SMARTECH

NetCheck Project
Initial Design Report

Neslihan Bulut

Kezban Demirtaş

Hande Çelikkanat

Gülşah Karaduman

Filiz Alaca

Department of Computer Engineering

METU

December 2005

TABLE OF CONTENTS

1 INTRODUCTION..... 3
1.1 Purpose of This Document..... 3
1.2 Scope and Definition of the Project..... 3
1.3 Project Overview..... 4

2 SYSTEM MODULES..... 5
2.1 Web Module..... 5
2.2 System Management Module..... 5
2.3 Network Traffic Monitoring Module..... 6
2.4 Content Filtering Module..... 6
2.5 Restriction Module..... 6
2.6 Statistics Module..... 6
2.7 Logging Module..... 7
2.8 Caching Module..... 7

3 SYSTEM DESIGN..... 8

3.1 Use Case Diagrams..... 8
3.1.1 Use Case for Administrator..... 8
3.1.2 Use Case for Local Client and Server..... 10
3.1.3 Use Case for Remote Client and Server..... 11
3.2 Class Diagrams..... 12
3.3 Sequence Diagrams..... 19
3.3.1 Web Module..... 19
3.3.1.1 Authentication on the Web..... 19
3.3.2 System Management Module..... 20
3.3.2.1 Specify Running Mode of the System..... 20
3.3.2.2 Update Users of the System..... 21
3.3.2.3 Update User Groups of the System..... 22
3.3.2.4 Update Administrators of the System..... 23
3.3.2.5 Update Black URL Lists of the System..... 24
3.3.2.6 Update Black URL Groups of the System..... 25
3.3.2.7 Update White URL List of the System..... 25
3.3.2.8 Update Black Word List of the System..... 26
3.3.2.9 Update Black Word Groups of the System..... 27
3.3.2.10 Update White Word List of the System..... 27
3.3.3 Network Traffic Monitoring Module..... 28
3.3.3.1 Saving the Network Traffic Logs..... 28
3.3.3.2 Monitoring Network Traffic..... 29
3.3.4 Content Filtering Module..... 30
3.3.4.1 Applying Content Filtering..... 30
3.3.4.2 Applying Confidential Data Filtering..... 31
3.3.5 Restriction Module..... 32
3.3.5.1 Applying Download Restriction..... 32
3.3.5.2 Applying URL Access Restriction..... 33
3.3.6 Statistics Module..... 34
3.3.6.1 Computing Daily Network Statistics..... 34
3.3.6.2 Computing Web Site Hit Rates and Confidential Data Violations..35

3.3.7 Logging Module ***36***
3.3.7.1 Saving the Configuration Logs *36*
3.3.8 Caching Module ***37***
3.3.8.1 Applying the Caching Mechanism *37*
3.4 Activity Diagrams **38**
3.4.1 Web Module ***38***
3.4.2 Restriction, Content Filtering, Caching and Logging Module ***42***

4 DATABASE DESIGN **45**
4.1 Database Table Specifications **45**
4.2 Database Table SQL's **51**

5 SYNTAX SPECIFICATION **56**
5.1 Naming Conventions **56**
5.2 Commenting Conventions **57**

6 HARDWARE AND SOFTWARE SPECIFICATION **58**
6.1 Software Specifications **58**
6.2 Hardware Specifications **58**
6.3 Tool Specifications **59**

7 UPDATED GANNT CHART **60**

1 INTRODUCTION

1.1 Purpose of this Document

The purpose of this document is to initiate the design specifications of the project. In this report, we intend to give detailed information about how our solutions fulfill the problem requirements. During our studies on this report, we have developed our sight to the problem and to the solution. We will present our project's modular specification and UML diagrams (use-case, class, sequence and activity diagrams) through which our understanding of the system improves.

1.2 Scope and Definition of the Project

In today's world, Internet has become the key tool in every aspect of life. With the increasing internet usage and the vulnerability of Internet to abuse, security gains more and more importance. Organizations are one of the areas where Internet is heavily used. In order not to lose confidential information about the company, about the projects they are working on, and mainly to ensure security policies, organizations have to take precautions for abuse. At this point, security tools appear to act as the protector against malicious usage.

NetCheck will be a web-based application level gateway which offers secure Internet access for the organizations. Our intended software 'NetCheck' will mainly provide the following facilities:

- Real-time network monitoring
- Content filtering
- Download restriction
- Access restriction
- Statistical data about network traffic
- Caching
- Web interface for the control of the program
- Confidential data hiding

1.3 Project Overview

Administrative Facilities: The administrator of the system can define access and download rights for individuals in the local network via the web interface. He/she can define black words and black word groups, black URL and black URL groups, and also he/she can define user groups and assign black URL groups and black word groups to user groups.

Network Traffic Monitoring: All incoming and outgoing web traffic will be displayed on a web page in real time. Source address, destination address, accessed URL, size of communication packets, and time of communication should be monitored.

Access Restriction: Restriction will be applied to individuals according to black URL groups which are assigned to the user. Also, in some time intervals, some specified URL's can be restricted to the user (e.g. URL x cannot be accessed between 09:00-11:00).

Download Restriction: The administrator will be able to specify a bandwidth limit for the users' download operation.

Content Filtering: If a user requests a site which contains words that are in black word list, than content of the incoming packet should be filtered according to the black word list. The content in the black word list will be defined by the administrator. For example the list may keep content about sex, alcohol/drug, violence, and gambling.

2 SYSTEM MODULES

2.1 Web Module

The web module will provide the administrator with an interface to manage the system. Firstly, the administrator will be asked for his/her username and password for the authentication and after the username/password verification, the administrator will have the right to control or monitor the following system features via that interface;

- Monitoring network traffic,
- Defining access and download rights for local IP's,
- Restricting URL access,
- Specifying black & white word lists,
- Specifying the running mode of the program,
- Defining security policies for protecting confidential data,
- Monitoring network statistics.

2.2 System Management Module

System management module will adjust the necessary settings of the system according to the running mode specified by the administrator. There will be three different running modes supported by our system which are defined below;

- ***Free Mode:*** The free mode will only provide monitoring network traffic. There will be no filtering or restriction mechanism.
- ***Normal Mode:*** This mode will be the default mode of the system. In addition to the network traffic monitoring, this mode will support URL access and download size restriction, content filtering and logging. Also, network statistics can be monitored when the system runs in normal mode.
- ***Secure Mode:*** In addition to normal mode facilities, this mode will include a mechanism for providing security for confidential data such as a formula invented by a company working for pharmaceutics.

2.3 Network Traffic Monitoring Module

This module is about monitoring all incoming and outgoing web traffic in real time. The following information of incoming and outgoing packets will first be written to database tables and afterwards that information will be displayed to the administrator via web interface;

- Source address,
- Destination address,
- Accessed URL,
- Size of communication packets,
- Time of communication.

2.4 Content Filtering Module

The content filtering module will have a mechanism to parse the data part of a packet for detecting whether the packet includes any black word. If such words exist in a packet, these words will be filtered according to a special algorithm and the filtered packet will be transferred afterwards. This mechanism aims to prevent malicious content coming from remote servers in normal and secure mode and confidential data going outside the local area network for the secure mode.

2.5 Restriction Module

Restriction module of our system will provide two types of restrictions. First type is about URL access restriction and the second one is about download restriction of local clients. URL black list or restricted time interval of URL's will be considered for URL access restriction and local IP's bandwidth limit will be the criteria for download restriction.

2.6 Statistics Module

Statistics module will produce statistics related to the system by applying an appropriate algorithm on the data kept in the database tables. These statistics will be displayed to the administrator via the web interface. This module will mainly produce the following statistics;

- Statistics about persistent requests from local server,
- Statistics about daily network traffic density,
- Statistics about local IP's download size (this may be for a specific IP or a specific user group),
- Statistics about the hit rates of web sites,
- Statistics about local IP's violation for confidential data protection,
- Statistics about local IP's URL requests.

2.7 Logging Module

Logging module will create a log file listing recent actions of the administrator in a human readable format and that file will be composed of lines in the following format:

user name of the administrator | *performed action* | *configured table's name*| *old configuration* | *new configuration*

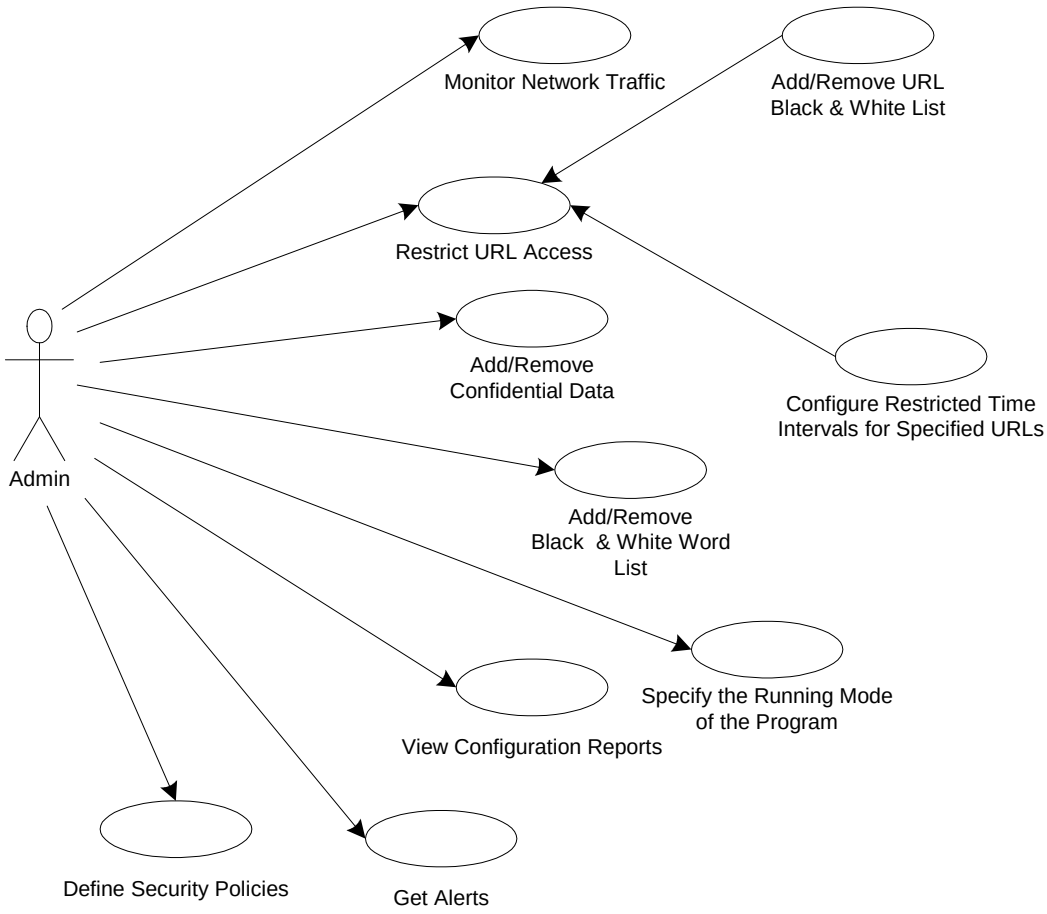
2.8 Caching Module

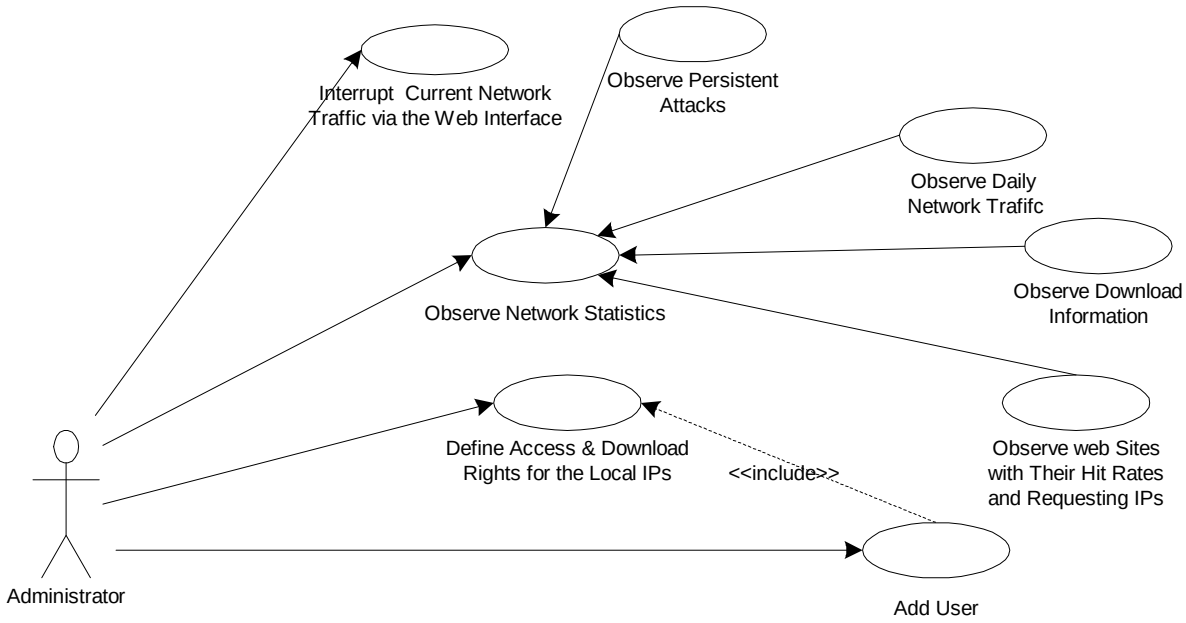
Caching module will apply an algorithm to specify the most frequently accessed URL's and will save the contents of these URL's in a proxy server for giving fast access to the local users and also for minimizing the network traffic density. The algorithm to be used will make use of the network traffic logs which are kept in the database.

