



MIDDLE EAST TECHNICAL UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT

CENG491

Initial Design Report

By

EsSoft

INDEX

1. INTRODUCTION.....	3
1.1 PROBLEM DEFINITION.....	3
1.2 PROJECT SCOPE.....	3
1.3 DESIGN OF GOALS AND OBJECTIVES.....	4
1.3.1 EFFECTIVENESS.....	4
1.3.2 USABILITY.....	4
1.3.3 TIME.....	5
2. DATA DESIGN.....	5
2.1 USER.....	5
2.2 ADMIN.....	6
2.3 KEYWORDS.....	8
2.4 PROTOCOLS.....	9
2.4.1 FTP.....	9
2.4.2 SMTP.....	10
2.4.3 HTTP.....	11
3. PROJECT REQUIREMENTS.....	12
3.1 SYSTEM REQUIREMENTS.....	12
3.1.1 HARDWARE REQUIREMENTS.....	12
3.1.2 SOFTWARE REQUIREMENTS.....	13
3.2 FUNCTIONAL REQUIREMENTS.....	14
3.3 NONFUNCTIONAL REQUIREMENTS.....	14
4. SYSTEM DESIGN.....	16
4.1 USE-CASE DIAGRAMS.....	16
4.2 DFD DIAGRAMS.....	17
4.2.1 DIAGRAMS.....	17
4.2.1.1 DFD LEVEL0.....	17
4.2.1.2 DFD LEVEL1-CLASSIM.....	18
4.2.1.3 DFD LEVEL2-AUTOSENSOR.....	19
4.2.1.4 DFD LEVEL2-USER VERIFICATION.....	20
4.2.1.5 DFD LEVEL2-OUTPUT HANDLER.....	21
4.2.1.6 DFD LEVEL2-NETWORK CAPTURE.....	21
4.2.2 DFD DATA DICTIONARY.....	22
4.2.3 DFD PROCESS DESCRIPTION.....	24
4.3 CLASS DIAGRAMS.....	26
4.3.1 USER VERIFICATION.....	26
4.3.2 NETWORK CAPTURE.....	27
4.3.3 DECODER.....	28
4.3.4 PROTOCOL HANDLERS.....	28
4.4 SEQUENCE DIAGRAMS.....	30
4.5 ACTIVITY DIAGRAMS.....	31
4.5.1 LOGIN.....	31
4.5.2 CAPTURING DATA.....	32

4.5.3	DECODER.....	33
4.5.4	PROTOCOL HANDLING AND GIVING OUTPUT.....	34
5.	USER INTERFECE.....	35
5.1	GENERAL OVERVIEW.....	35
5.2	LOGIN.....	36
5.3	MENUBAR.....	37
5.3.1	FILE MENU.....	37
5.3.2	EDIT MENU.....	37
5.3.3	VIEW MENU.....	38
5.3.4	CAPTURE MENU.....	39
5.3.5	STATISTIC MENU.....	39
5.3.6	HELP MENU.....	40
5.4	TOOLBAR.....	40
5.5	PANELS.....	41
5.5.1	CAPTURED PACKETS PANEL.....	41
5.5.2	DETAILS of CAPTURED PACKAGE.....	41
6.	PROJECT SCHEDULE.....	42
7.	TESTING.....	42
8.	CONCLUSION.....	42
APPENDIX A		

1. INTRODUCTION

Network is the critical part of the computer systems. Especially, after the widely usage of internet, network becomes more important compared with the past. Every person, whether interested in computer systems or not, begins using network systems widely in daily life. Therefore, network programs, such as online chat, e-mail, are made more effective to attract users. These programs are designed by using different techniques, different algorithms, different protocols ,if we look at the concept of network systems. However, these differences cause some problems. Since there are lots of differences, identification and classification phase for computer is a big problem.

1.1 PROBLEM DEFINITION

Our Project will be used for analysing, identifying, and classifying incoming network data in terms of their protocols. For this project, there is three steps in analysing the protocols which are filtering, feature analysing and classification of protocols. Filtering means throwing the unnecessary items from the information list. After this step, by using pattern recognition, program identify the possibilities which protocols can be. At the last step, namely classification, we determine the right protocol.

1.2 PROJECT SCOPE

Firstly, since our project basically concerns about protocols, we want to give brief explanation of which protocols will be identified in our project:

Simple Mail Transfer Protocol (SMTP) is a relatively simple, text-based protocol, in which one or more recipients of a message are specified (and in most cases verified to exist) and then the message text is transferred.

- Hypertext Transfer Protocol (HTTP) is a communications protocol used to

transfer or convey information on intranets and the World Wide Web.

- File Transfer Protocol (FTP) is used to transfer data from one computer to another over the Internet, or through a network.
- Post Office Protocol version 3 (POP3), an application-layer Internet standard protocol, to retrieve e-mail from a remote server over a TCP/IP connection.
- The Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. It can be used to create two-party, multiparty, or multicast sessions that include Internet telephone calls, multimedia distribution, and multimedia conferences.

1.3 DESIGN GOALS AND OBJECTIVES

1.3.1 EFFECTIVENESS

Since our project is about identifying and classifying protocols, its main success depends on how many protocols it can identify. In other words, can our program recognize new protocols without looking at its port number?. For this reason, instead of analysing lots of protocols and putting it in our database system, we try to develop our decoder phase such a way that when new protocol comes, our program can create this protocols keywords. As a result, with this feature, at the end of our project, there will be no need to put new protocols in database by hand. This feature increase effectiveness.

1.3.2 USABILITY

Our project is not a specific area project. For instance, one company which is not related with the computer systems may want to control its employers network traffic. Therefore, if this company wants to use our program, user should understand easily the syntax of program. For this necessity, our project's interface will be as possible as simple.

1.3.3 TIME

Since network traffic usually is not linear, time becomes critical issue that have to be fast in order to capture files and classifying it. It is directly related with the effectiveness of program. Because, time is the most important issue when it is compared with the other issues in terms of effectiveness. First step, to increase the time effectiveness, we choose our programming language as c++. After this step, we make our algorithms and tools that is not negatively affect our program.

2. DATA DESIGN

Database system is very important part of our project, because whole the special keywords are hold in there. Classification and identification is made according to the data that is searched in our project's database. In our database system, there will be three tables, namely user, admin, and protocols.

2.1 USER

User table holds the information of the user that is not admin but using the project in the least security level compared with admin. Additionally, user can not start capturing the files from the network traffic.

Attributes are:

- 2 Username
- 3 Password
- 4 User_secret_question
- 5 User_secret_answer
- 6 Name
- 7 Surname
- 8 E-mail

SQL code will be like that:

```
CREATE TABLE User(  
    Username VARCHAR(20) NOT NULL,  
    Password VARCHAR(15) NOT NULL,  
    User_Secret_Question VARCHAR(50) NOT NULL,  
    User_Secret_Answer TEXT NOT NULL,  
    Name VARCHAR(20) NOT NULL,  
    Surname VARCHAR(20) NOT NULL,  
    E-mail VARCHAR(30),  
);
```

Username, Password, Use_Secret_Question, User_Secret_Answer, Name, Surname and E-mail is used for the datas about user.

2.2 ADMIN

Admin has more permission than user in which using the tools of project. For instance, admin can start program, that means program can be started capturing the

files from the network traffic by admin. Furthermore, admin can add database protocols and their keywords. That means, our program has an ability to understand and classificate new protocols with itself, however admin can also add new protocols and their spesific keywords program's database.

Attributes are:

- 9 AdminName
- 10 AdminPassword
- 11 Admin_secret_question
- 12 Admin_secret_answer
- 13 Name
- 14 Surname
- 15 E-mail

SQL code will be like that:

```
CREATE TABLE Admin(  
    AdminName VARCHAR(20) NOT NULL,  
    AdminPassword VARCHAR(15) NOT NULL,  
    Admin_Secret_Question VARCHAR(50) NOT NULL,  
    Admin_Secret_Answer TEXT NOT NULL,  
    Name VARCHAR(20) NOT NULL,  
    Surname VARCHAR(20) NOT NULL,  
    E-mail VARCHAR(30),  
);
```

AdminName, AdminPassword, Admin_Secret_Question, Admin_Secret_Answer, Name, Surname and E-mail is used for the datas about admin.

2.3 KEYWORDS

This table is main part of our project, because all keywords belong to the protocols are hold in here. After decoder phase, searching of which protocol captured is done according to these keywords. Additionally, if new procotol is captured, after analysing its structure, its own keywords are added in this table.

Attributes are:

16 StringName

17ProtocolName

18StringPlace

SQL code will be like that:

```
CREATE TABLE Keywords(  
    StringName VARCHAR(20) NOT NULL,  
    ProtocolName VARCHAR(15) NOT NULL,  
    StringPlace INTEGER NOT NULL,  
    FOREIGN KEY(ProtocolName) REFERENCES ( Protocols (ProtocolName)  
)  
);
```

StringName is the string that is keyword for the protocol, ProtocolName is which protocol string belongs, and finally, StringPlace is in the file where string can be found in file. This attribute is integer, because for easibility, file will be analysed integer by integer.

2.4 PROTOCOLS

Protocol table only holds the name of procotols, user can easily see which protocols can be recognized without additional recognition operation is being done. This table is directly related with the keyword table, because after making keyword analysis, output will be given from this table.

- ProtocolName

SQL code will be like that:

```
CREATE TABLE Protocols(  
    ProtocolName VARCHAR(15) NOT NULL,  
    FOREIGN KEY(ProtocolName) REFERENCES ( Protocols (ProtocolName)  
)  
);
```

2.4.1 FTP (FILE TRANSFER PROTOCOL)

This protocol is used for transferring data. After identification of this protocol, our program should give output as which file is sent through the IP address. As a result structure of table is:

19 ProtocolName
20 FileName
21 IPaddress
22 Date
23 CaptureTime

SQL code will be like that:

```
CREATE TABLE FTP(  
    ProtocolName VARCHAR(15) NOT NULL,  
    FileName VARCHAR(20) NOT NULL,  
    IPaddress VARCHAR(30) NOT NULL,  
    Date VARCHAR(20) NOT NULL,  
    CaptureTime VARCHAR(20) NOT NULL,  
    FOREIGN KEY(ProtocolName) REFERENCES ( Protocols (ProtocolName)  
);
```

2.4.2 SMTP (SIMPLE MAIL TRANSFER PROTOCOL)

This protocol is used for mail transferring. This table contains such as who sent mail is "from", who took mail is "to", and the content of mail is "MailContent".

Attributes are listed:

- ProtocolName

- From
- To
- MailContent
- Date
- CaptureTime

SQL code will be like that:

```
CREATE TABLE SMTP(
    ProtocolName VARCHAR(15) NOT NULL,
    From VARCHAR(20) NOT NULL,
    To VARCHAR(30) NOT NULL,
    MailContent VARCHAR(20) NOT NULL,
    Date VARCHAR(20) NOT NULL,
    CaptureTime VARCHAR(20) NOT NULL,
    FOREIGN KEY(ProtocolName) REFERENCES ( Protocols (ProtocolName)
);
```

2.4.2 HTTP (HYPERTEXT TRANSFER PROTOCOL)

This protocol is a communications protocol used to transfer or convey information on intranets and the World Wide Web. Structure of this protocol looks like File Transfer Protocol. Therefore,

Attributes are:

- 19 ProtocolName
- 20 FileName
- 21 IPadress
- 22 Date
- 23 CaptureTime

SQL code will be like that:

```
CREATE TABLE FTP(  
    ProtocolName VARCHAR(15) NOT NULL,  
    FileName VARCHAR(20) NOT NULL,  
    IPadress VARCHAR(30) NOT NULL,  
    Date VARCHAR(20) NOT NULL,  
    CaptureTime VARCHAR(20) NOT NULL,  
    FOREIGN KEY(ProtocolName) REFERENCES ( Protocols (ProtocolName)  
);
```

Finally, these three protocols are the basic protocols, we firstly make our project according to their structures, after that we add new protocols time by time.

