process will only takes a few button of clicks and the whole process of attendance taking is unsupervised and passive.

5.2 Limitation

There are some limitation with the current system. The limitation of the system are stated below -:

- The difference smartphone have different vendor specific probing behaviours.

  As our system heavily depends on the probing of the smartphone in the attendance taking process, the probing behaviours of the smartphone become an essential part that could affect the accuracy and reliability of the system.

- Hardware limitation

  Our system uses Wireshark to capture all the packets to input into our attendance system. However, monitor mode is needed to be turned on in order to capture all the raw 802.11 frames. Some of the wireless adapter does not support the monitor mode in the Wireshark.

- System Interface

  The GUI of the system is a simple interface that is created to demonstrate the algorithms inside the main program. It only has a very limited functionalities and limited input validation.

- Securities issues

  There is an anti MAC spoofing algorithm in our system. However, the accuracy and reliability of the algorithm is relatively low due to the limited amount of identifier used to generate the fingerprint. Some device might have the same fingerprint using the identifiers in our system.

- Specific setting required in exporting captured packets in Wireshark

  In order for the system to function properly, the text file exported from Wireshark need to follow a specific setting, which is expanding all the field in the captured packets. Otherwise, the system is not able to function properly.

5.3 Future Work

As discussed in the previous section, there are still many limitation in the system that can be improved to enhance the overall accuracy and usability of the system. First, a fully functional graphical user interface can be developed to fully utilize the algorithms behind the system. A better interface can further enhance the usability of the system and largely increases the commercialization value of the system. The security issues of the system can also be improved by implementing a more secure method to prevent any possible attacks on the network that could affect the accuracy of the system. Some method such as behavioural analysis on the probing behaviours of the mobile devices that determine the potential attacks based on the behaviour and patterns of the mobile devices can be used to improve the security of the system.

Next, the system can also add some functionalities such as integrating the Wireshark in the system to unify both program together into one program so that the users do not required to use two separate program to check the attendance. Additionally, the system can also connect to the central database of the education institution to directly retrieve or update the information from and to the central database. Last by not least, a web application can also be created for the students or lecturer to access and check their attendance information. The system can also can be integrate into the current student portal for the student to obtain their attendance information.

References

1. C. Idland, 2011. *Detecting MAC Spoofing Attacks in 802.11 Networks through Fingerprinting on the MAC Laye*r. Masters thesis, Department of Telematics, NTNU, June 2011. 96 pages.

2. Gu, W. et al., 2008. *On Security Vulnerabilities of Null Data Frames in IEEE 802.11 based WLANs, The 28th International Conference on Distributed Computing Systems* pp. 28-35.

3. H. K. Nguyen & M. T. Chew, 2017. *RFID-based attendance management system.* 2nd Workshop on Recent Trends in Telecommunications Research (RTTR), Palmerston North, pp. 1-6.

4. Khatun, A. K. M. F. Haque, S. Ahmed and M. M. Rahman, 2015. *Design and implementation of iris recognition based attendance management system.* International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka, pp. 1-6.

5. S. Pss & M. Bhaskar, 2016. *RFID and pose invariant face verification based automated classroom attendance system*. International Conference on Microelectronics, Computing and Communications (MicroCom), Durgapur, pp. 1-6.

6. Saraswat, C. & Kumar, A., 2010. *An efficient automatic attendance system using fingerprint verification technique*. International Journal on Computer Science and Engineering, 2(02), pp.264-269.

7. Shoewu, O. & Idowu, O.A., 2012. *Development of attendance management system using biometrics.* The Pacific Journal of Science and Technology, 13(1), pp.300-307.

8. Software Testing Fundamental. 2018. *White Box Testing*. Available at: http://softwaretestingfundamentals.com/white-box-testing/. [Accessed 18 August 2018].

9. T. S. Lim, S. C. Sim & M. M. Mansor, 2009. *RFID based attendance system*. IEEE Symposium on Industrial Electronics & Applications, Kuala Lumpur, pp. 778-782.

10. T. Sharma & S. L. Aarthy, 2016. *An automatic attendance monitoring system using RFID and IOT using Cloud.* Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, pp. 1-4.

11. Talaviya, G., Ramteke, R. & Shete, A.K., 2013. *Wireless fingerprint based college attendance system using ZigBee technology*. International Journal of Engineering and Advanced Technology (IJEAT) ISSN, 2249, p.8958.