3 SYSTEM DESIGN

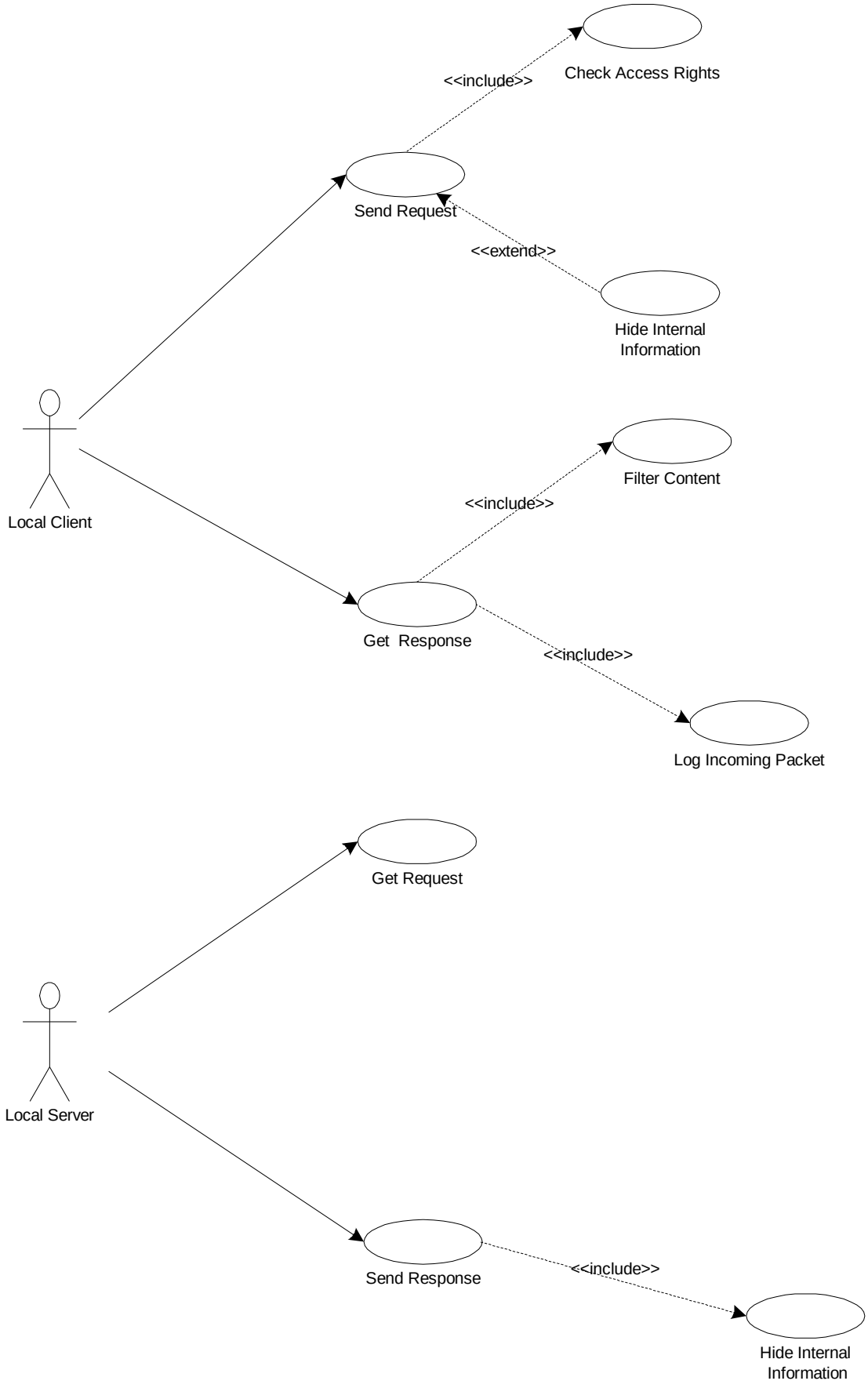
3.1 Use Case Diagrams

3.1.1 Use Case for Administrator

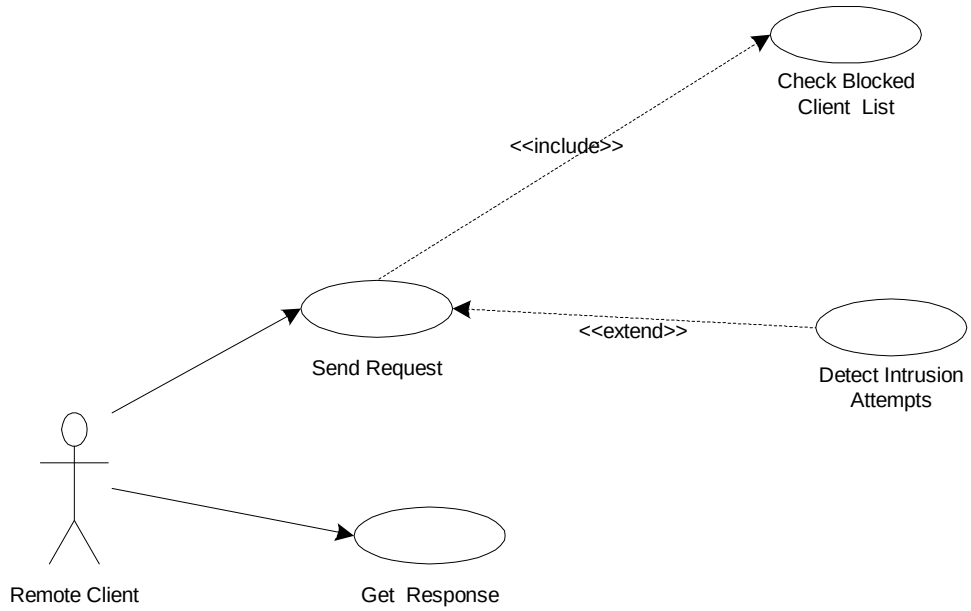


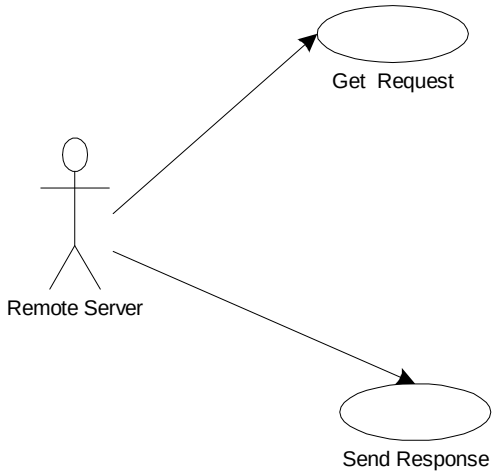

Interrupt Current Network Traffic via the Web Interface
Observe Persistent Attacks
Observe Daily Network Trafifc
Observe Download Information
Observe Network Statistics
Observe web Sites with Their Hit Rates and Requesting IPs
Define Access & Download Rights for the Local IPs
<<include>>
Administrator
Add User

3.1.2 Use Case for Local Client and Server

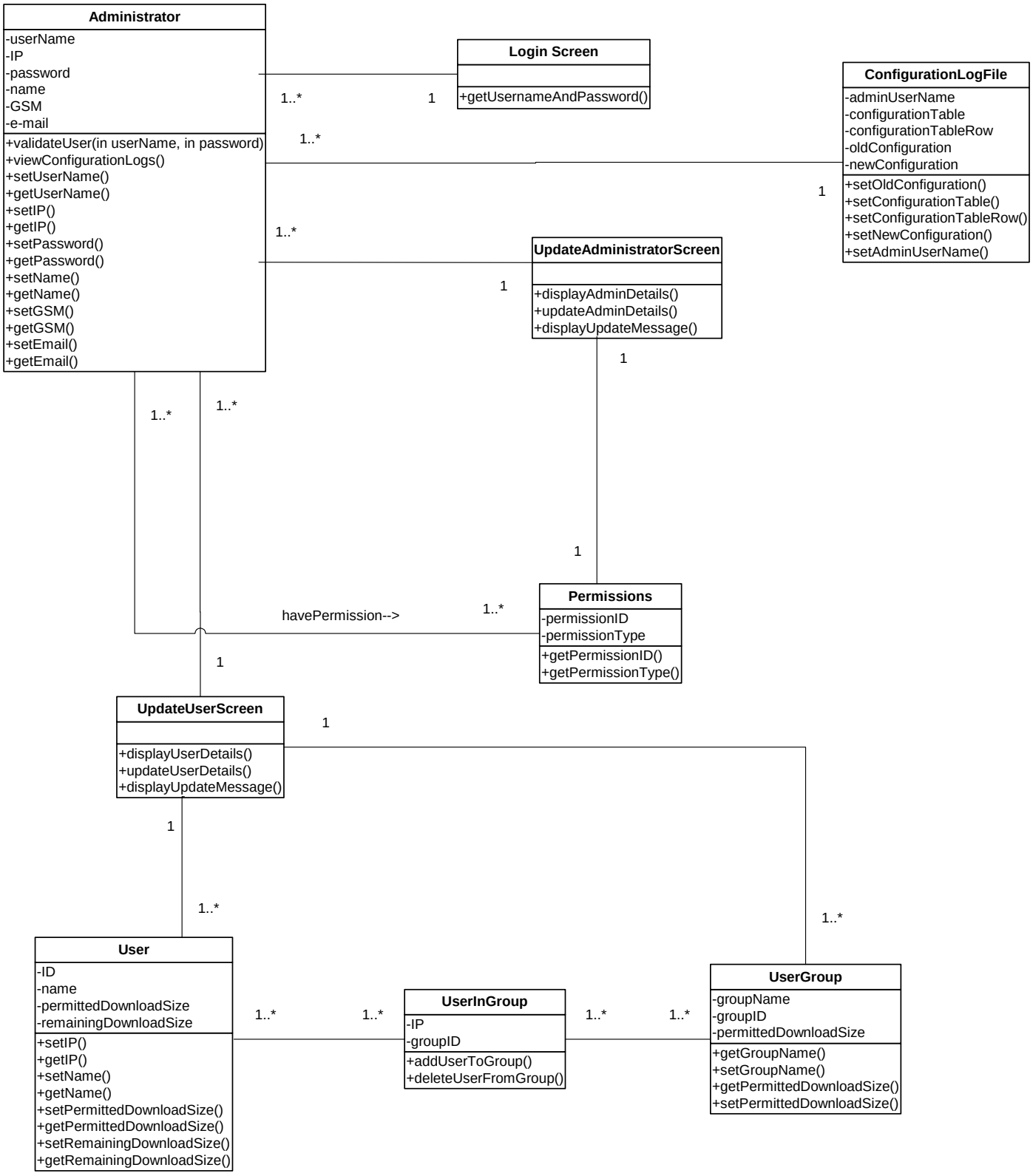


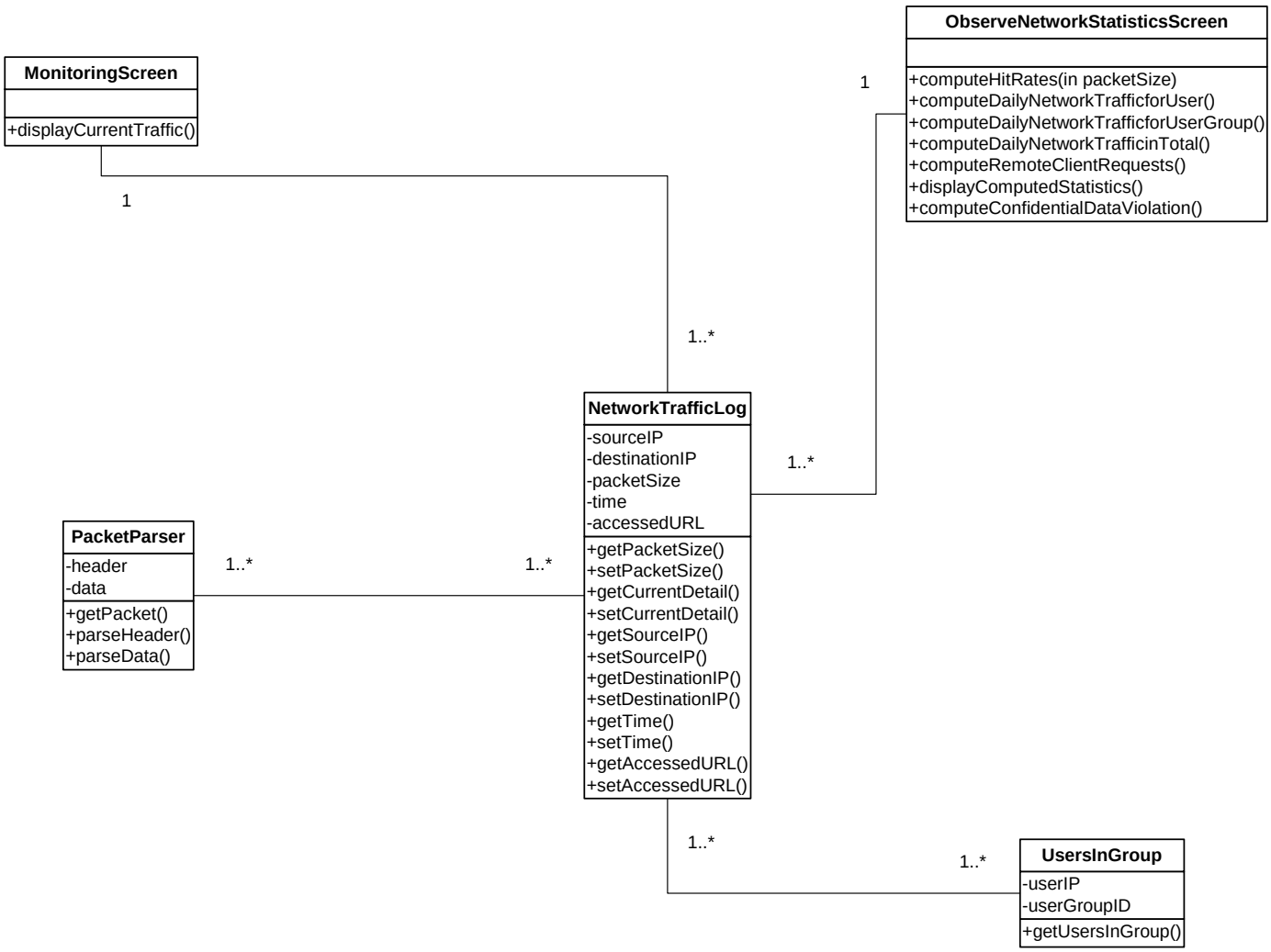
3.1.3 Use Case for Remote Client and Server

