CS 6400 Database Systems Concepts and Design - Fall, 2015
Final Group Project Interim Report

Wireshark-Captured Networking Data Management

Jiaxiu Zhao (Telecom of ECE) : jzhao17@gatech.edu

DB Background: Basic understanding of SQL language by self-study and recently got deeper understanding from this course. Have no experience on any programming and implementations related to database. Currently I've learnt how to use Workbench of MySQL to create, manage database and have done some implementation using GUI, JDBC, Basic Java and MySQL together.

Yiling Yin (Telecom of ECE): yyin60@gatech.edu

DB Background: Simple database related knowledge. Have no practical experience in database application implementation before. By doing this project so far, I've learnt how to use MySQL workbench to do database management. Also, I've learnt how to design a database application and realize it in Java using GUI, JDBC, SQL and other programming skills.

Zhongyi Luo (Electronic Design & Application of ECE) : zluo60@gatech.edu

DB Background: Basic knowledge in SQL and Java language by self-study and recent lectures. Have no former experience in the field of database. Currently, I learned how to import packages produced by Wireshark and dissemble them into tuples for the update of entry table as well as how to drive SQL by Java and insert these package messages into database system.

1. Goals

The goal of this project is to implement a database system which deals with real-time networking data captured by Wireshark to help with network monitoring. We are going to use wireshark to capture real-time networking data including destination and source IP address, domain names, protocols involved and so on. The data collected will be generated by various devices (Apple, Android device, laptop, pad, phones and so on) used by multiple users and will be stored into the selected database and managed by the database management system we implemented. Network administrator could check the access entry history to see if there were illegal website access. The appropriate actions would be taken to either warn the users or restrict the access of specific user. Detaied functions will be illustrated in next section.

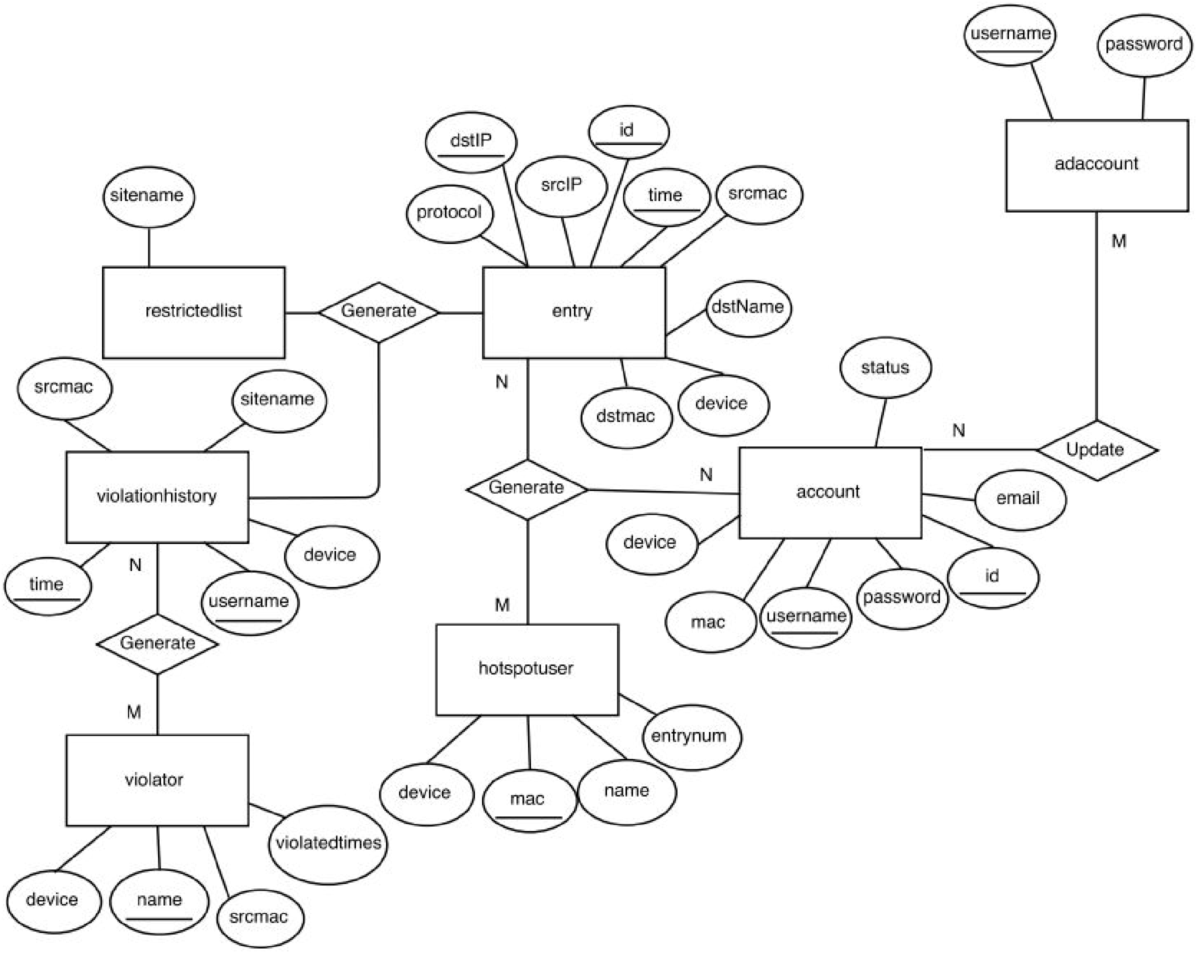
From this project we are expecting to learn how to design and implement a real-life Database Management System. To be more specific, we can learn how the database should be configured, created, managed and accessed by using MySQL Workbench and Java program. We are supposed to learn how to write JDBC to connect to database to perform updating, query, and altering to achieve different purposes. And during the implementation process, we will learn the details of table configuration, the interaction between Java program and the database. Also, it is expected that we will encounter some common problems and bugs and figure out the solutions to them.