3 PROJECT REQUIREMENTS

The most important part of the project works is determining the requirements. The requirements analysis must be done carefully to get project maintenance. Therefore, during the development of the project, it must be guaranteed that no unexpected events and requirements should occur. In order to avoid failure, requirements must cover all our needs. For our project, requirements are divided in three main issues.

3.1 System Requirements

3.1.1-Hardware Requirements

Minimal hardware requirements for our project are: A PC with the following configuration will be needed:

A minimum of 1 GB DDRAM

A minimum of 50 GB free space on hard disk, for huge database storage

A Pentium IV processor

Internet Connection

Local or Wide Area Network

3.1.2-Software Requirements

We will use several tools for different phases during development of the project.

These

development phases are;

☐ Documentation Part

- Analysis

- Design

☐ Develeopment Part

- Implementation

- Testing

Documentation Tools

Microsoft Office 2003 Word

Adobe Acrobat Professional

SmartDraw

Development Tools

Windows XP

C++

WireShark(WireShark is a sniffer program as it mentioned before.Usage of wireshark's purpose is testing and also to create PCAP files)

3.2 Functional Requirements

- In our Project, required protocols that should be identified are:
 - HTTP
 - SIP
 - POP3
 - FTP
- The data which is captured from network, which will be identified, is to be fed in the PCAP file format.
- Our identification process will also determine when the identified protocol is no longer available in the flow through the identified port.
- When we identificate protocols our outputs will be like these:
 - HTTP : If connected HHTTP server is Yahoo! Mail and user downloads a mail, then RFC 822 complied e-mail messages(.eml).
 - POP3 : Download mail messages.
 - FTP : Transferred files.
 - SIP : Voice files in Microsoft ASF format.

3.3 Nonfunctional Requirements

Usability:

As for all softwares, graphical user interface is very important for our project too since easy to use, easy to learn and adaptation is crucial for all softwares. Therefore, we will make the user interface clear and understandable. Moreover, graphical user interface will meet the user's needs as far as it is possible. Hence, while we were

deciding our tool's features, we took into consideration both the inexperienced and professionals. As a conclusion, we considered the user satisfaction as the primary goal of our project.

Reliability:

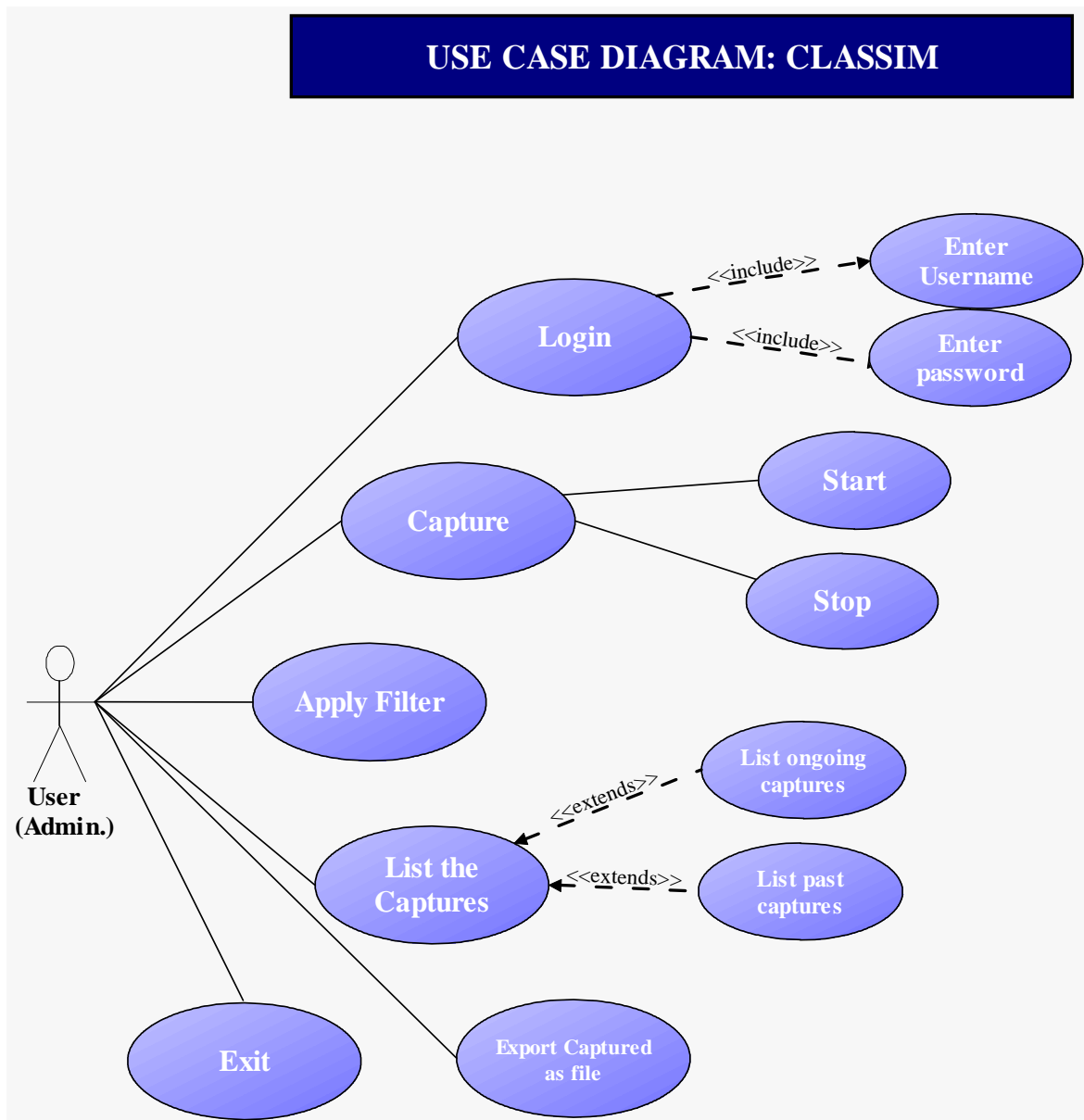
The system should be as bug free as possible. All sub components should work asynchronously, so that any delay caused by one of the components should not block other components work on its own. Moreover, we plan to do many tests after implementation to minimize the bugs on the program.

Performance:

Speed of our product becomes an important issue in our design. In general, wrong programming methodologies that are used in the applications slow down the programs, not the complex algorithms used in the applications. Therefore, we will try not to do this mistake. The usage of system resources will be reduced as much as possible to increase the performance of our design. User can run other applications easily while our program working.