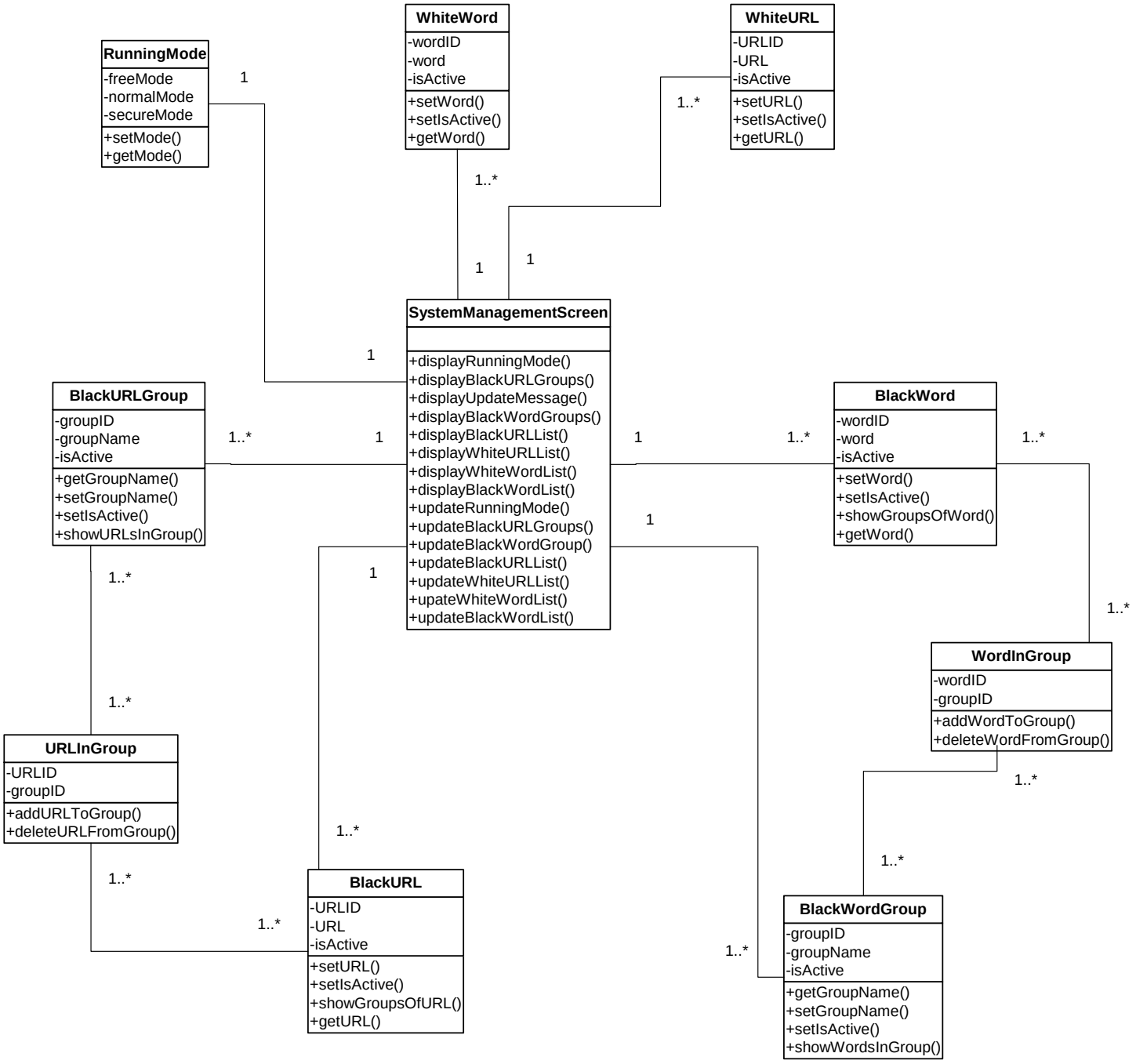


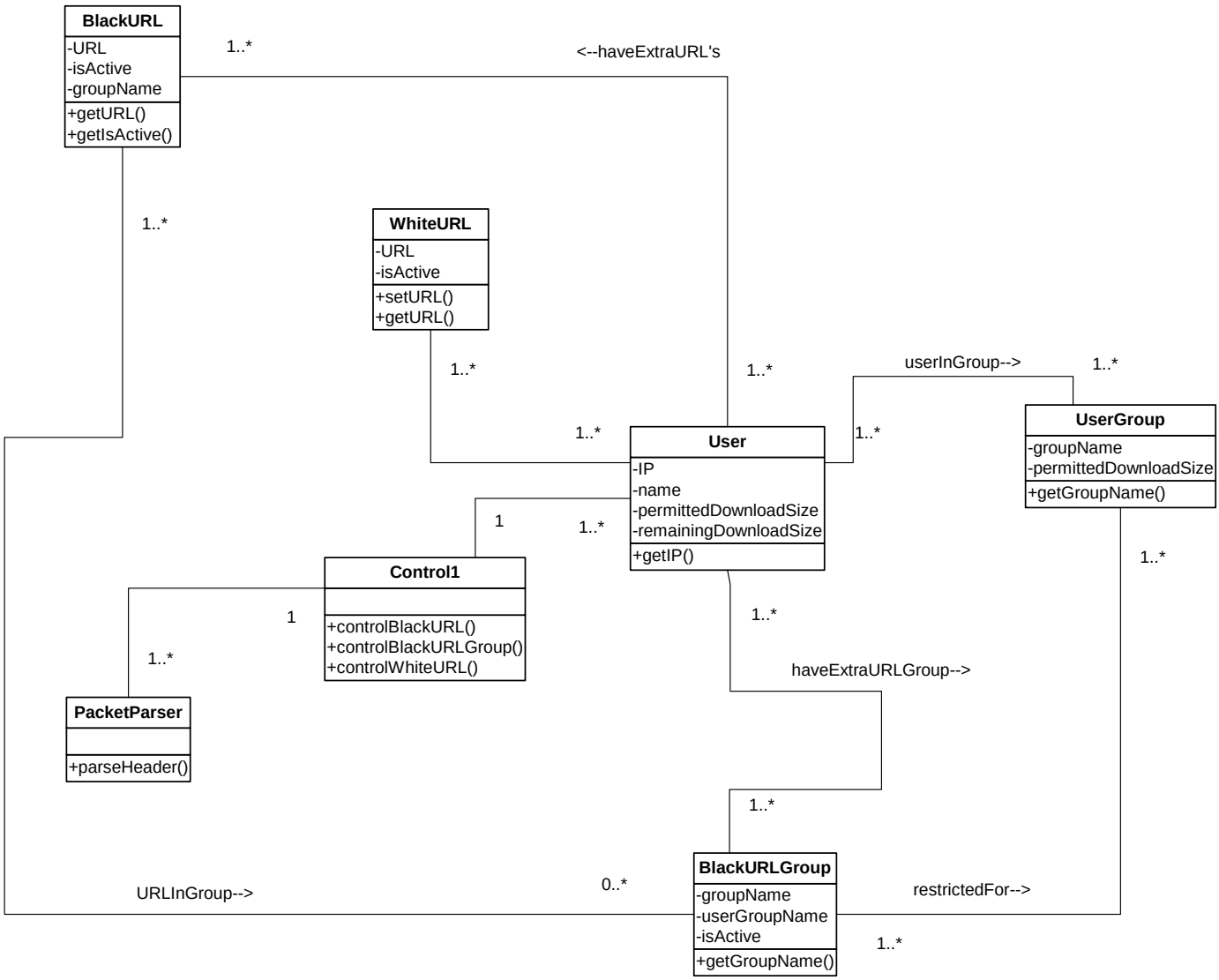


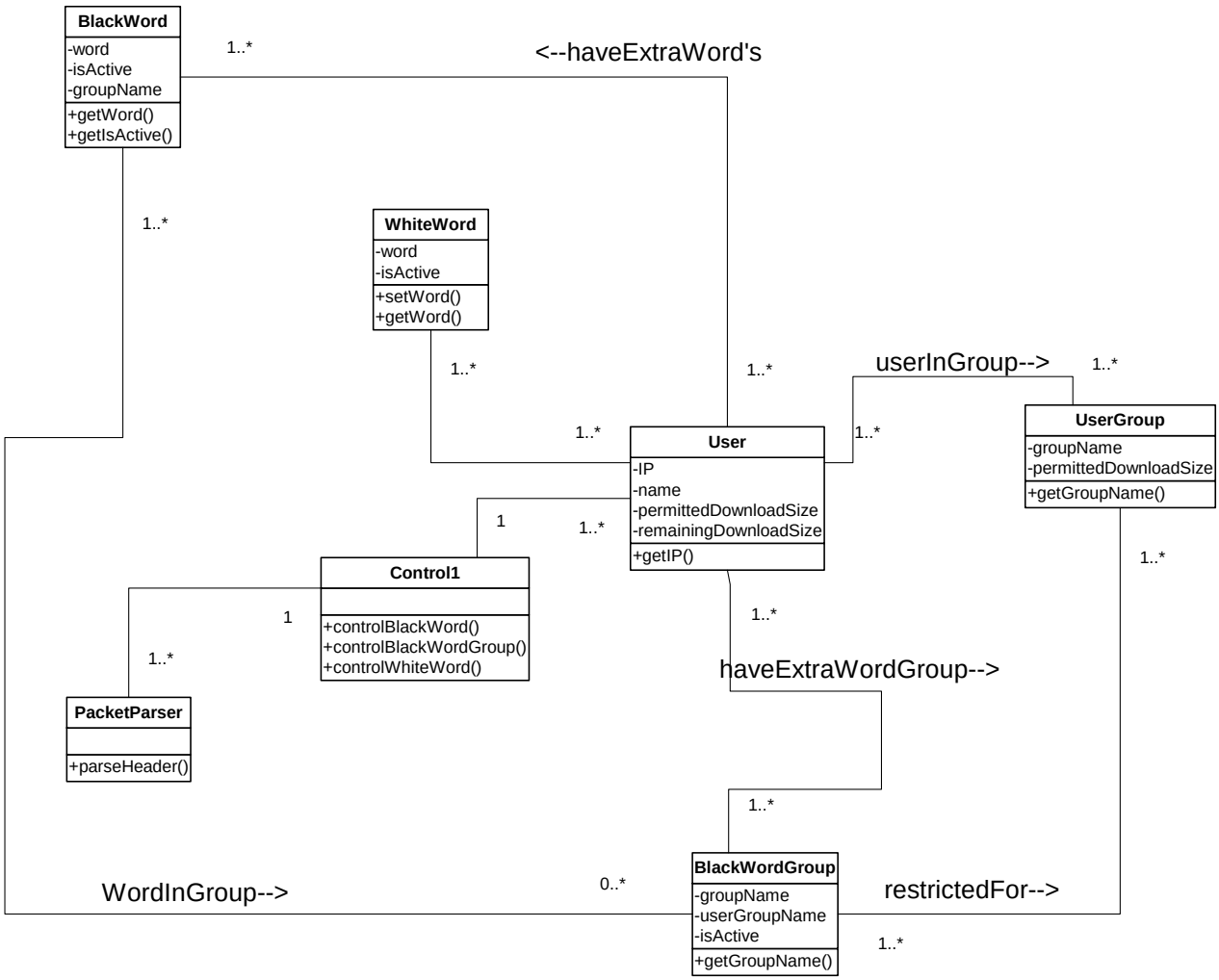
3.2 Class Diagrams

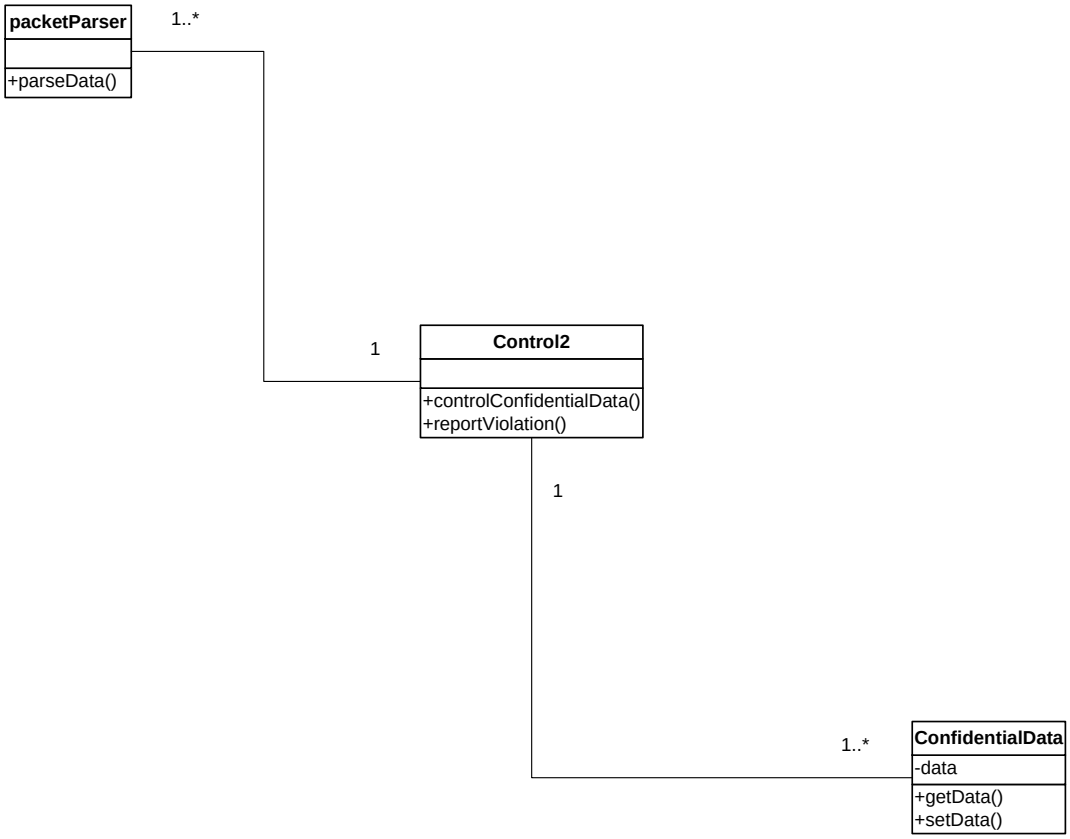


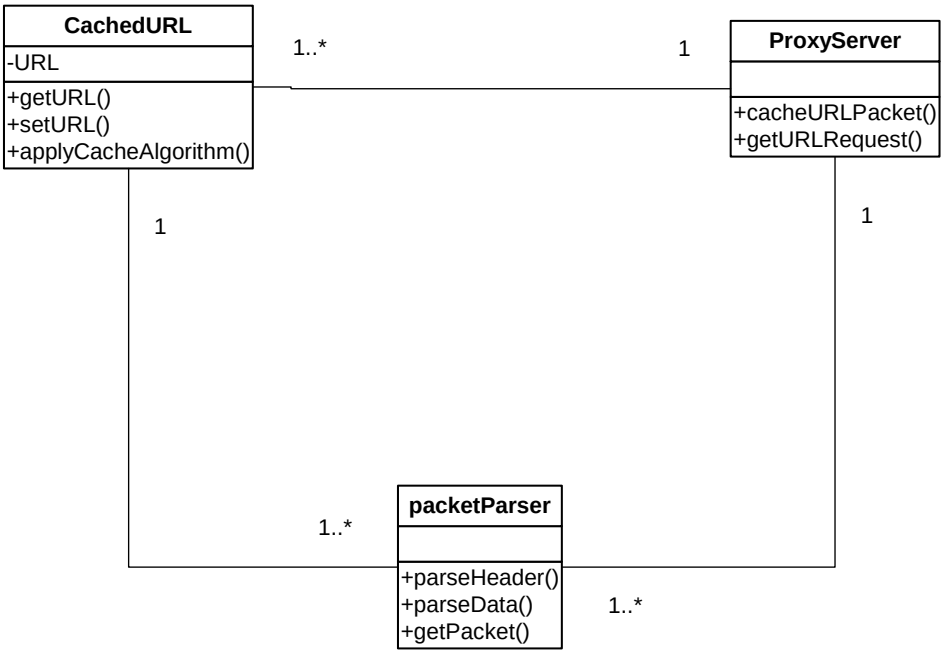

ObserveNetworkStatisticsScreen
+computeHitRates(in packetSize)
+computeDailyNetworkTrafficforUser()
+computeDailyNetworkTrafficforUserGroup()
+computeDailyNetworkTrafficinTotal()
+computeRemoteClientRequests()
+displayComputedStatistics()
+computeConfidentialDataViolation()
MonitoringScreen
+displayCurrentTraffic()
1
1
1..*
NetworkTrafficLog
-sourceIP
-destinationIP
-packetSize
-time
-accessedURL
+getPacketSize()
+setPacketSize()
+getCurrentDetail()
+setCurrentDetail()
+getSourceIP()
+setSourceIP()
+getDestinationIP()
+setDestinationIP()
+getTime()
+setTime()
+getAccessedURL()
+setAccessedURL()
1..*
PacketParser
-header
-data
+getPacket()
+parseHeader()
+parseData()
1..*
1..*
1..*
1..*
UsersInGroup
-userIP
-userGroupID
+getUsersInGroup()


RunningMode
-freeMode
-normalMode
-secureMode
+setMode()
+getMode()
WhiteWord
-wordID
-word
-isActive
+setWord()
+setIsActive()
+getWord()
WhiteURL
-URLID
-URL
-isActive
+setURL()
+setIsActive()
+getURL()
SystemManagementScreen
+displayRunningMode()
+displayBlackURLGroups()
+displayUpdateMessage()
+displayBlackWordGroups()
+displayBlackURLList()
+displayWhiteURLList()
+displayWhiteWordList()
+displayBlackWordList()
+updateRunningMode()
+updateBlackURLGroups()
+updateBlackWordGroup()
+updateBlackURLList()
+updateWhiteURLList()
+upateWhiteWordList()
+updateBlackWordList()
BlackURLGroup
-groupID
-groupName
-isActive
+getGroupName()
+setGroupName()
+setIsActive()
+showURLsInGroup()
BlackWord
-wordID
-word
-isActive
+setWord()
+setIsActive()
+showGroupsOfWord()
+getWord()
WordInGroup
-wordID
-groupID
+addWordToGroup()
+deleteWordFromGroup()
URLInGroup
-URLID
-groupID
+addURLToGroup()
+deleteURLFromGroup()
BlackURL
-URLID
-URL