2. Functionality of End Product

a. Users could register and apply for an account with required information. *
b. Administrators can check and manage the account information and approve potential hotspot users' registration request. Once approved, an email will be sent to that user containing the hotspot's name and password. *
c. Hotspot users' access entry information can be collected by Wireshark, then parsed and imported into the database by the administrator.
d. Administrator is able to query for specific access histories according to user name and device type.
e. Administrator is able to update and import the restricted website list into the database.
f. Administrator can check for the violation history to see if a specific user violates the rules or not.
g. Administrator is able to send a warning message to the hotspot user by email to warn his/her inappropriate networking behaviors. *
h. Administrator is able to manage the hotspot according to the violation histories collected and perform some access control to block the Internet access of those violators.
i. Administrator is able to monitor the network using Wireshark. *
j. Hotspot user is able to reset his/her email and password after log in.
k. Hotspot user is able to check his/her own access entries with advanced search.

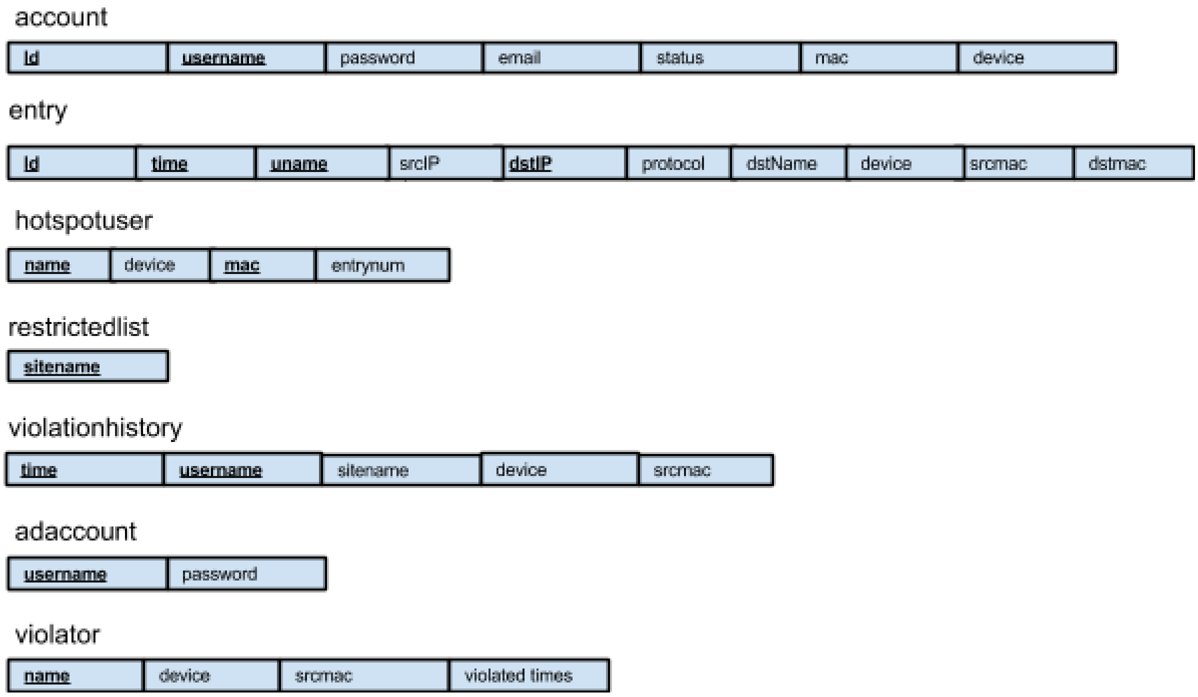
*: Completed by the time that Interim Report Submitted

3. ER Diagram



Technical details of implementation will be illustrated in Architecture section.