4.SYSTEM DESIGN

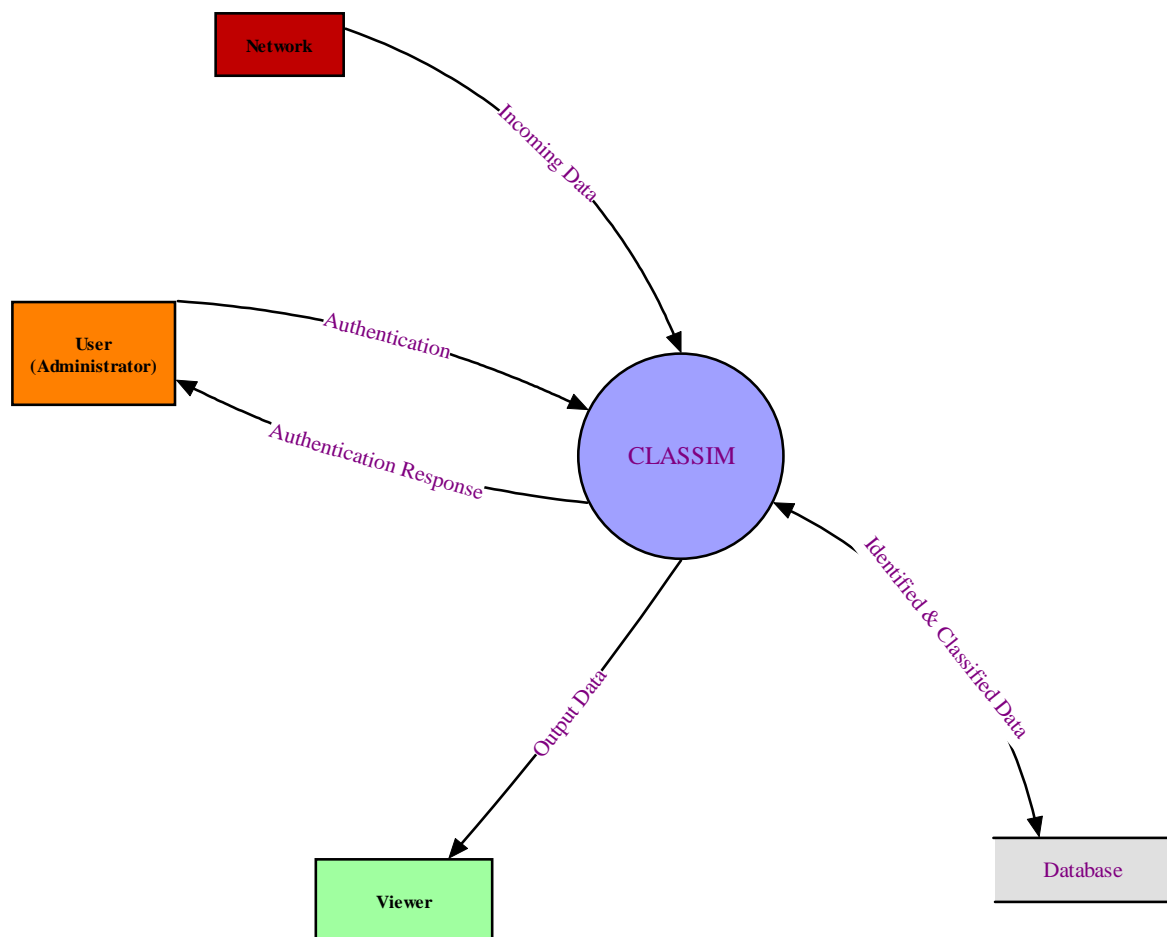
4.1 USE CASE DIAGRAMS



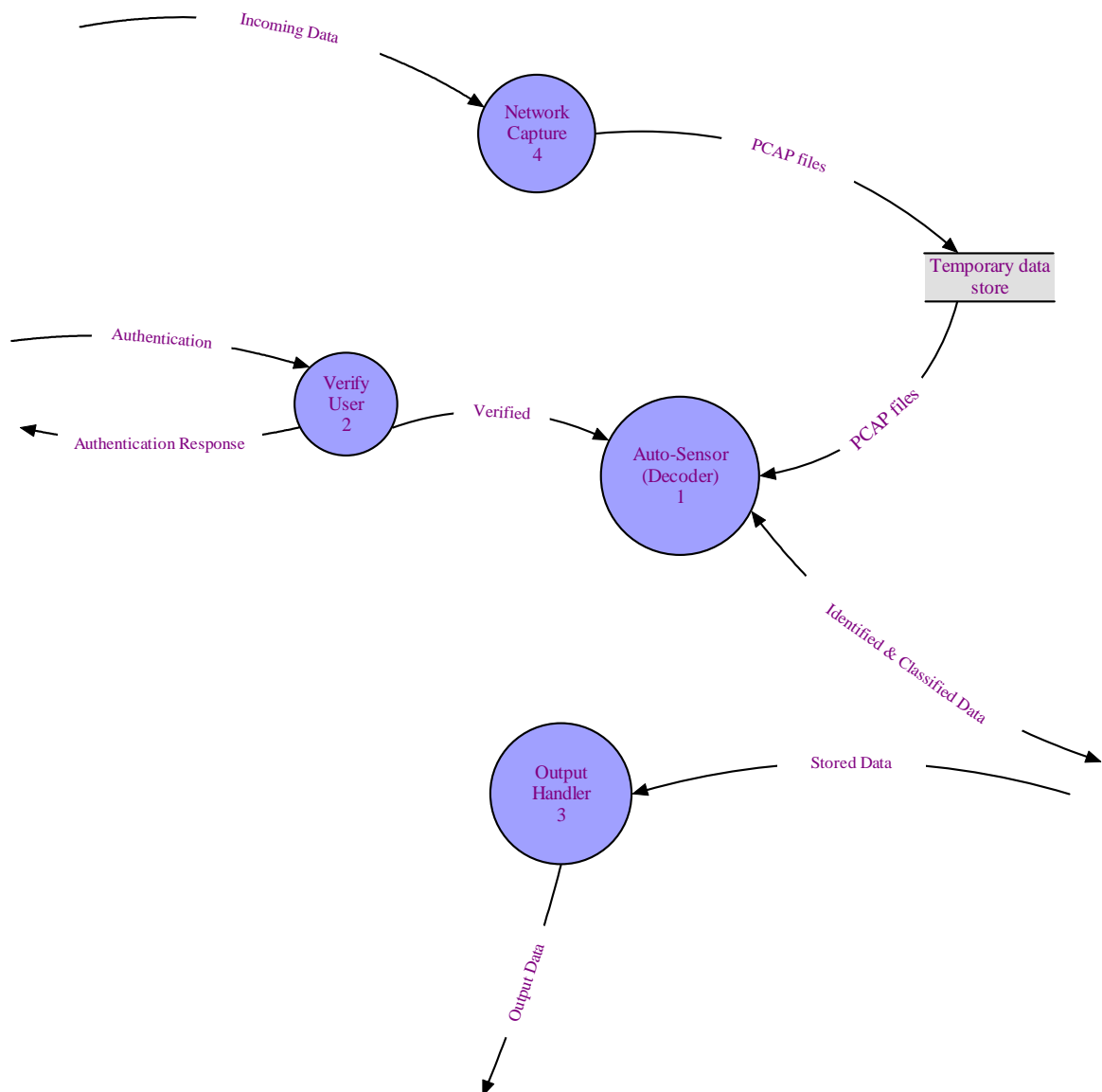
4.2 DFD DIAGRAMS

4.2.1 DIAGRAMS

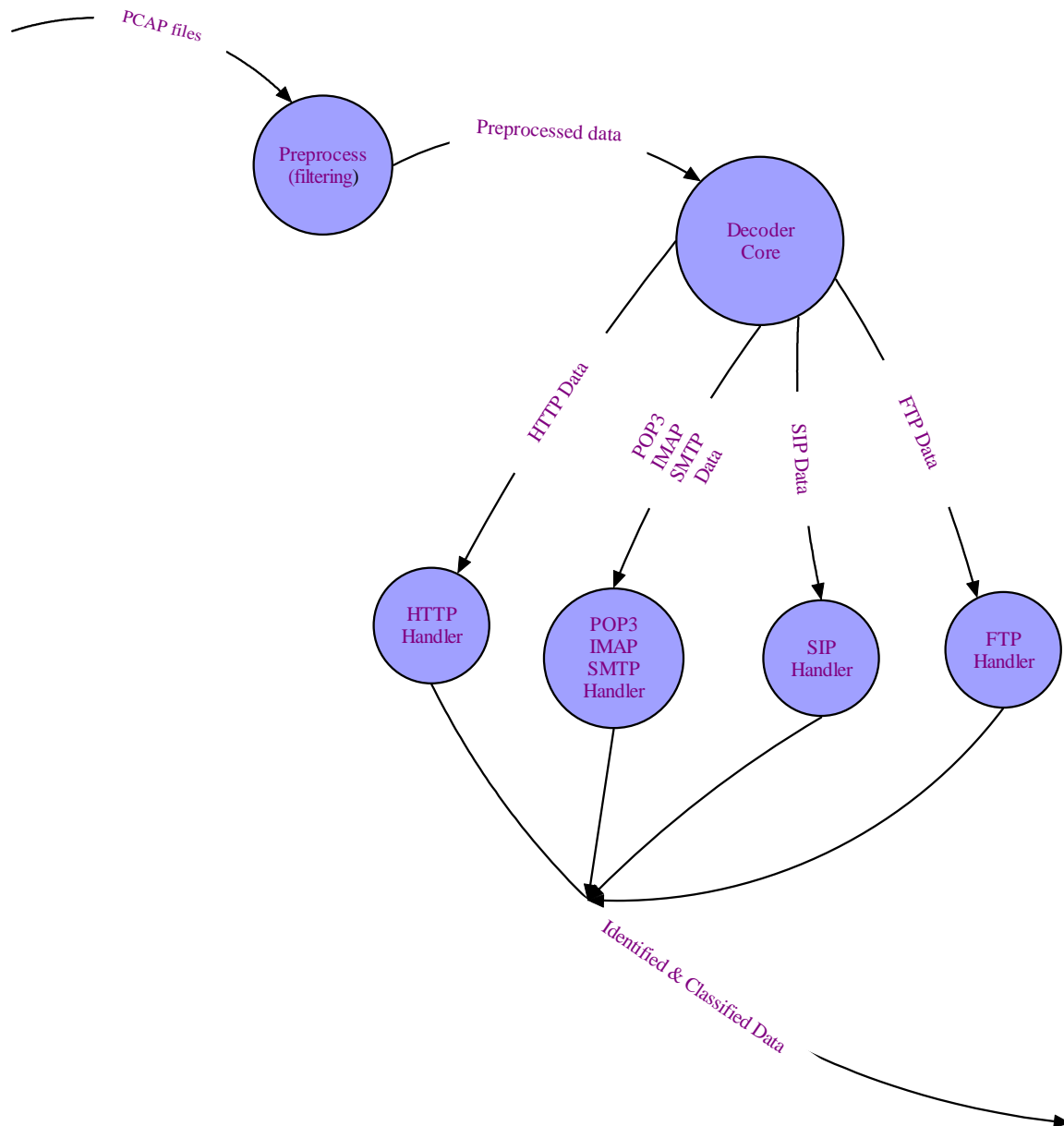
DFD0



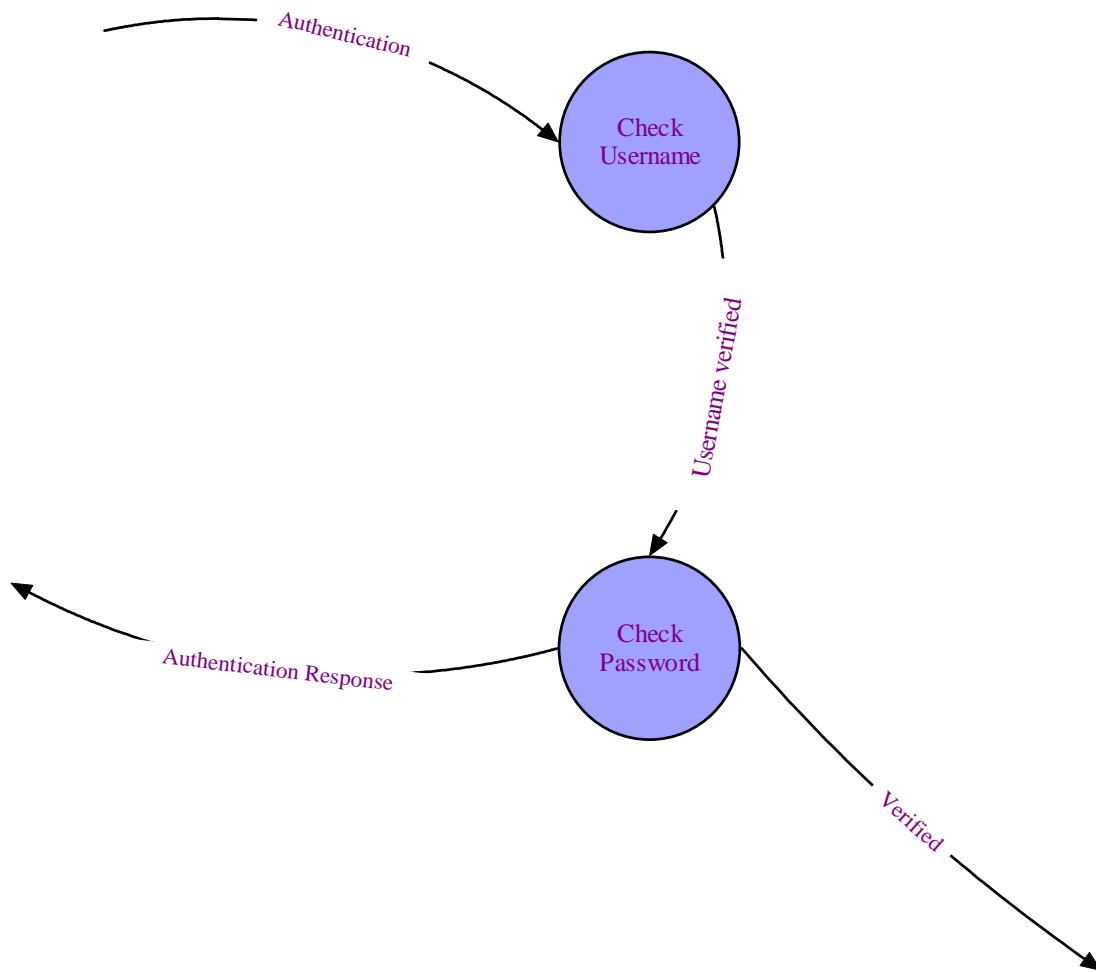
DFD1-CLASSIM



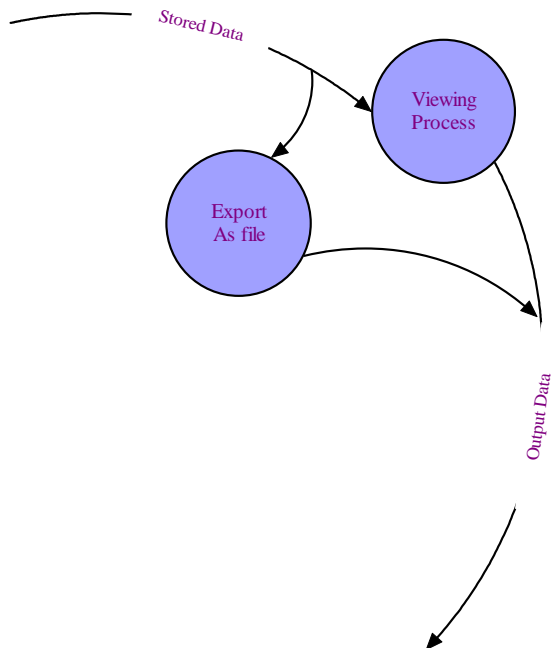
DFD2-Decoder



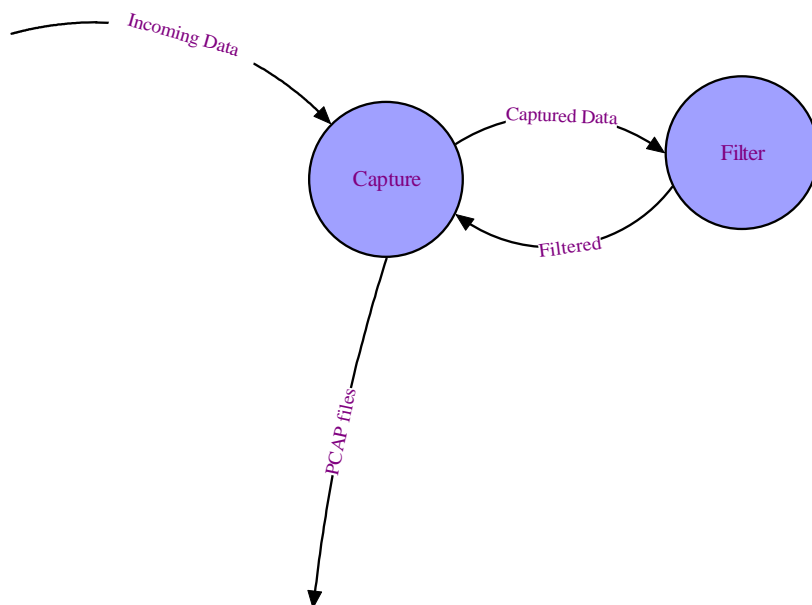
DFD2- Verify User



DFD2- Output Handler



DFD2- Network Capture



4.2.2 - DFD Data Dictionary

Name : Incoming Data

Input for: Network Capture

Output for : Network (external)

Description : The Data coming from Network that will be captured as PCAP.

Name : PCAP files

Input for: Auto-Sensor , Temporary data store

Output for : Network Capture

Description : The file which has PCAP format that identifying will be done according.

Name : Identified & Classified Data

Input for: Database, output handler

Output for : Auto-Sensor , database

Description : Includes identified protocol type, source port, destination port, time info, transmitted data etc.

Name : Output Data

Input for: viewer

Output for : Output Handler

Description : data to be outputted in the format determined according to the identified protocol e.g. asf format for SIP protocol.

Name : HTTP Data

Input for: HTTP Handler

Output for : Decoder Core

Description : The identified data in the pattern classification part that is recognized as HTTP in the decoder.

Name : SMTP/POP3/IMAP Data

Input for: SMTP/POP3/IMAP Handler

Output for : Decoder Core

Description : The identified data in the pattern classification part that is recognized as SMTP/POP3/IMAP in the decoder.

Name : SIP Data

Input for: SIP Handler

Output for : Decoder Core

Description : The identified data in the pattern classification part that is recognized as SIP in the decoder.

Name : FTP Data

Input for: FTP Handler

Output for : Decoder Core

Description : The identified data in the pattern classification part that is recognized as FTP in the decoder.

Name : Authentication

Input for: User verify

Output for : User

Description : The information specific to the User(Username and Password)

Name : Authentication Response

Input for: User

Output for : User verify

Description : The response of verification as verified username and password or not.

4.2.3 – DFD Process Description

Name : Auto-Sensor

Input : PCAP files

Output : Identified & Classified Data

Description : This process takes PCAP files and decodes the information which is in PCAP format. According the decoding, incoming information sent to the protocol handlers. After protocol handlers, data becomes ready for storing and sending to output.

Name : Output Handler

Input : Identified & Classified Data

Output : Output Data

Description : This process gets the identified data. According to the protocol of data, process uses the required output format. Looking the protocol type, outputs the transmitted data in wanted format. For example in pop3 protocol outputs the mail message in 'eml' format. After all it sends the output data to viewer.