-isActive
+setURL()
+setIsActive()
+showGroupsOfURL()
+getURL()
BlackWordGroup
-groupID
-groupName
-isActive
+getGroupName()
+setGroupName()
+setIsActive()
+showWordsInGroup()
1
1..*


BlackURL
-URL
-isActive
-groupName
+getURL()
+getIsActive()
1..*
<--haveExtraURL's
1..*
WhiteURL
-URL
-isActive
+setURL()
+getURL()
1..*
1..*
userInGroup-->
1..*
UserGroup
-groupName
-permittedDownloadSize
+getGroupName()
1..*
User
-IP
-name
-permittedDownloadSize
-remainingDownloadSize
+getIP()
1..*
1..*
1
1..*
Control1
+controlBlackURL()
+controlBlackURLGroup()
+controlWhiteURL()
1
1..*
1..*
haveExtraURLGroup-->
PacketParser
+parseHeader()
1..*
BlackURLGroup
-groupName
-userGroupName
-isActive
+getGroupName()
0..*
URLInGroup-->
restrictedFor-->
1..*


BlackWord
-word
-isActive
-groupName
+getWord()
+getIsActive()
1..*
<--haveExtraWord's
1..*
WhiteWord
-word
-isActive
+setWord()
+getWord()
1..*
1..*
userInGroup-->
1..*
UserGroup
-groupName
-permittedDownloadSize
+getGroupName()
1..*
1..*
User
-IP
-name
-permittedDownloadSize
-remainingDownloadSize
+getIP()
1
1..*
1..*
Control1
+controlBlackWord()
+controlBlackWordGroup()
+controlWhiteWord()
1
1..*
haveExtraWordGroup-->
PacketParser
+parseHeader()
1..*
BlackWordGroup
-groupName
-userGroupName
-isActive
+getGroupName()
0..*
WordInGroup-->
restrictedFor-->
1..*


packetParser
+parseData()
1..*
1
Control2
+controlConfidentialData()
+reportViolation()
1
1..*
ConfidentialData
-data
+getData()
+setData()

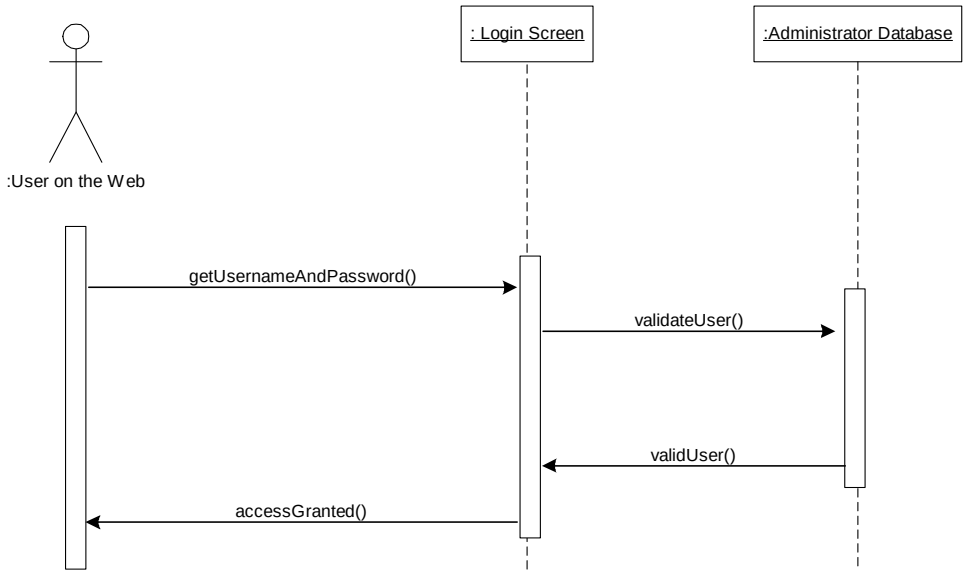

CachedURL
-URL
+getURL()
+setURL()
+applyCacheAlgorithm()
1..*
1
ProxyServer
+cacheURLPacket()
+getURLRequest()
1
1
1..*
packetParser
+parseHeader()
+parseData()
+getPacket()
1..*

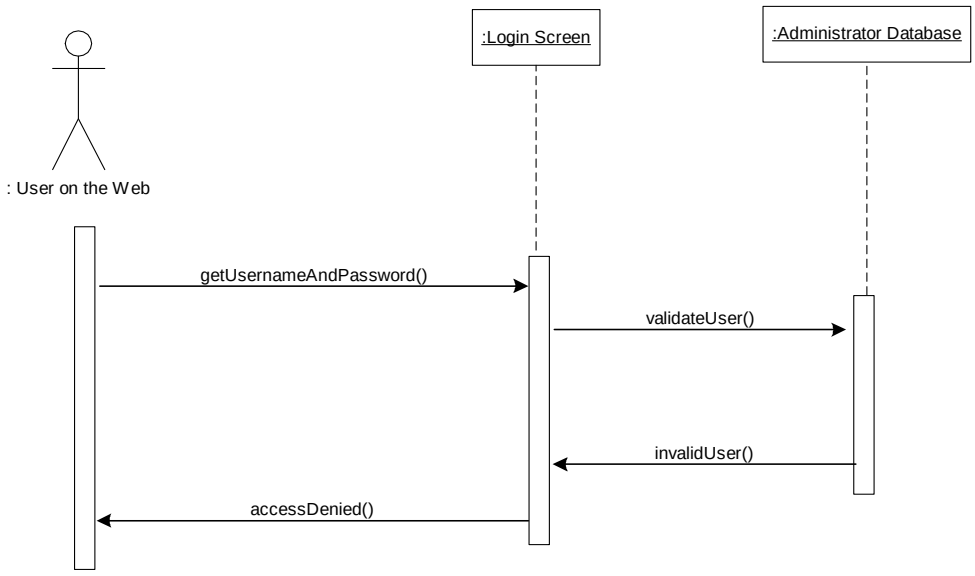
3.3 Sequence Diagrams

3.3.1 Web Module

Under the web module, we have only presented the authentication sequence diagram, but web module is also responsible for the user interface of the other modules.