4. Tables



account

Id	username	password	email	status	mac	device

entry

Id	time	uname	srcIP	dstIP	protocol	dstName	device	srcmac	dstmac

hotspotuser

name	device	mac	entrynum

restrictedlist

sitename

violationhistory

time	username	sitename	device	srcmac

adaccount

username	password

violator

name	device	srcmac	violated times

5. Data Resources

a. The potential hotspot users' information, which includes username, password and email address, is collected during the user registration phase and stored in the ACCOUNT table.

b. The administrator data is pre-created and stored in the ADACCOUNT table. It includes the username and password.

c. The access entry data, which is collected by Wireshark and imported into the database by the administrator, includes the information of access time, username, srcIP, dstIP, srcMAC, dstMAC, protocols involved, dstName and device. It is stored in the ENTRY table.

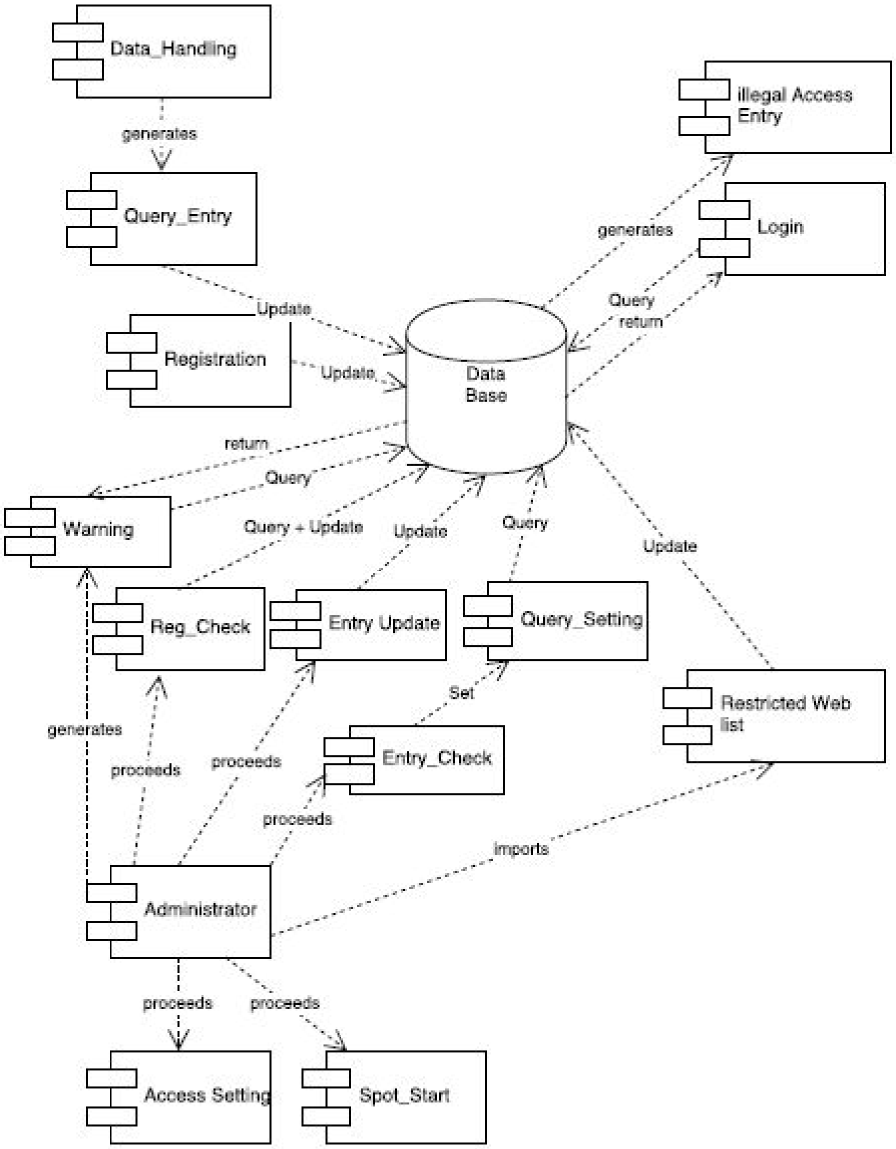
d. The data of a list of restriced website is collected from an official website and stored in the RESTRICTEDLIST table.

e. The data of violation history, which is created by joining the ENTRY and RESTRICTEDLIST table, is stored in the VIOLATIONHISTORY table.

f. The data of violators, which includes the violator's username, device, srcmac and violated times, is generated by grouping the violationhistory table by username and stored in the VIOLATOR table.

g. The data of current hotspot users' information, which includes the username, device, mac and number of entries generated by that user, is created by the join of ENTRY and ACCOUNT table and stored in the HOTSPOTUSER table.