Name : Network Capture

Input : Incoming Data

Output : PCAP files

Description : This process gets the incoming data from network. After optional filtering process it generates PCAP files for our classifier to be used.

Name : User Verify

Input : Authentication

Output : Authentication Response

Description : This process gets the username and password of the User and gives permission to the User to pass through the program if he/she has the right to use the program.

Name : HTTP Handler

Input : HTTP Data

Output : Identified & Classified Data

Description : This process gets the data from decoder that has been classified to be in HTTP protocol. Process analyzes coming data according to RFC document of HTTP and decomposes the data.

Name : FTP Handler

Input : FTP Data

Output : Identified & Classified Data

Description : This process gets the data from decoder that has been classified to be in FTP protocol. Process analyzes coming data according to RFC document of FTP and decomposes the data.

Name : SMTP/POP3/IMAP Handler

Input : SMTP/POP3/IMAP Data

Output : Identified & Classified Data

Description : This process gets the data from decoder that has been classified to be in SMTP/POP3/IMAP protocol. Process analyzes coming data according to RFC document of SMTP/POP3/IMAP and decomposes the data.

Name : SIP Handler

Input : SIP Data

Output : Identified & Classified Data

Description : This process gets the data from decoder that has been classified to be in SIP protocol. Process analyzes coming data according to RFC document of SIP and decomposes the data.

4.3. CLASS DIAGRAMS

4.3.1-User Verification

User verification part of our project will only allow the users that is registered to the system by admin. Login class will have a method to get the username and password.

Another method that reach the database and handle the requested query will be implemented.

We will also implement a method to verify that entered username and password belongs to an existing user.

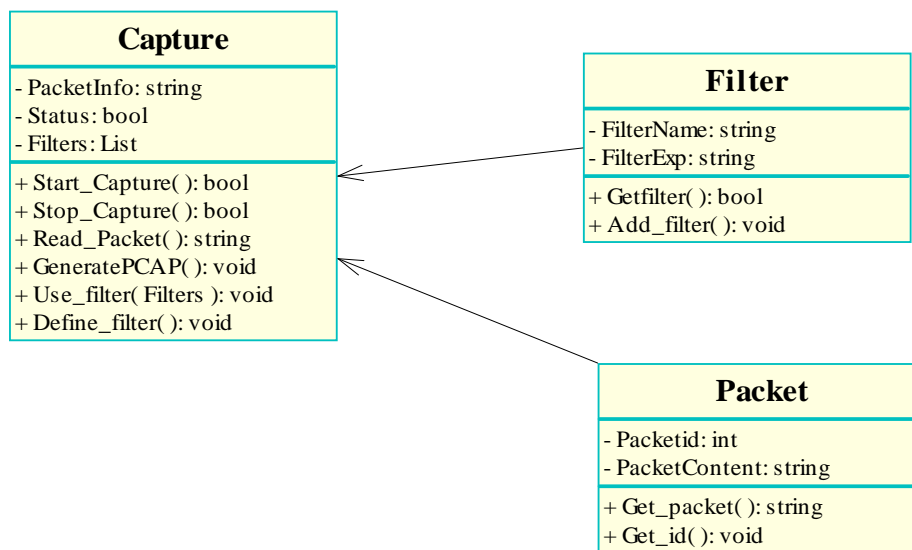
Our class will look like this:

Login
-Username: string -Password: string
+ getUsername(): string + setUsername(): void + getPassword(): string + setPassword(): void + Handle_Database_Query(): void + verify(): bool

4.3.2 Network Capture

'Network capture' will be pcap provider of our decoder. We will have methods to start and stop capturing. This class will read incoming packets and generate us the pcap files that we will use in classification part.

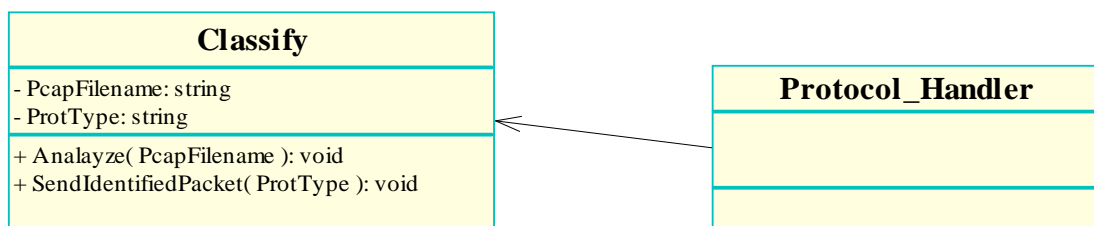
Classes will look like this:



4.3.3 Decoder

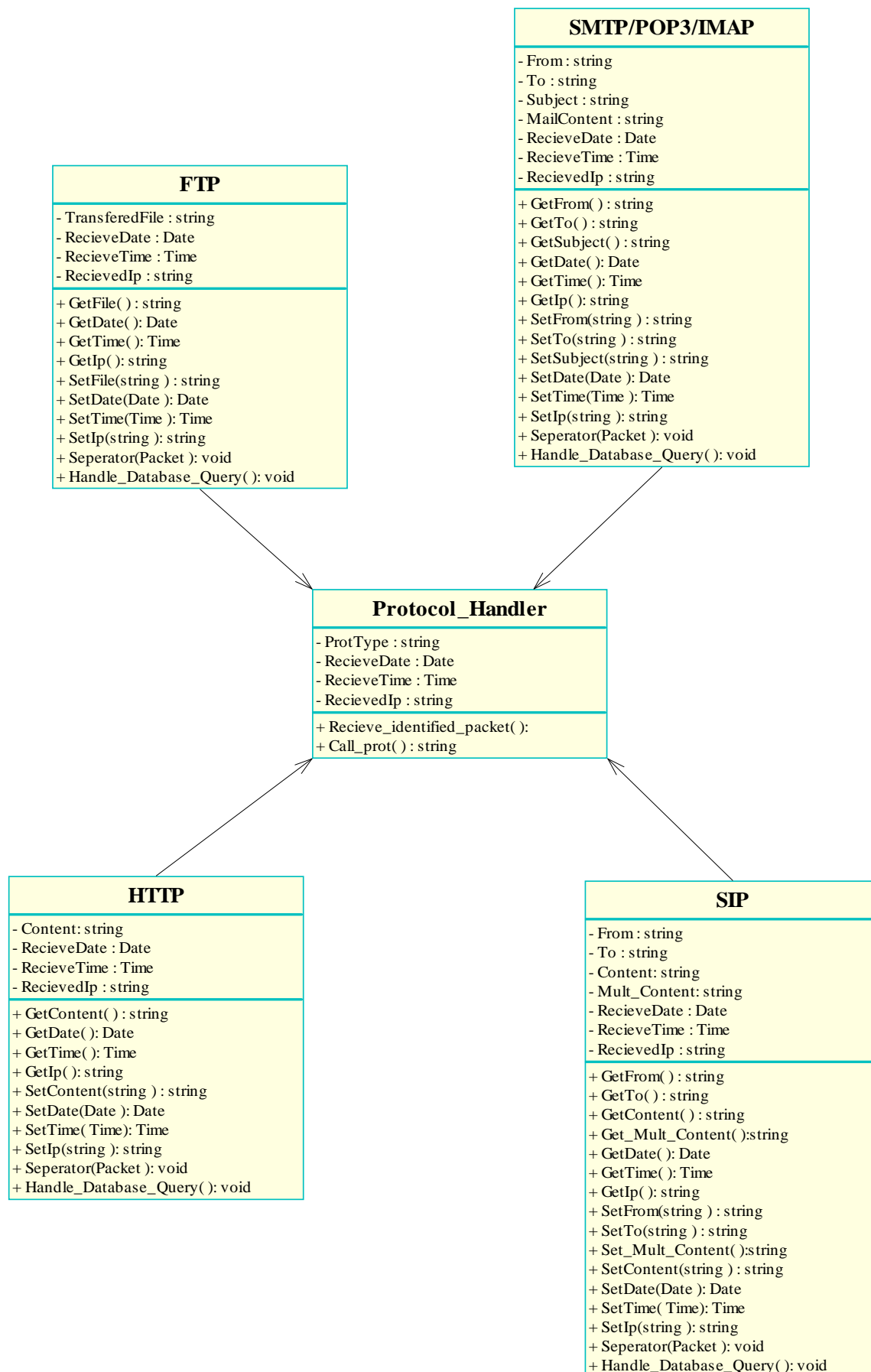
Decoder is the main part of the project. Classification without having port info will be done in this phase. This part includes Preprocessing, Feature analysis, Pattern classification.