12. V. Shehu & A. Dika, 2010. *Using real time computer vision algorithms in automatic attendance management systems*. Proceedings of the ITI 2010, 32nd International Conference on Information Technology Interfaces, Cavtat/Dubrovnik, pp. 397-402

.

Turnitin Originality Report

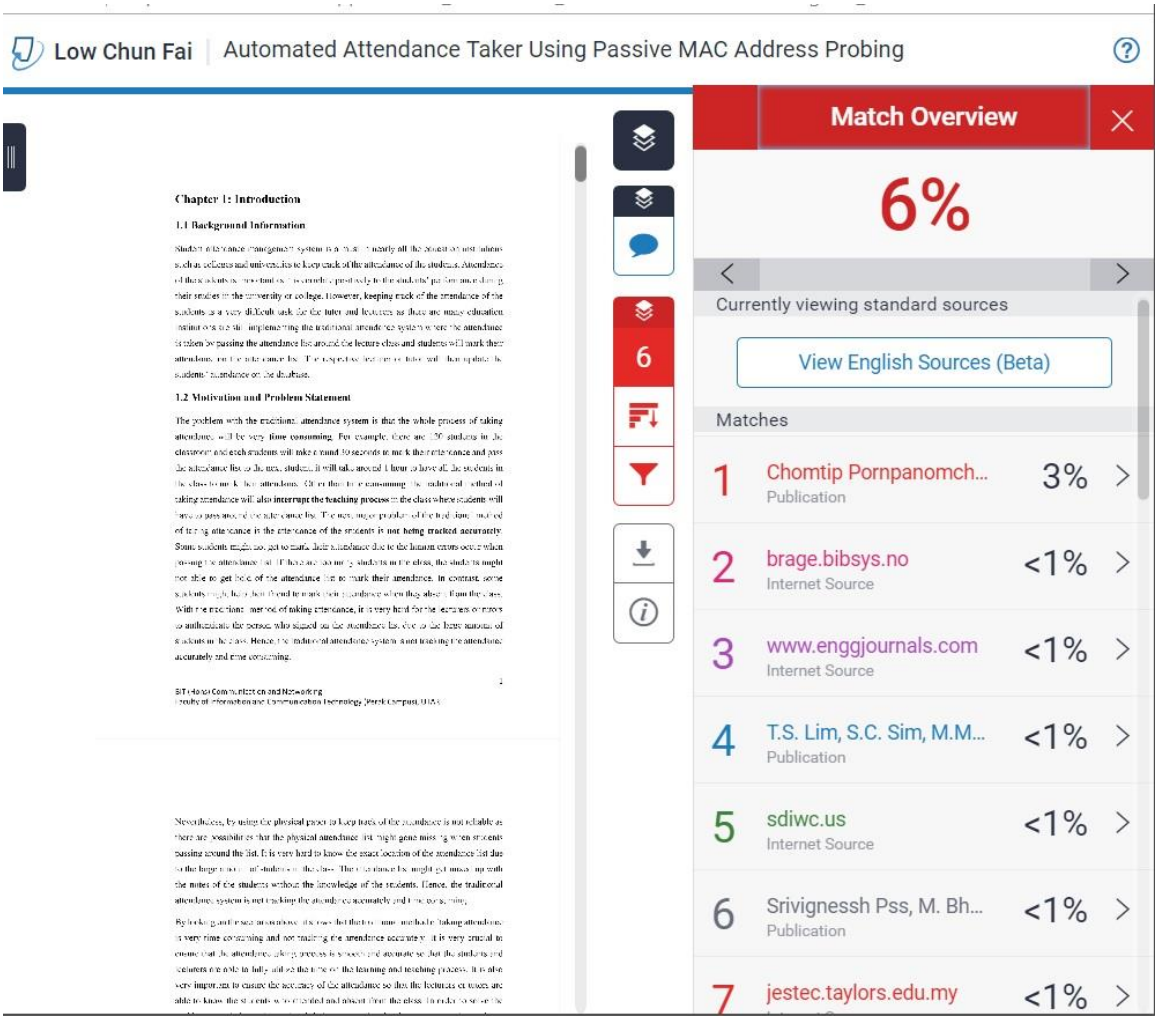


Figure A-1 Turnitin Originality Report