6. Architecture:



a. Implementation details of application

1) User Registration & Login as well as Administrator Login are implemented using Java and will be shown as GUI. JDBC for connecting these GUIs will be included within the same class and algorithms such as comparison between DB username-password pair and User input are achieved under ActionListener of relative JButton. User can also change email and password after login by updating corresponding tuple in databse, which will alter his 'status' attribute and wait for the approvement of administrator.
2) The raw data is captured by Wireshark automatically in fixed time interval (or fixed export number), and then filtered by DNS protocol. Since these data are in certain format, essential information can be found and insert into a temporary table of database.
3) As user connect to connectify, an IP will be assigned to him, the last digits of IP will be the ID of user. By joining with the exsisted account by id, username are matched to cooresbinding entry records and stored in a permanent table of database.
4) Registration Check and Entry Check by administrators are implemented by querying related tables, the settings of checking input in GUI will be used as parameters in SQL queries and

sent to database by JDBC connection, update corresbonding attributes (or delete corresbonding tuples).

5) Warning message forwarding can also be achived by checking violation history/violators and message, select destination account & email address through a corresponding GUI and forwarrd the message.
6) Query_Setting has multiple parameters in its class and show multiple choices of querying parameter setting on GUI. The settings will be stored in one SQL sentence hence being sent to database to get desired entry. This is being used in entry checking of both administrator and hotspot user, for example, user's searching records based on timeslots, device and keyword.
7) Violation history will be generated by joining Restricted Web List and user Entry, and then stored in database. The join command will be sent to database after the corresponding GUI is activated as a part of Administrator functionality.
8) Administration of access control can be achieved by invoking the currently available hotspot management APP using Java. The denial of Internet access for a certain user can be done manually by the network administrator according to the violator table.

b. Components functional features

1) Login Component: In login GUI, typed information, which includes username and password, will be compared with the account record in database. If matched, the user will be guided to the user interface, the administrator will be guided to the administration interface; otherwise error message will be displayed.
2) Registration Component: In registartion GUI, typed information will be stored into the account table once registered and later can be modified by administrators as well as the user himself..
3) Warning Message Component: In this GUI, administrator can choose which user to send the warning email to. The email will be sent by clicking the confirm button after entering the required information.
4) Account Management Component: Administator can view and update the account information and approve under-checking registration queries under this section. Once approved, the account status will be set to "approved".
5) Entry Check Component: All recent entry records will be displayed and provide advanced entry search functions..
6) Data Import Component: Wireshark data packages can be parsed and imported to database in this session.
7) Violation History Component: Shows all the illegal access entries.
8) Update Restricted List Component: Provide an interface which allows administrator to import and update a new version of the restricted website list into the database.
9) Access Control Component：The connectify hotspot can be started automatically using Java functions to allow the administrator to perform some access control..
10) Network Monitor Component: The Wireshark can be started automatically using Java functions to allow the administrator to monitor the network.
11) User Entry Component: Display that user's recent access entries and provide search functions.
12) Email&Password Reset Component: Provide an interface where user can reset the password and email. Once reset, the data will be updated in the account table.

c. Tools and libraries:

Tools: MySQL 5.6, Eclipse, Wireshark, WiFi HotSpot-Connectifyme.
Library: MySQL-connector-Java.

6. Demo Plan

a. Users will firstly register the system with serveral personal information and wait for registration checking.

b. Administrator can login onto Admin GUI directly and check the registration status. The account and registration can be approved or deleted. Additional information of a user such as MAC/ID will be gathered from Hot Spot and added into user account. The ACCOUNT Info Message will be sent to User for access purpose.

c. After approval from Admin, User can login the system, view their visit entry history. Users can also change their email, password in user GUI. The relative data change in database will be showed.

d. Administrator can check the access entry of all approved users. And search parameter can be set as filter for advance searching. Users can register at most three devices so Administrator can check which device has generated entries.

e. Administrator can check the violation history of all users and show who has accessed restricted websites.

f. Administrator can send warning messages to user who violates the rule

g. Administrator can perform access control and manage the network monitor by invoking Connectify and Wireshark directly on Admin GUI.

h. Administrator can update restricted website list to database for future extension.

i. Wireshark will periodically generate captured entries and Administrator can import these monitoring files into database for future check.