Decoder phase will be detailed in Final Design Report. Related class definition will also be added.

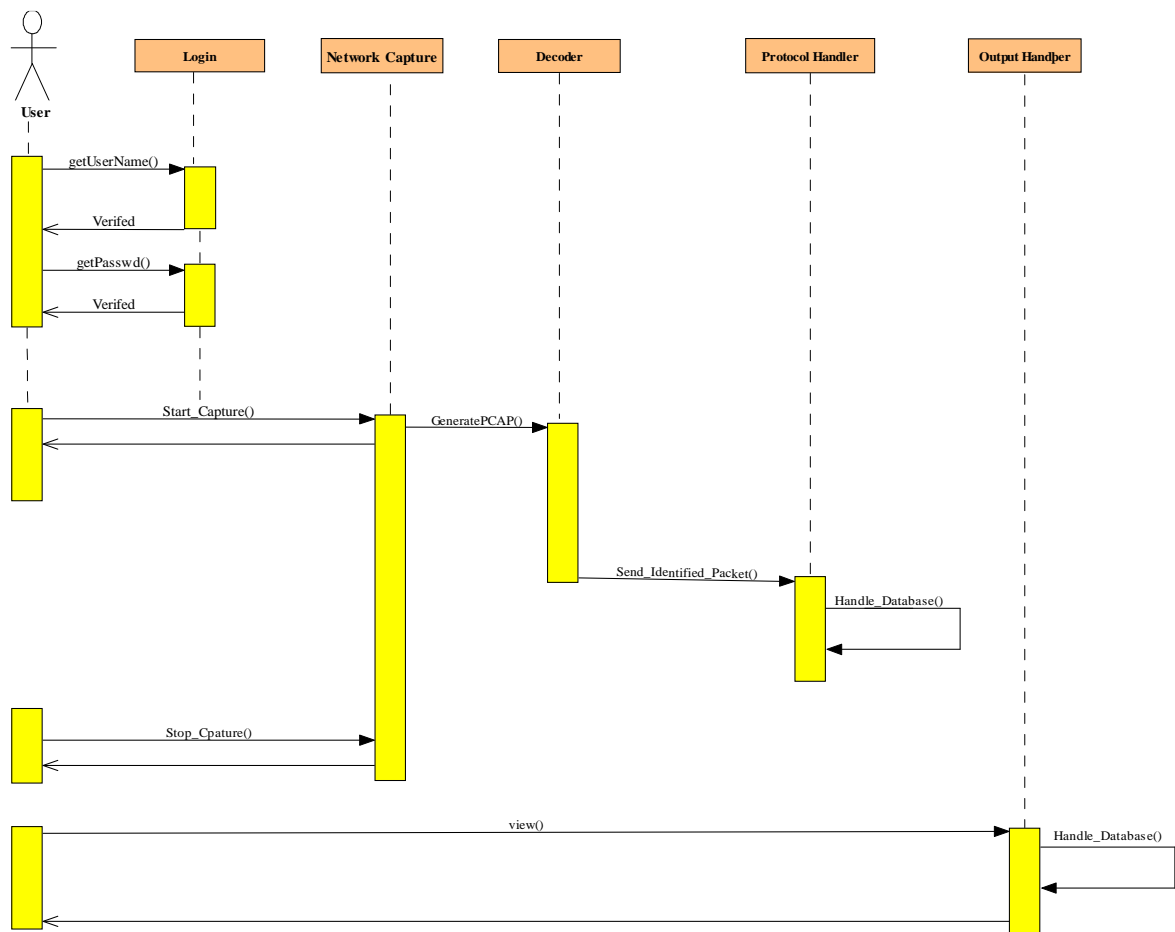


4.3.4 Protocol Handlers

Classified packets will be sent to the related Protocol Handler and will be analyzed according to the RFC of this protocol. Each protocol handler will use a method to analyze the packet and get the necessary information. And each protocol will have a method to reach the database and perform requested query.