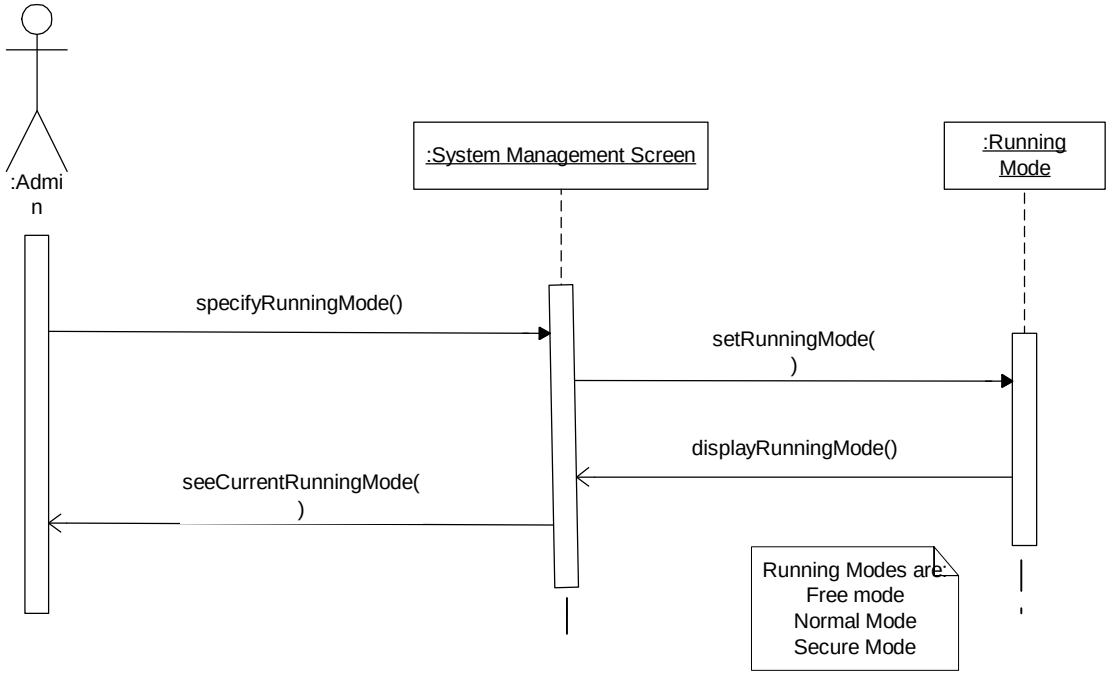
3.3.1.1 Authentication on the Web



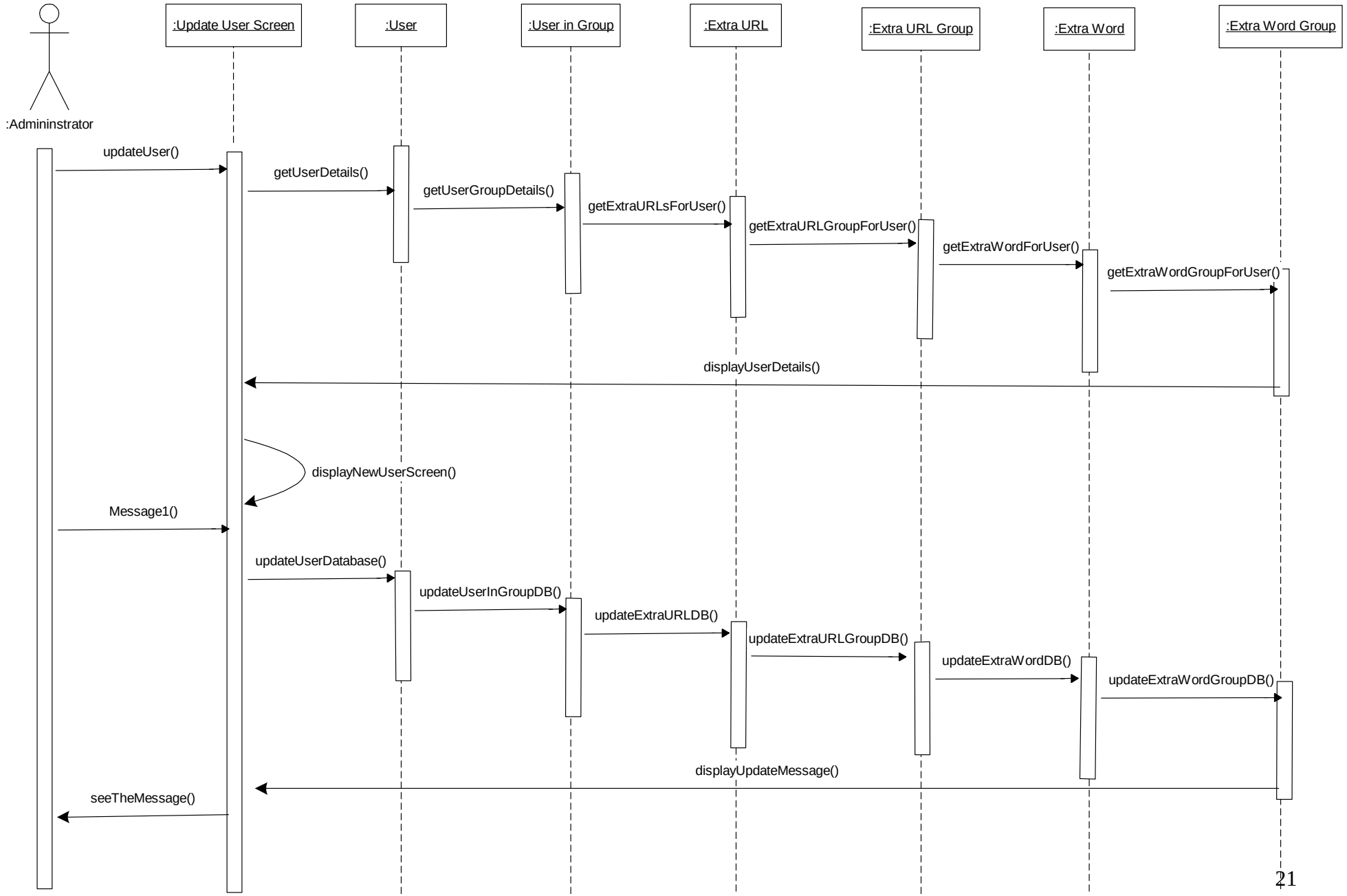


3.3.2 System Management Module

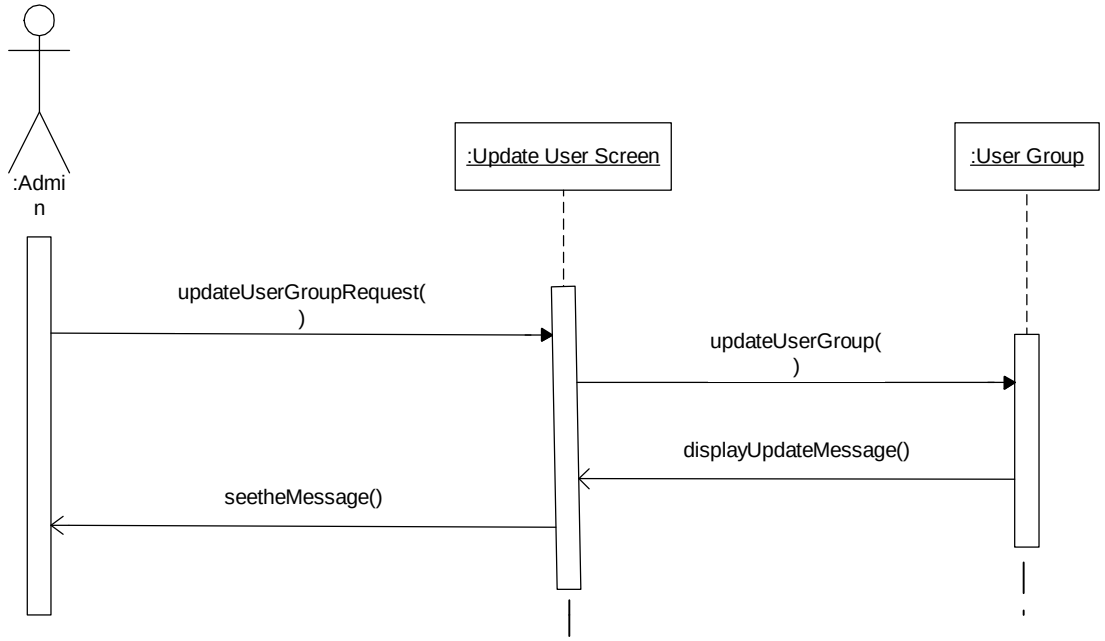
3.3.2.1 Specify the Running Mode of the System