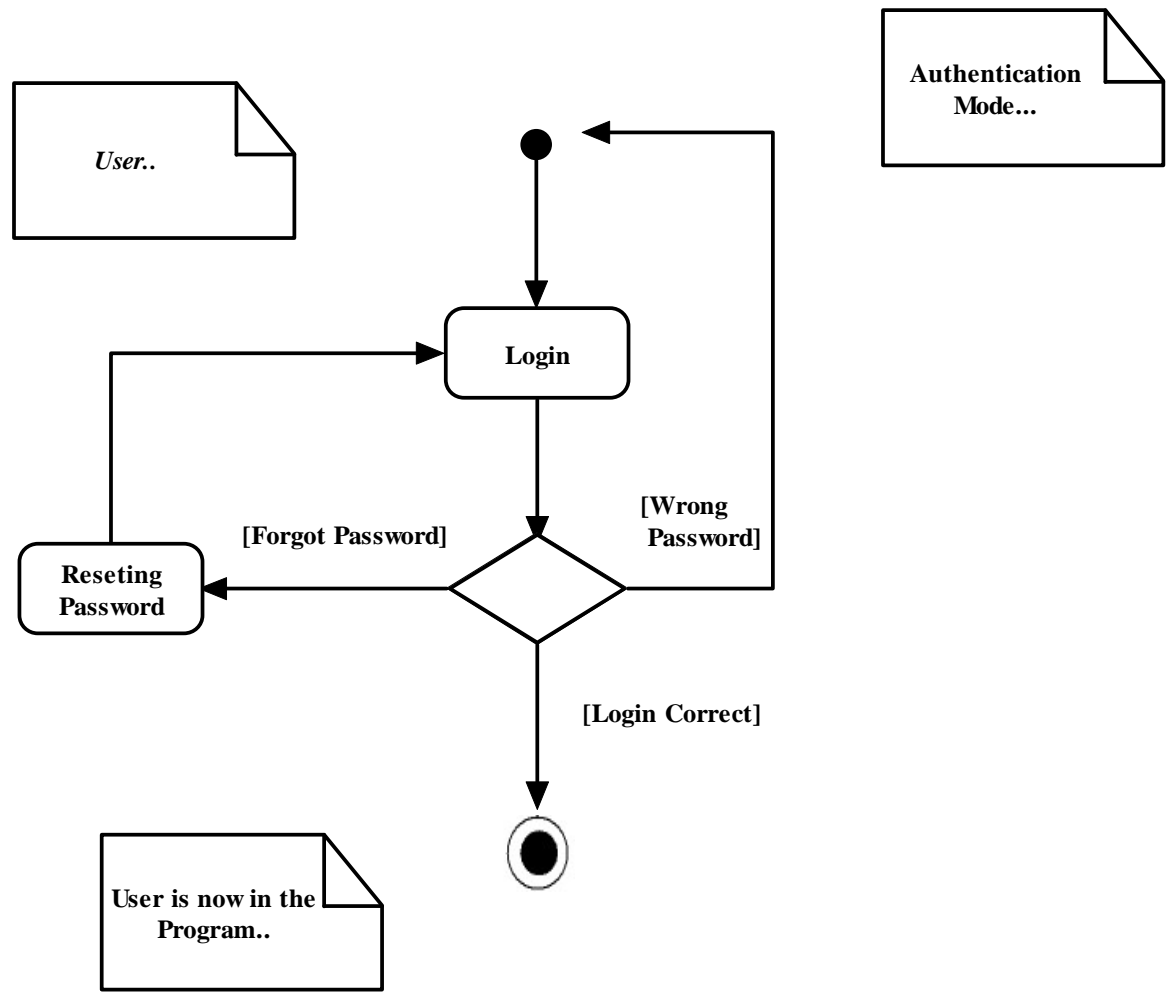


4.4. SEQUENCE DIAGRAM

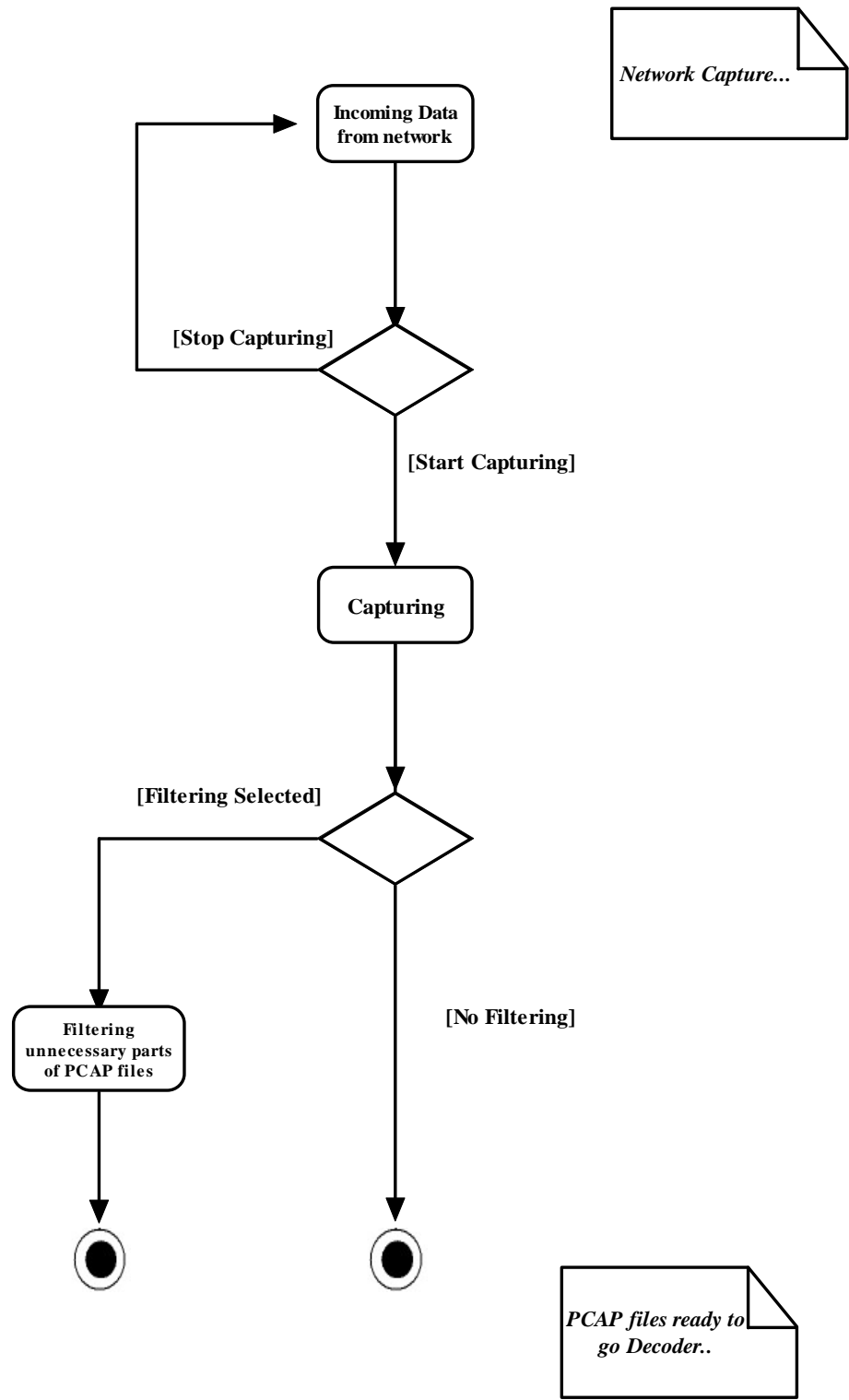


4.5.ACTIVITY DIAGRAMS

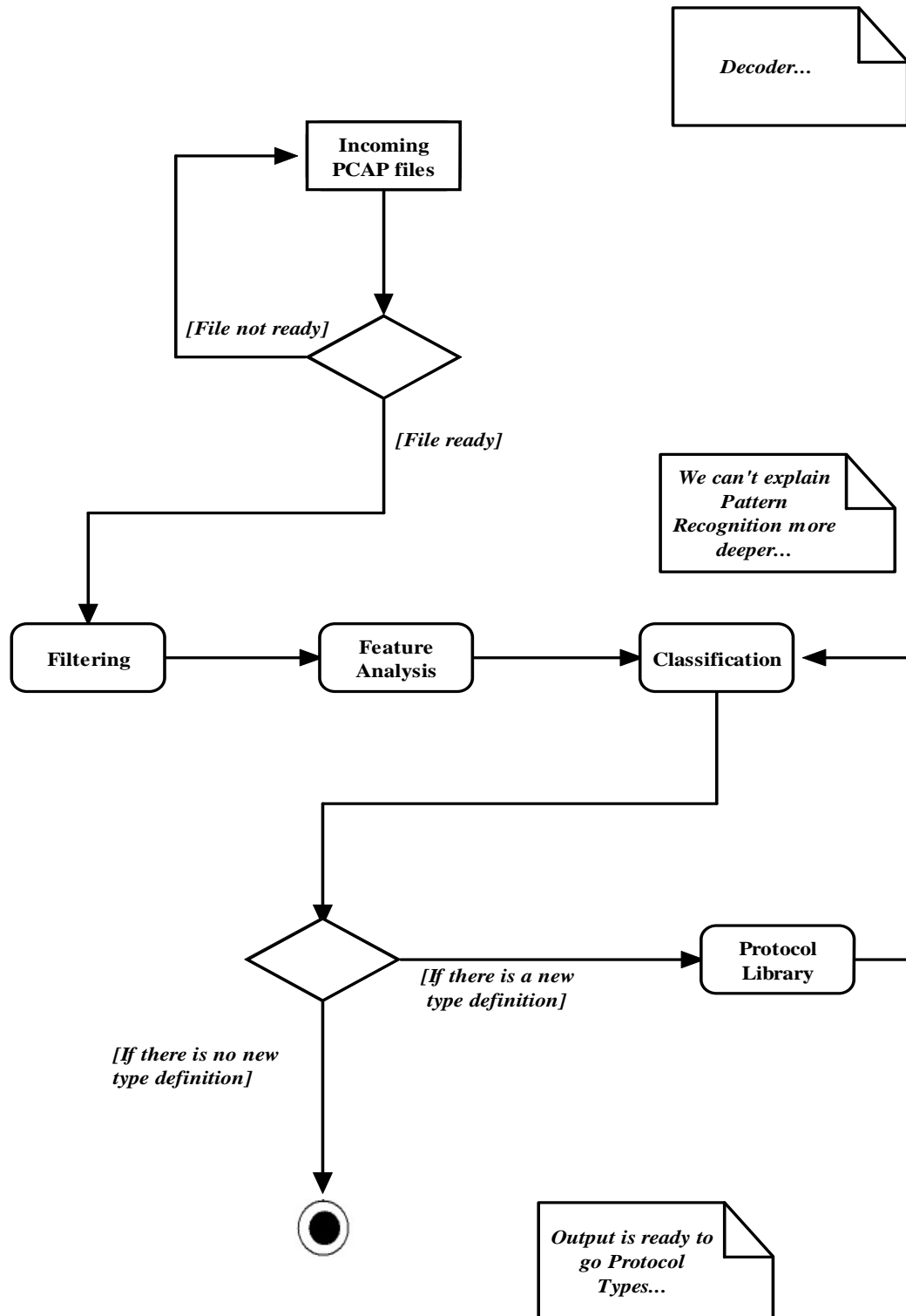
4.5.1 Login



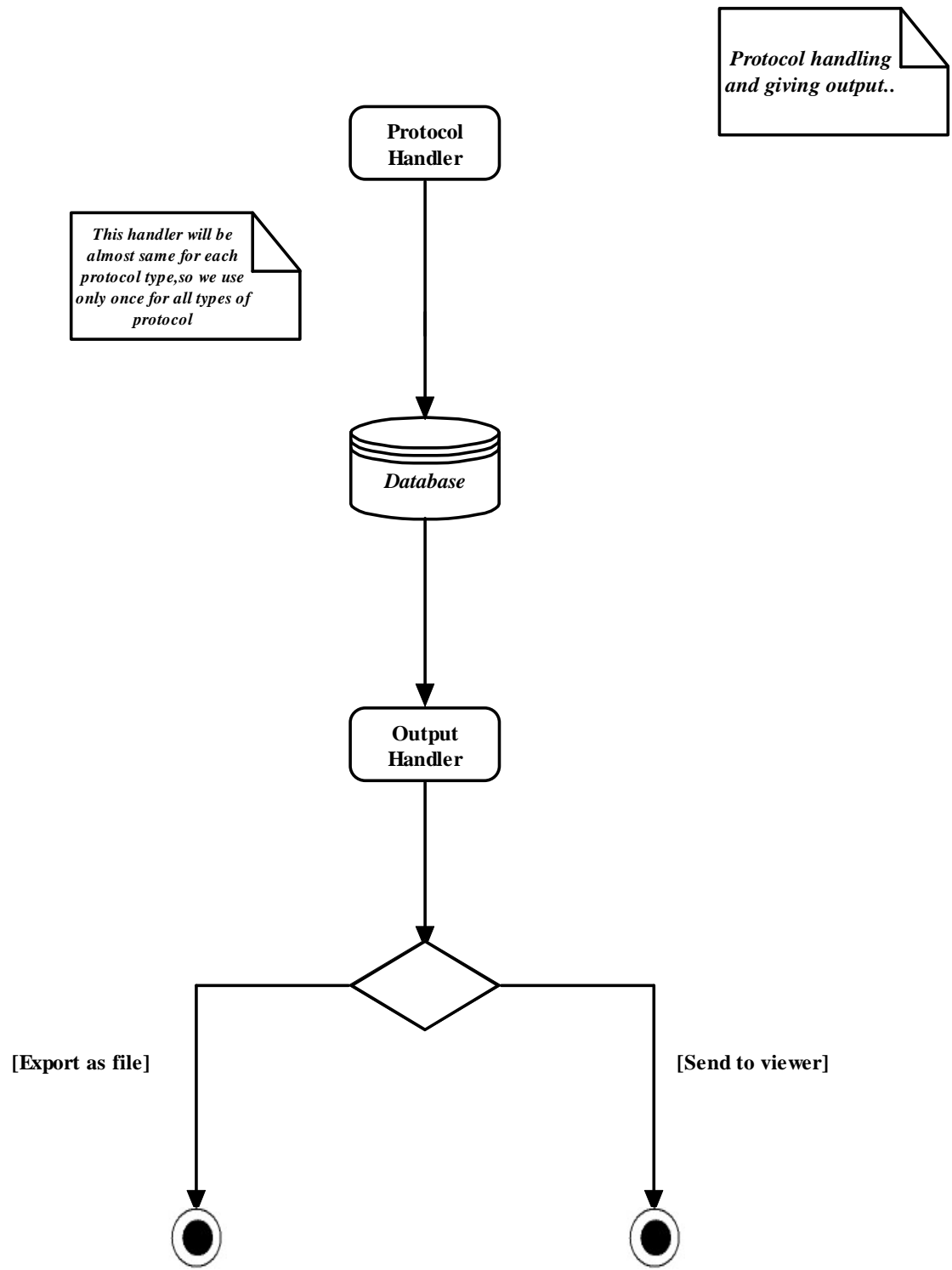
4.5.2 Capturing Data



4.5.3 Decoder



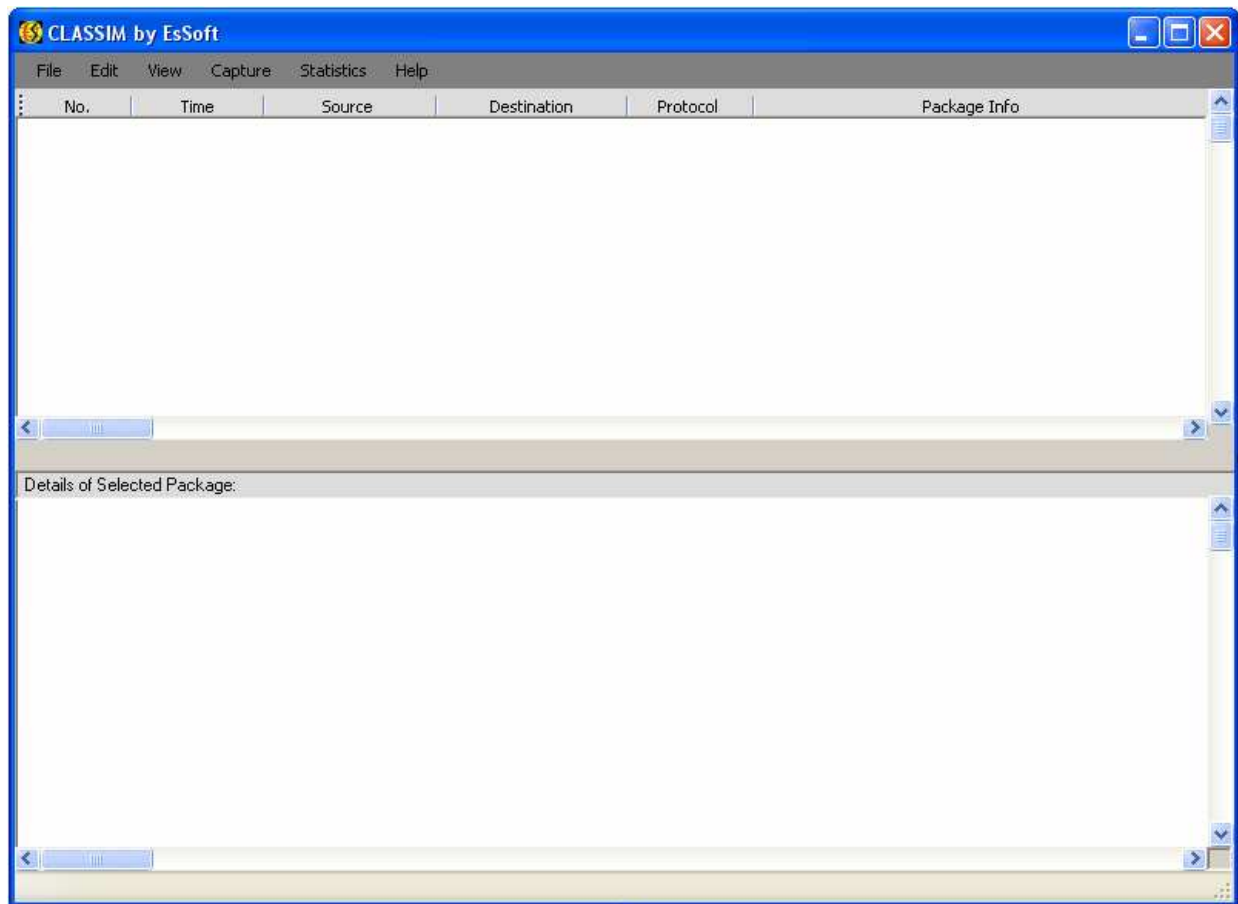
4.5.4 Protocol Handling and Giving Output



5. User Interface

5.1 General Overview

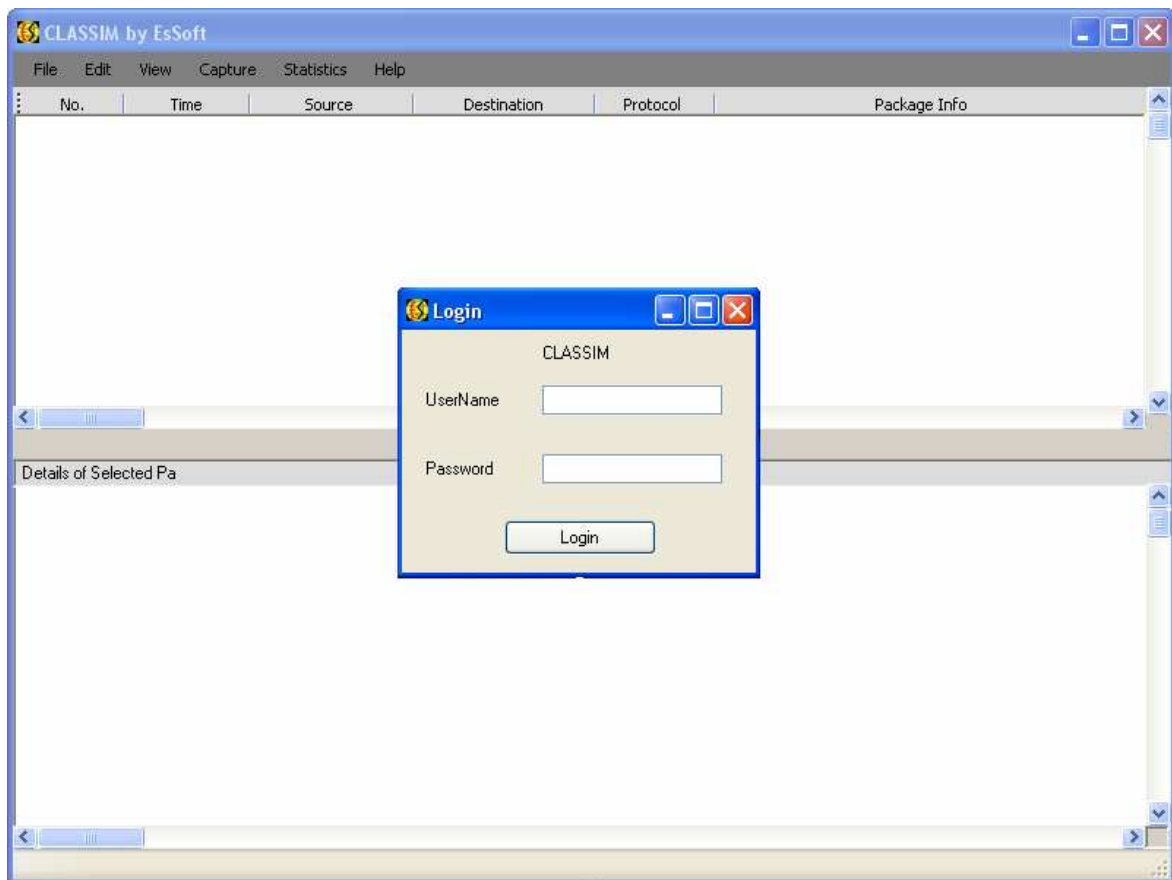
The Prototype of CLASSIM's MAIN GUI will be look as below:



As seen from this screenshot, GUI includes two panel. One of the panel is captured packages panel and the other panel is details of the selected captured

packages. Also there is a menu bar that provide user to view other informations about selected captured packages. All these part of the GUI mentioned on 5.2 Menu Bar part, 5.2 and 5.4 Panels.

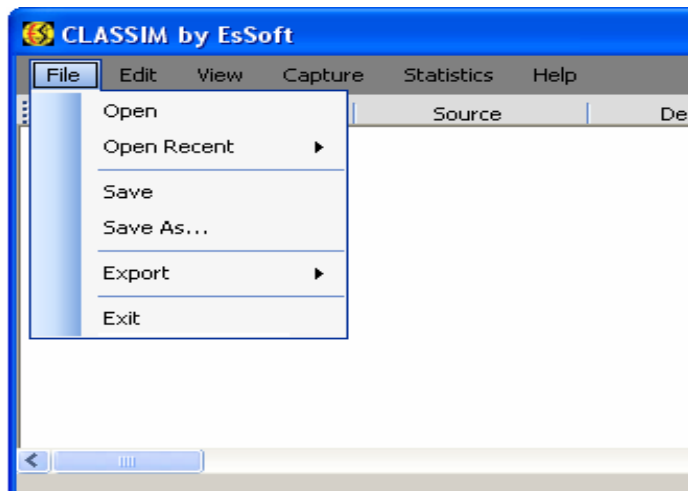
5.2 LOGIN



When program starts Login Window will appear in the first coming screen. User have to enter the correct username and password to log in.

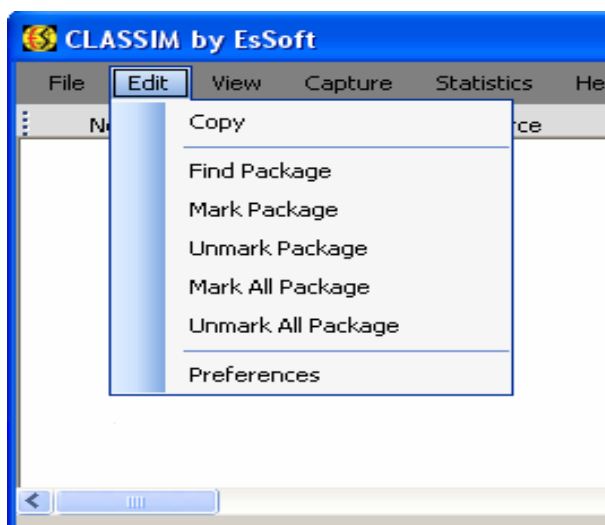
5.3 MENUBAR

5.3.1 FILE MENU



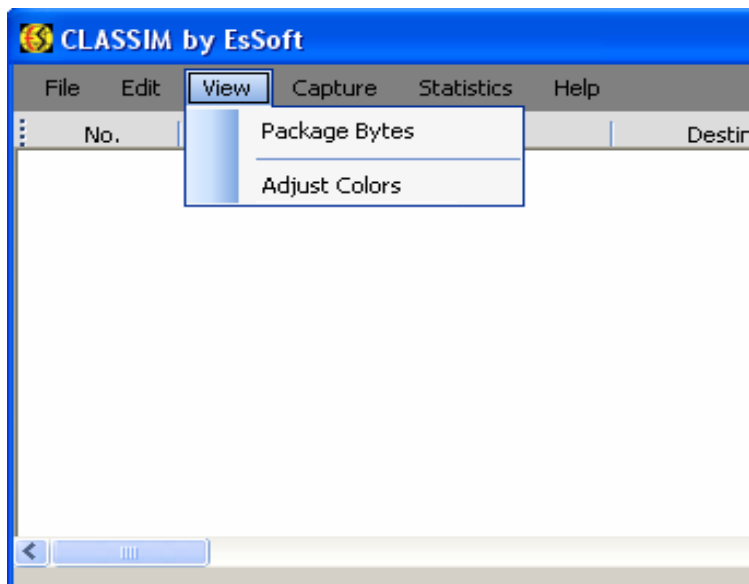
By this menu user can open an existing files by “Open”. User can open recent saved files by using “Open Recent”. User can save captured file list by “Save” or “Save As”. By “Export” user can save selected package's .eml,.asf,.avi vs. files if selected package contents these type of files . User exits the program with “Exit”.

5.3.2 EDIT MENU



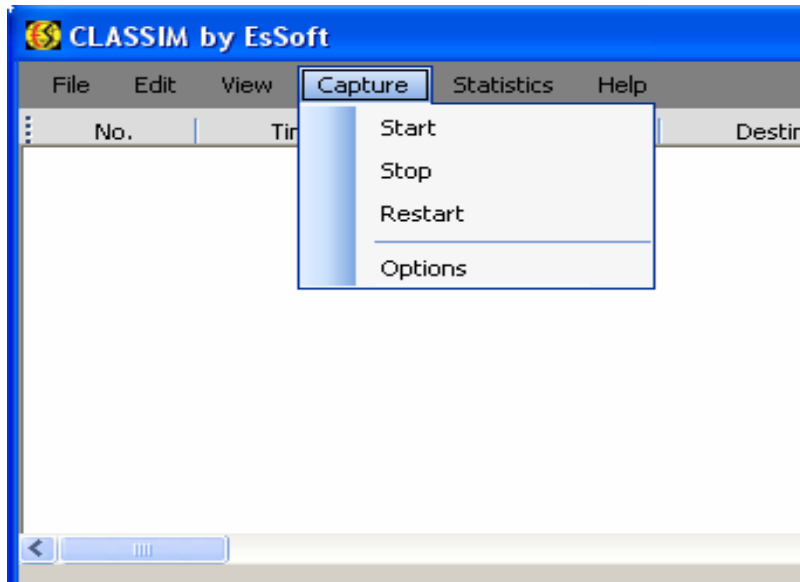
In edit menu user can copy the information of the captured files or details of the selected file. User can find a package with specifying its protocol or capture date or its package content by “Find Package”. Mark, unmark a package or mark, unmark all packages by “Mark Package”, “Unmark Package”, “Mark All Packages”, “Unmark All Packages”. And by “Preferences” user can change the preferences of the user interface.