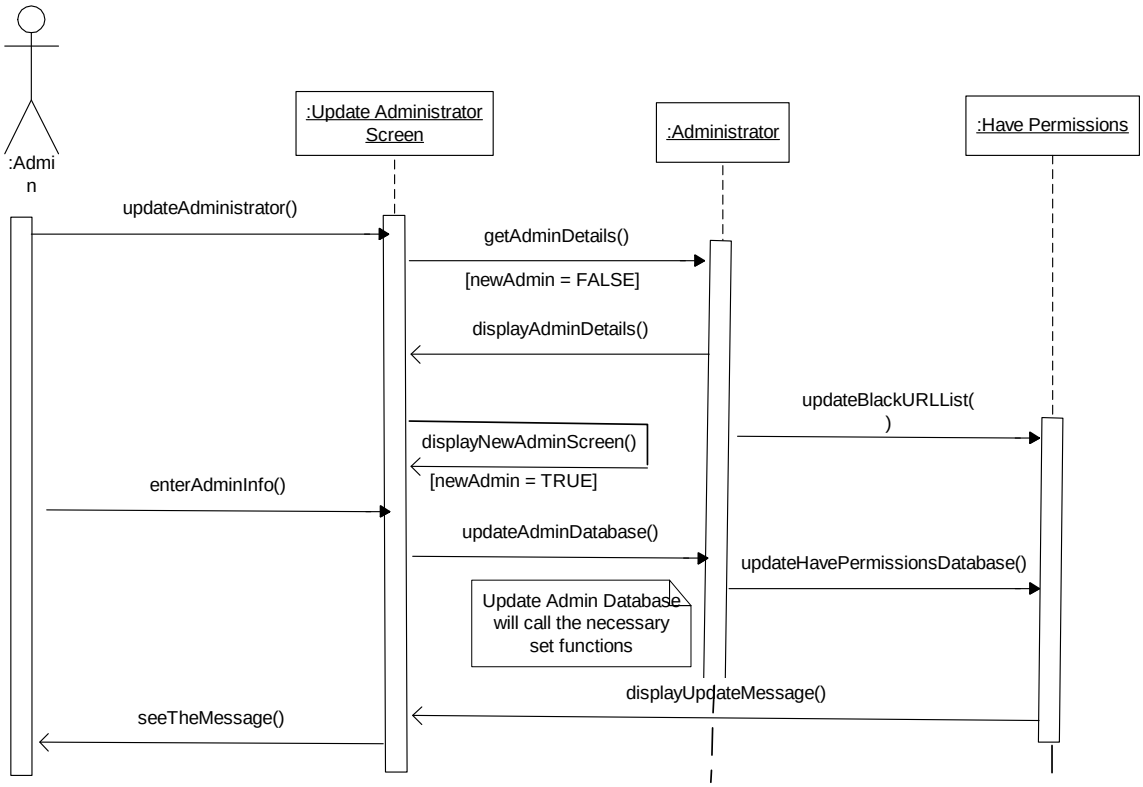
3.3.2.2 Update Users of the System

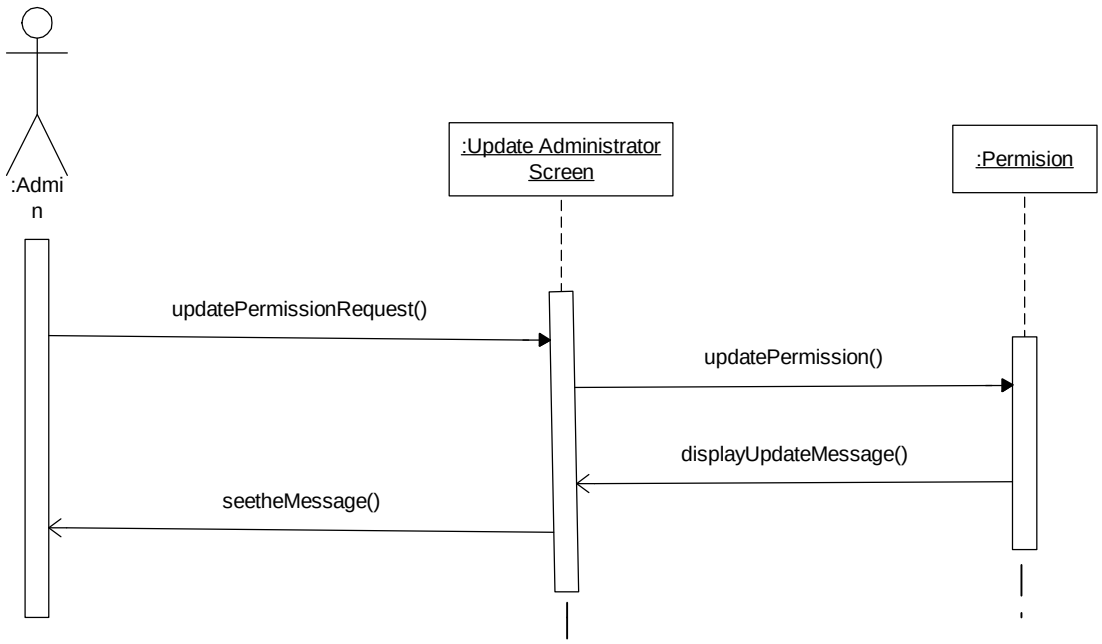


3.3.2.3 Update User Groups of the System

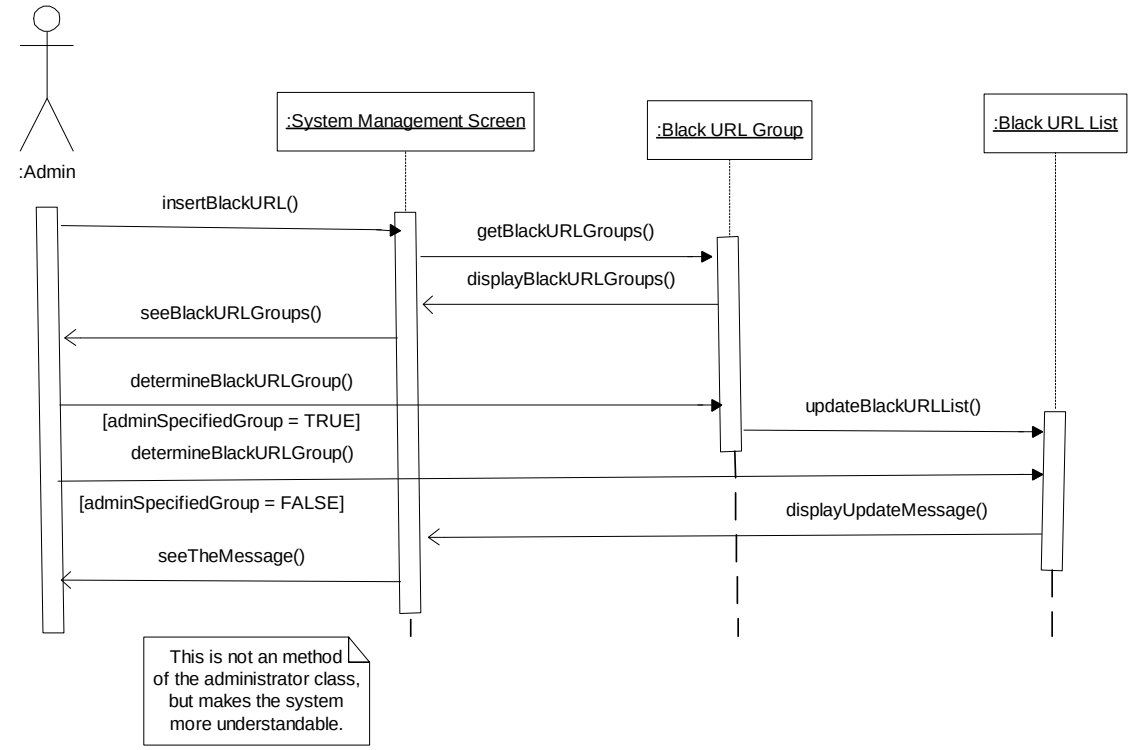


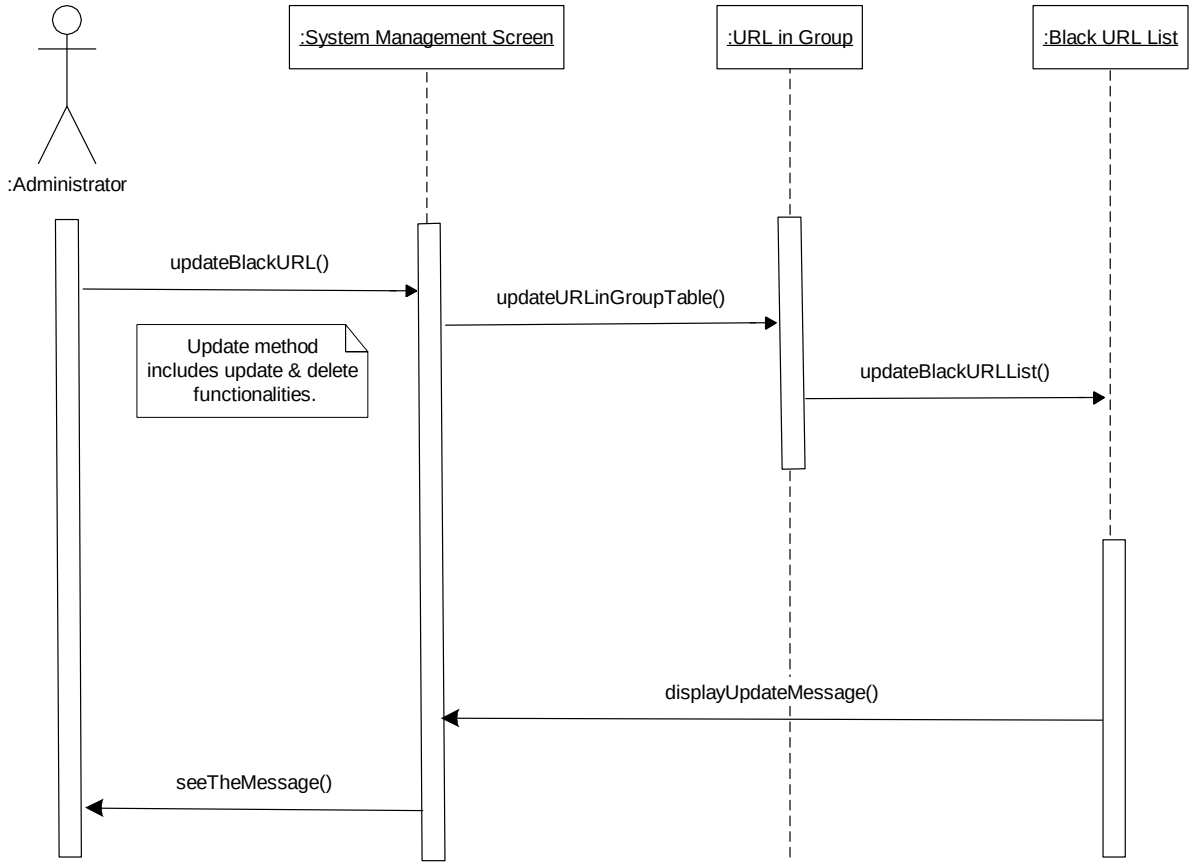
3.3.2.4 Update Administrators of the System



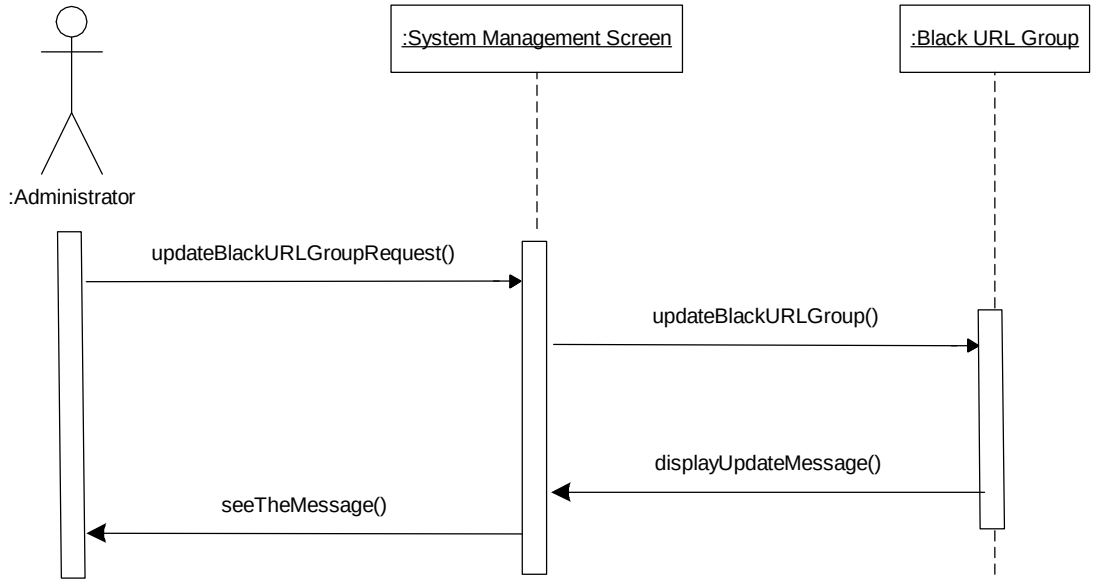


3.3.2.5 Update Black URL List of the System

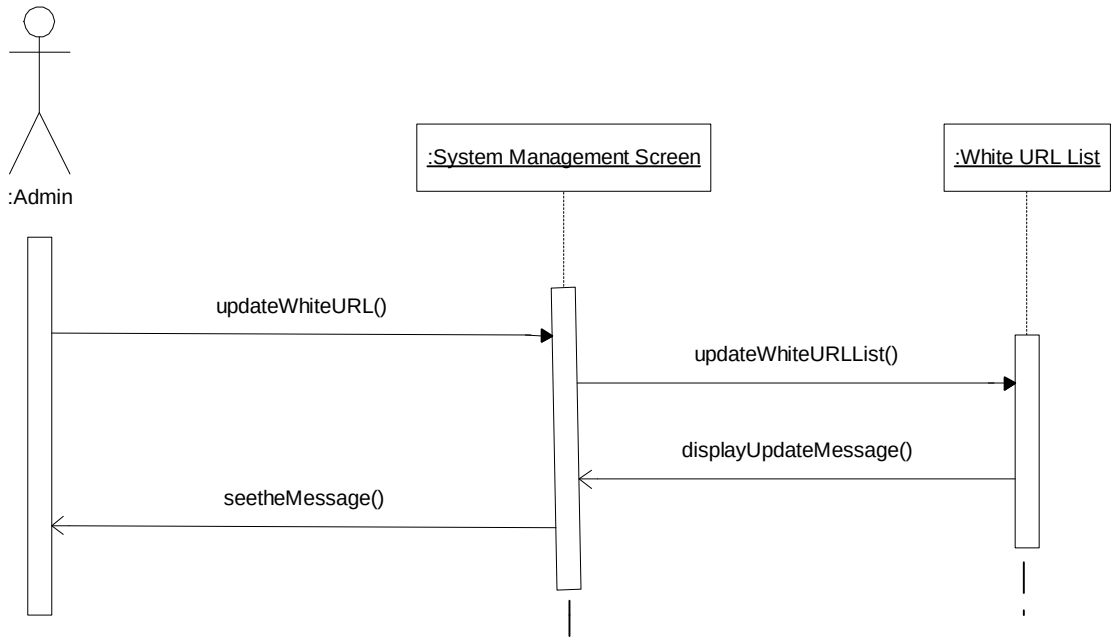




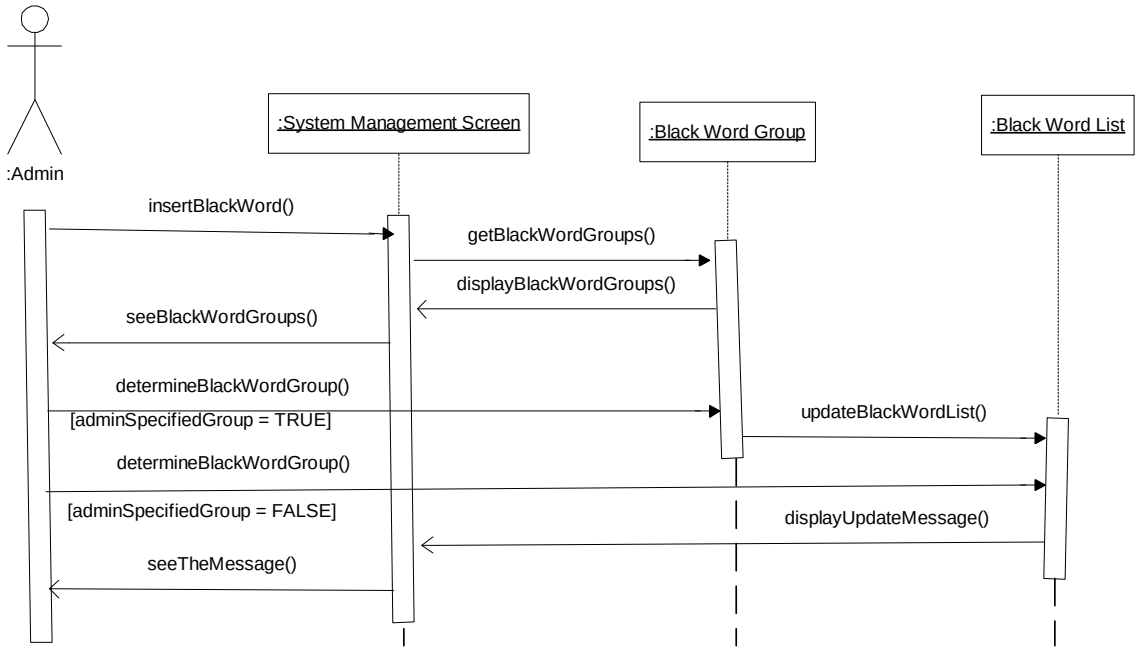
3.3.2.6 Update Black URL Groups of the System