5.3.3 VIEW MENU



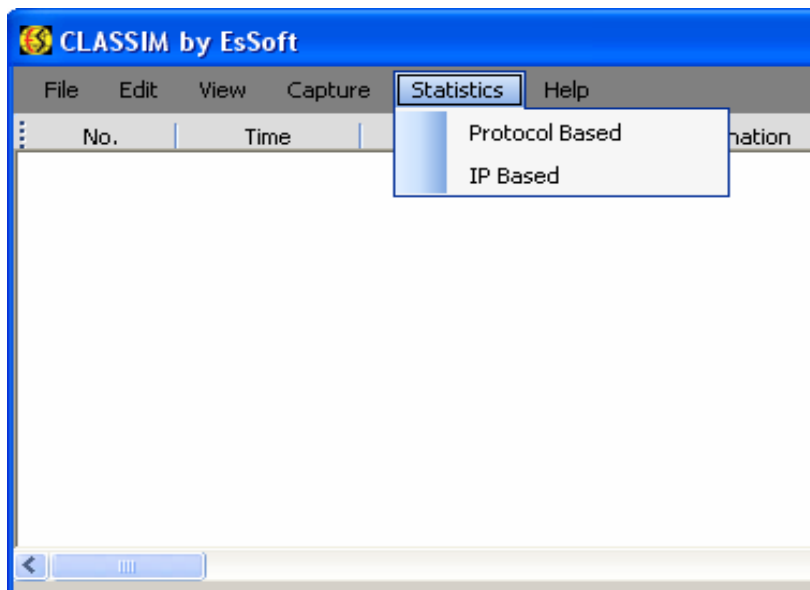
User can view the package content in hexadecimal form by “Package Bytes”. And by “Adjust Colors” user can change the color of the view of the captured packages.(for example : can assign different color to captured packages according to captured packages’ protocols.)

5.3.4 CAPTURE MENU



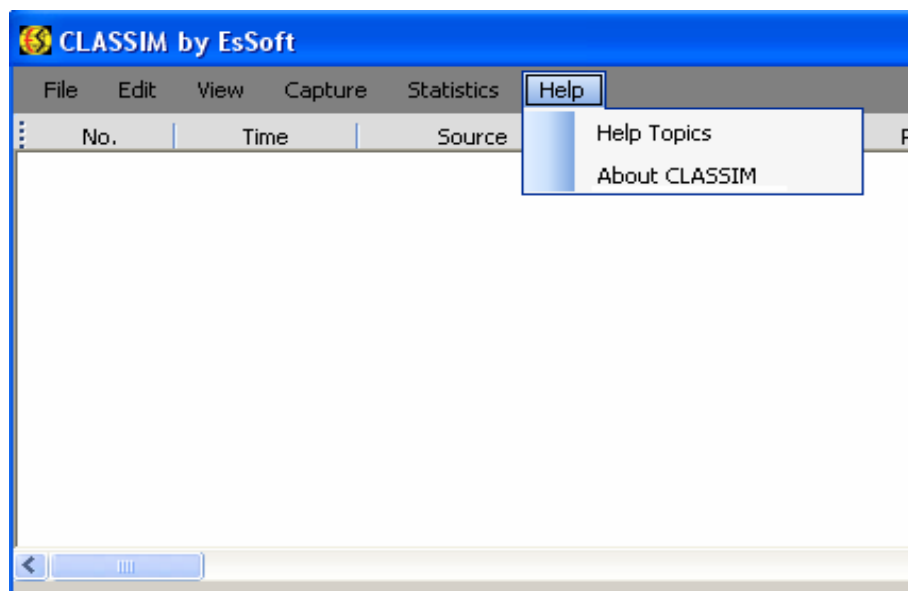
User can start capturing by “Start”. User can stop or restart the capturing by “Stop”, “Restart”. And also user can change capturing options by “Options” (for example : user can filter the protocols, IP’s, etc..)

5.3.5 STATISTIC MENU



By statistic menu user can view the statistic based on protocols or IP's. Protocol based statistics can be viewed by "Protocols Based". IP based statistics can be viewed by "IP Based".

5.3.6 HELP MENU



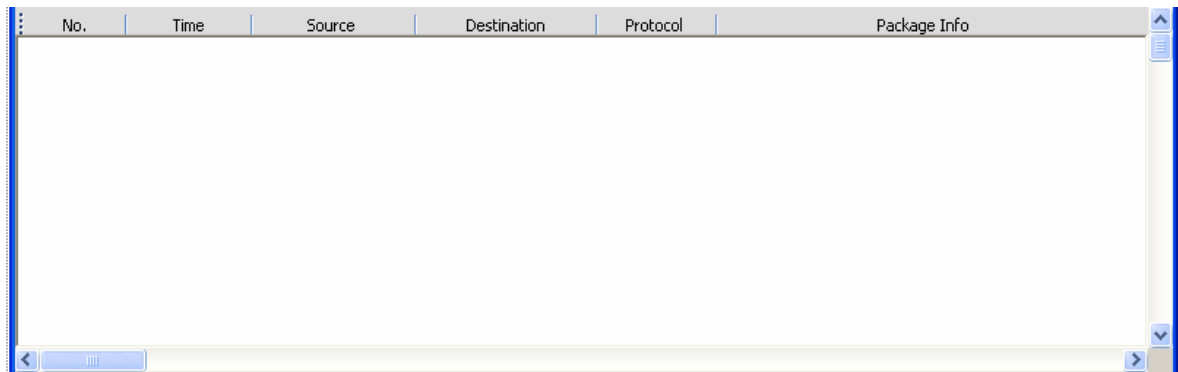
User can view "Help Topics" or "About CLASSIM".

5.4 TOOLBAR

Tool Bar will be implemented in the Final Design Report. Use will be able to select the tools he/she wants to use.

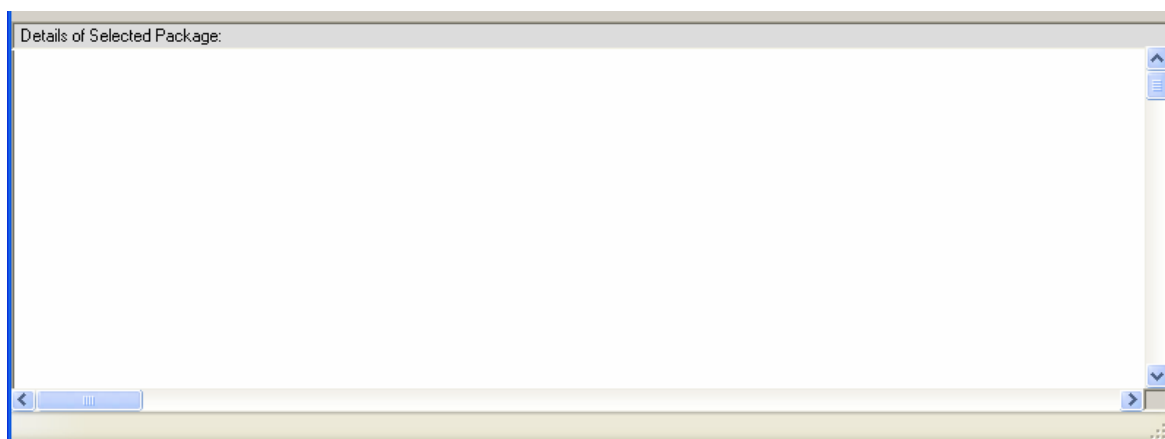
5.5 PANELS

5.5.1 CAPTURED PACKAGES PANEL



In Captured Packages Panel, packages which are captured according to filter is viewed. Captured packages' package info, protocol, destination etc. are also viewed in this panel.

5.5.2 DETAILS of CAPTURED PACKAGE



In Details of Captured Package, the selected package's content is viewed.

6. PROJECT SCHEDULE

Gantt Chart is provided in Appendix A.

7. TESTING STRATEGIES

This initial design plan will be improve in next design phases.

Parts of the project will be tested for functionality and performance before integration.

Tests:

- Can the administrator log in to the system securely?
- Are the packages correctly captured from network traffic?
- Are the packages decoded correctly?
- Does the decoder performs pattern recognition well?(most important)
- Is the package information output correctly?
- Does the interface work correctly?

8. CONCLUSION

Thanks to development of computer technology, network systems become complex and sometimes make problems for users that have to be dealed. For instance, increase of usage of internet in the world, control of network systems is a critical issue. Hardwares and softwares to connect one computer to the other have to do identification and classification of programs in terms of protocols.

Through the whole process, we appreciate needs about network systems. We put different features to our project in order to make it more efficient in terms of time and easibility.

APPENDIX A

Number	Task	Start	End	Duration	Q4 - 2007				Q1 - 2008				Q2 - 2008			
					October	November	December	January	February	March	April	May	June			
1	Project Topic Selection	10/17/2007	10/24/2007	5	<div><div></div></div>	<div><div>K,G,M,A</div></div>										
2	Topic Research	10/25/2007	11/2/2007	6		<div><div>K,G,M,A</div></div>										
3	Proposal Report	10/28/2007	11/6/2007	6		<div><div>K,G,M,A</div></div>										
4	Market Research	11/6/2007	11/14/2007	6		<div><div></div></div>	<div><div>K</div></div>									
5	Similar Product Analysis	11/3/2007	11/22/2007	13		<div><div></div></div>	<div><div>K,A</div></div>									
6	Internet Research of Product	11/3/2007	12/13/2007	28		<div><div></div></div>	<div><div></div></div>				<div><div>K,G,M,A</div></div>					
7	Data Dictionary	11/13/2007	11/27/2007	10		<div><div></div></div>	<div><div>M</div></div>									
8	Requiremental Analysis Report	11/23/2007	12/7/2007	10		<div><div></div></div>	<div><div>G,M,A</div></div>									
9	Data Design	11/29/2007	12/6/2007	5		<div><div></div></div>	<div><div>M,K</div></div>									
10	User Interface Design	12/3/2007	12/12/2007	7		<div><div></div></div>	<div><div>G</div></div>									
11	Reading the Protocol Documents (RFC)	10/23/2007	3/11/2008	100		<div><div></div></div>	<div><div></div></div>									
12	Component-Level Design	12/9/2007	12/21/2007	9		<div><div></div></div>	<div><div></div></div>					<div><div>M,K</div></div>				
13	Initial Desing	12/6/2007	12/19/2007	9		<div><div></div></div>	<div><div></div></div>					<div><div>K,G,M,A</div></div>				
14	Detailed Data Design	12/15/2007	12/28/2007	9		<div><div></div></div>	<div><div></div></div>					<div><div>M,K</div></div>				
15	Detailed User Interface Desing	12/18/2007	12/29/2007	9		<div><div></div></div>	<div><div></div></div>					<div><div>G</div></div>				
16	Desing Class Hierarchy	12/22/2007	1/8/2008	11		<div><div></div></div>	<div><div></div></div>					<div><div>A</div></div>				
17	Small Prototype Implementation	12/12/2007	12/19/2007	5		<div><div></div></div>	<div><div></div></div>					<div><div>K,G,M,A</div></div>				
18	Final Desing Report	1/7/2008	1/12/2008	5		<div><div></div></div>	<div><div></div></div>	<div><div></div></div>					<div><div>K,G,M,A</div></div>			
19	Prototype Implementation	1/12/2008	1/18/2008	4		<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>				<div><div>K,G,M,A</div></div>			
20	Development of Prototype	1/19/2008	1/24/2008	3		<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>				<div><div>K,G,M,A</div></div>			
21	Prototype Demo	1/21/2008	1/26/2008	5		<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>				<div><div>K,G,M,A</div></div>			
22	Pattern Recognition	1/30/2008	3/8/2008	28		<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>			<div><div>A,M</div></div>			
23	Decoder Design	3/6/2008	4/3/2008	20		<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>			<div><div>G,K</div></div>				
24	Database Management	4/1/2008	4/19/2008	14		<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>			<div><div>A</div></div>				
25	Graphical Interface Design	4/23/2008	5/9/2008	12		<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>			<div><div></div></div>	<div><div></div></div>		<div><div>M</div></div>	
26	Testing	5/4/2008	5/31/2008	20		<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>			<div><div></div></div>	<div><div></div></div>		<div><div></div></div>	<div><div>G,K</div></div>