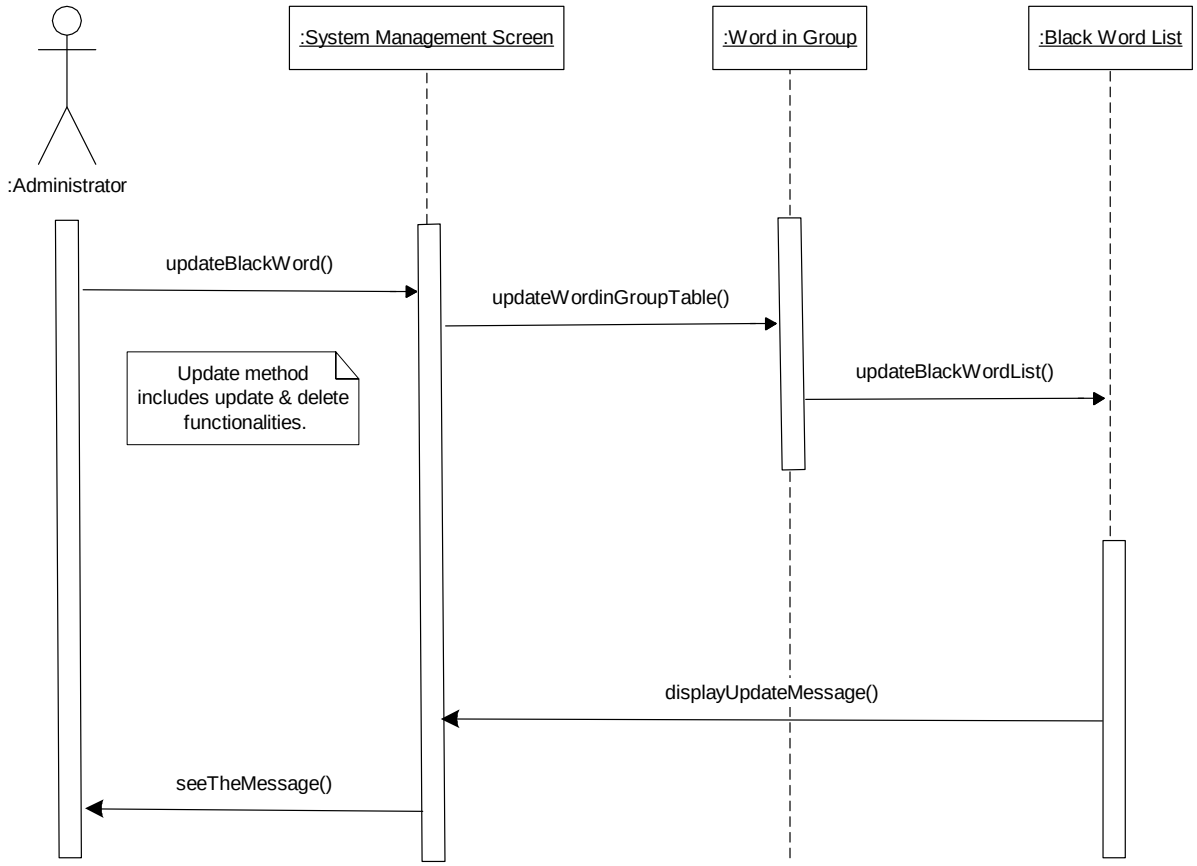


3.3.2.7 Update White URL List of the System

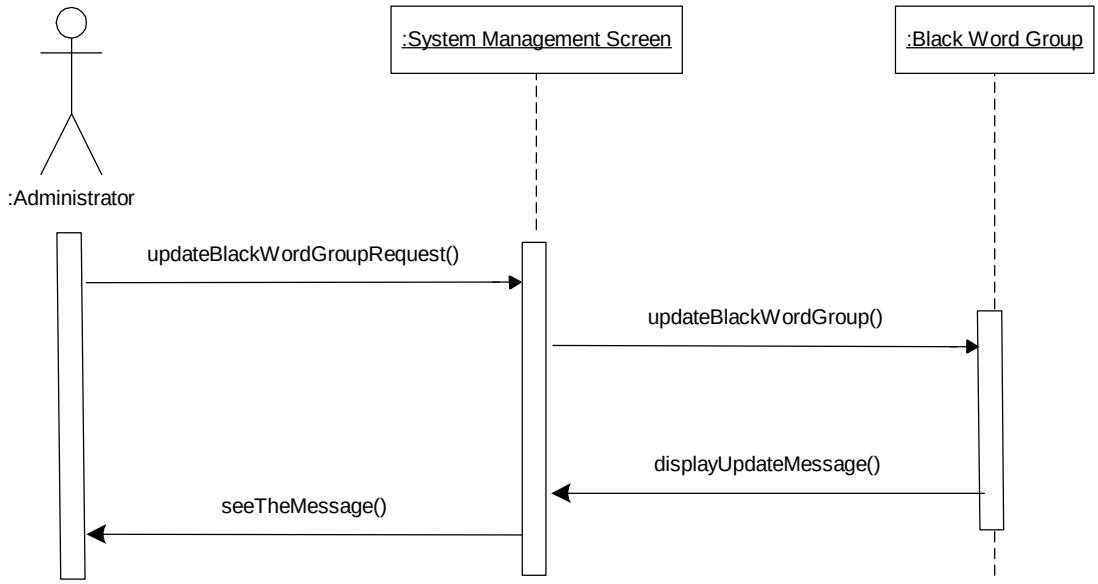


3.3.2.8 Update Black Word List of the System

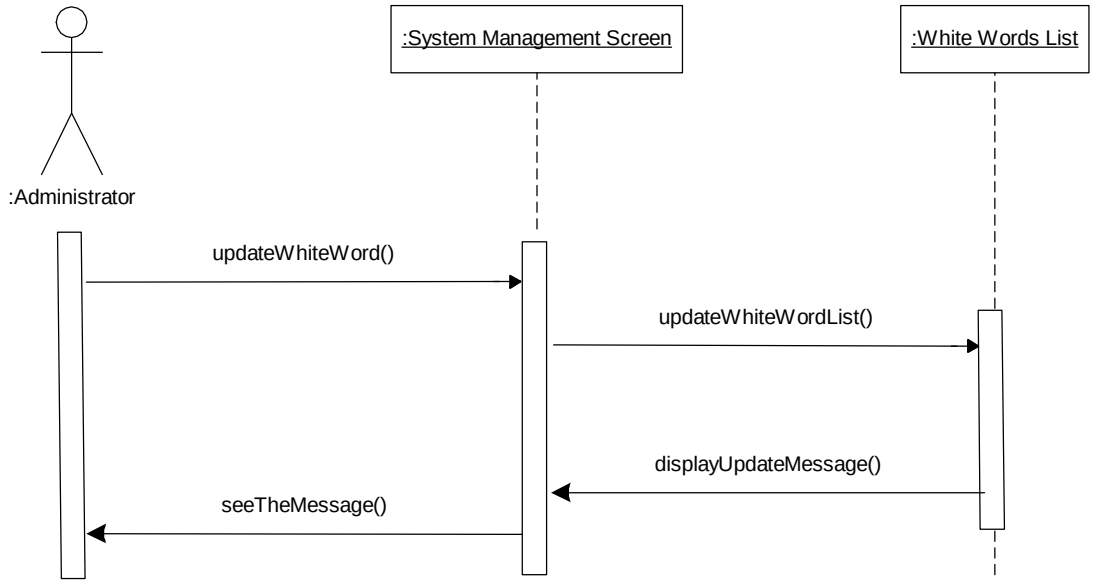




3.3.2.9 Update Black Word Groups of the System

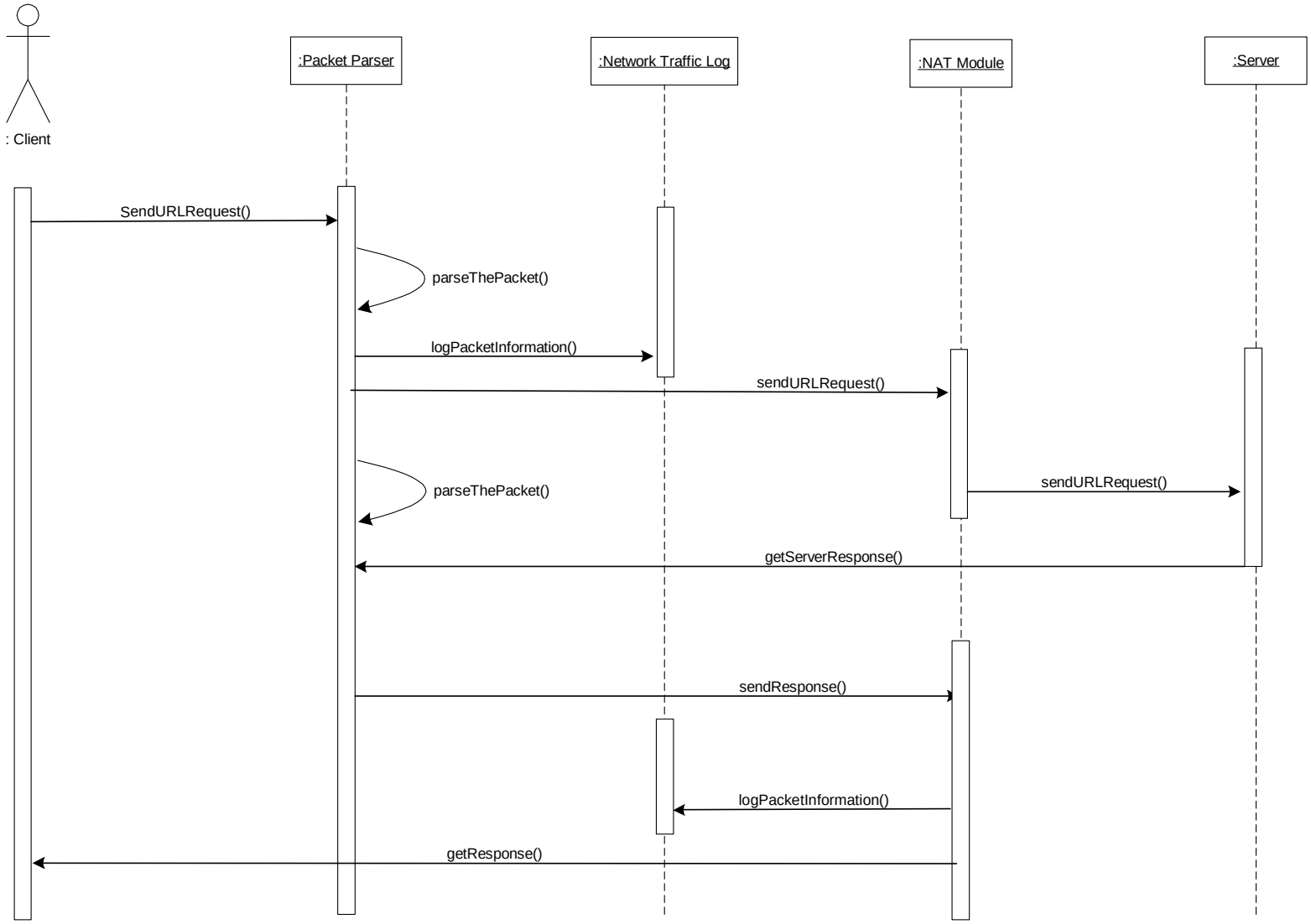


3.3.2.10 Update White Word List of the System

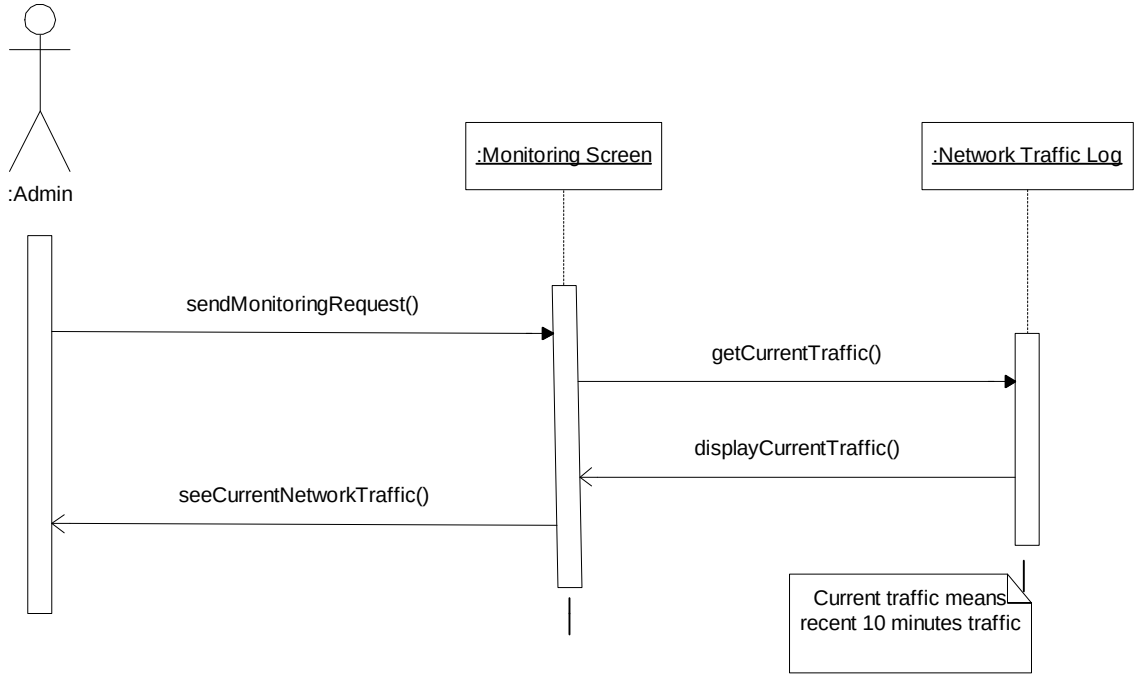


3.3.3 Network Traffic Monitoring Module

3.3.3.1 Saving the Network Traffic Logs

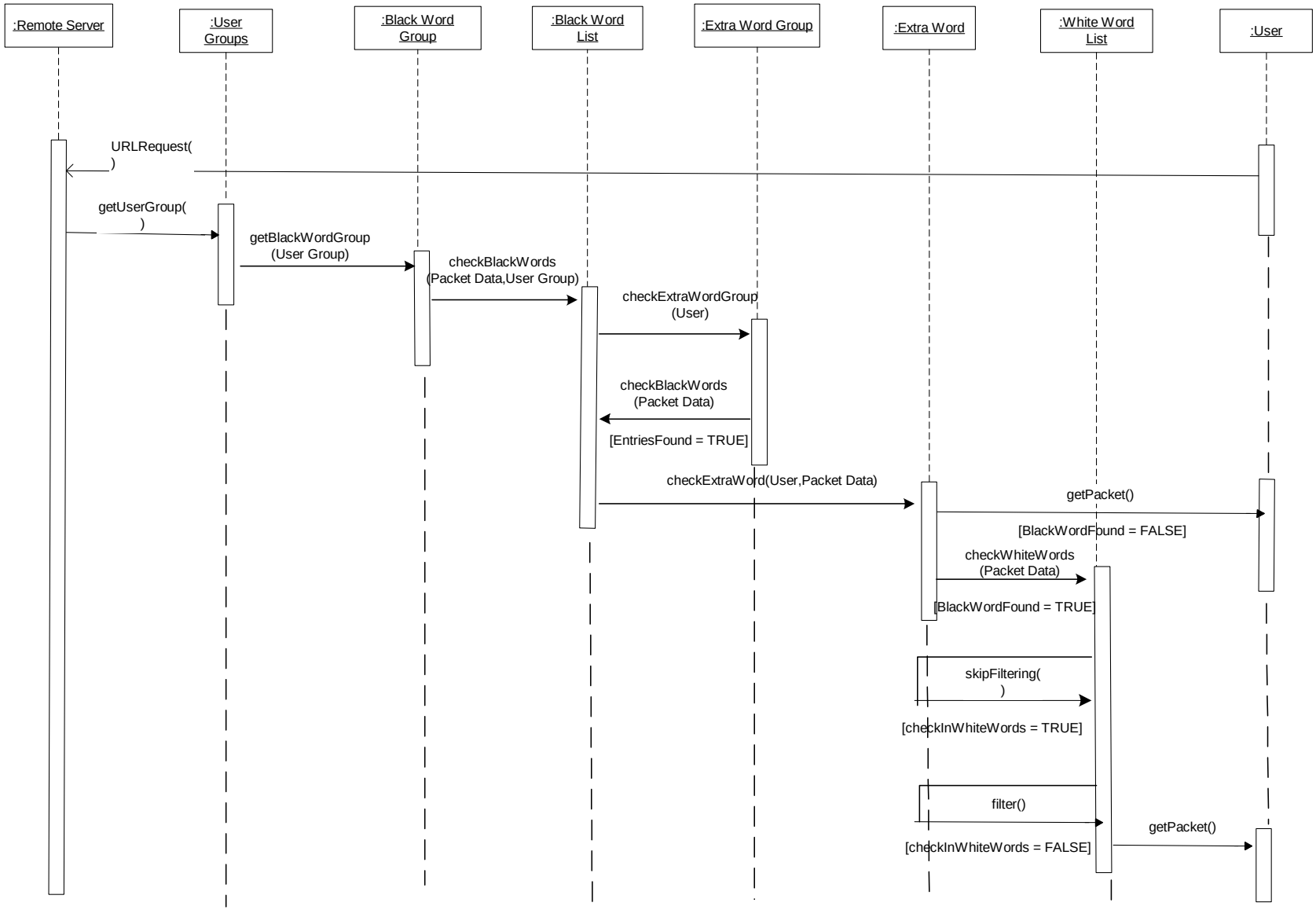


3.3.3.2 Monitoring Network Traffic

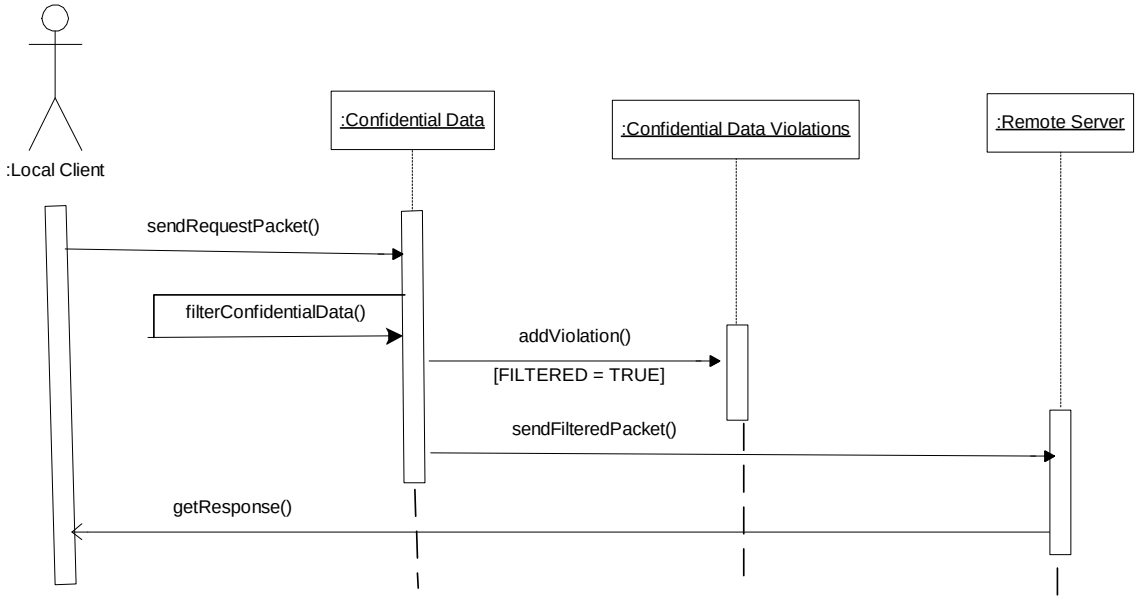


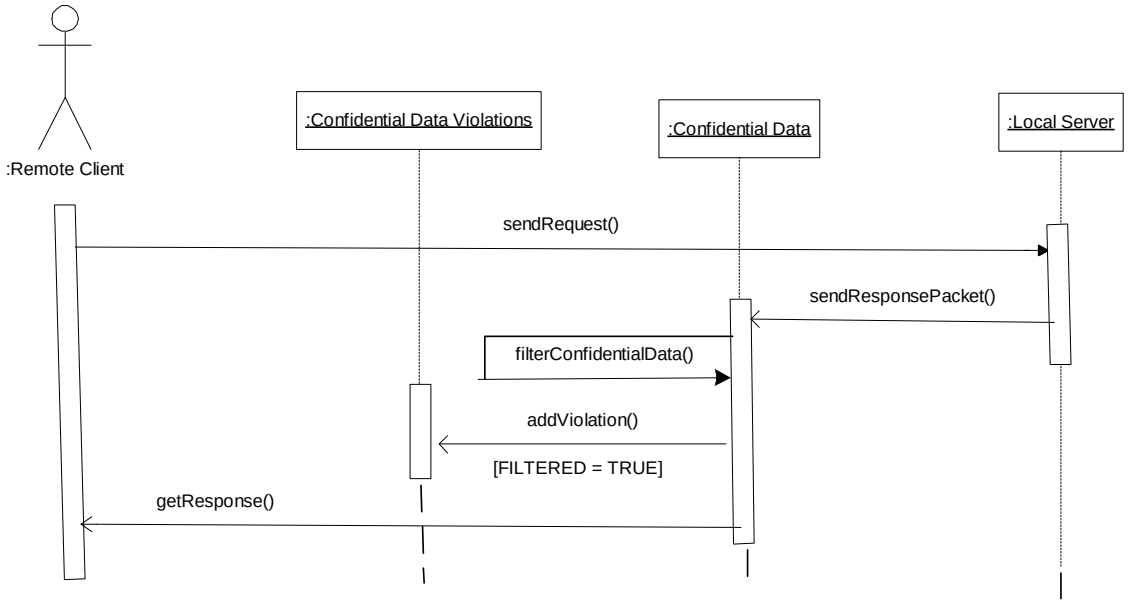
3.3.4 Content Filtering Module

3.3.4.1 Applying Content Filtering



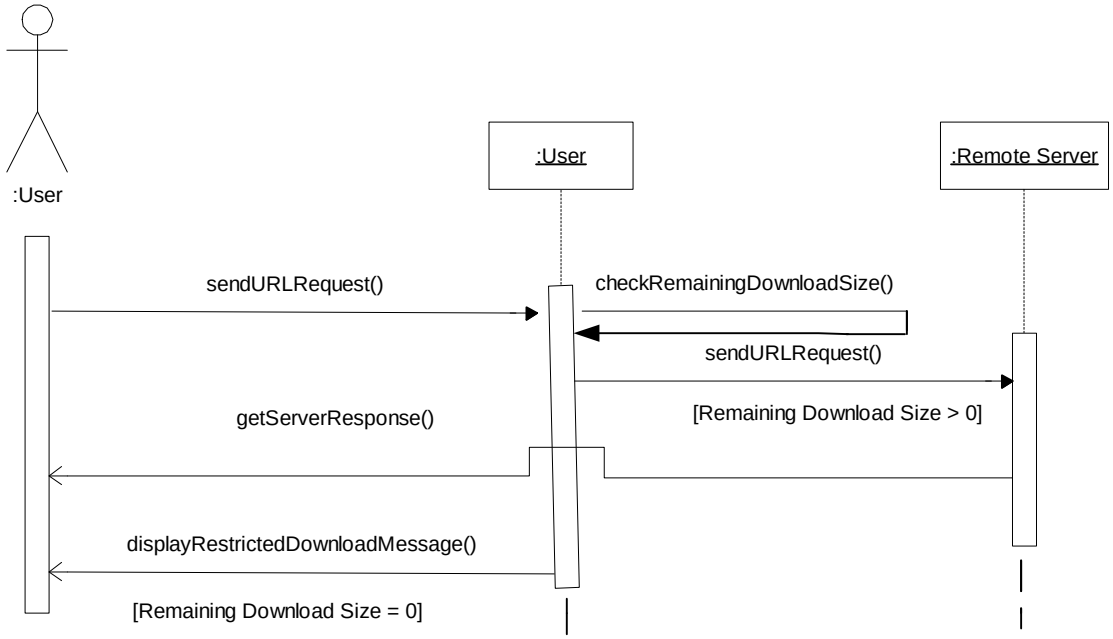
3.3.4.2 Applying Confidential Data Filtering



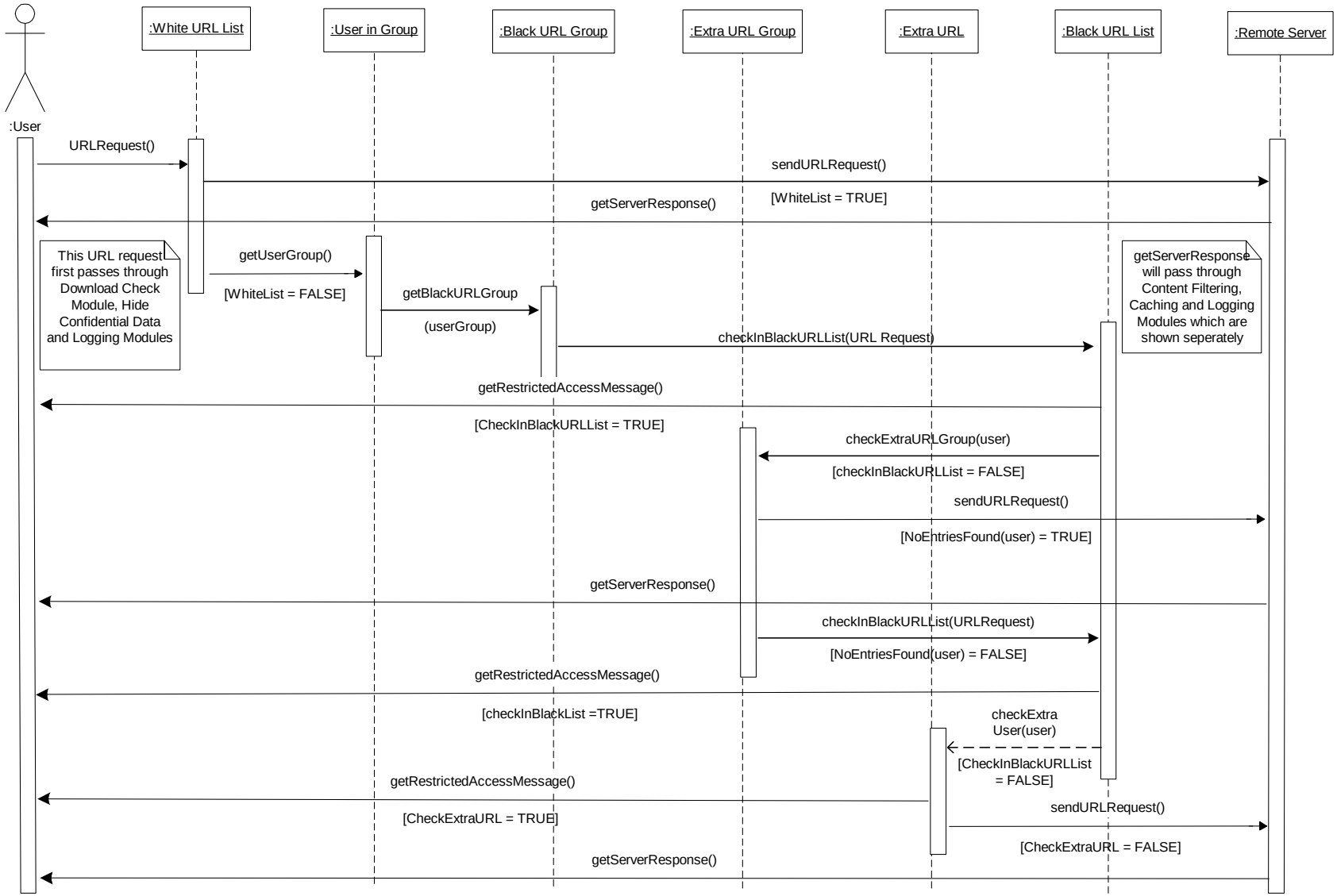


3.3.5 Restriction Module

3.3.5.1 Applying Download Restriction

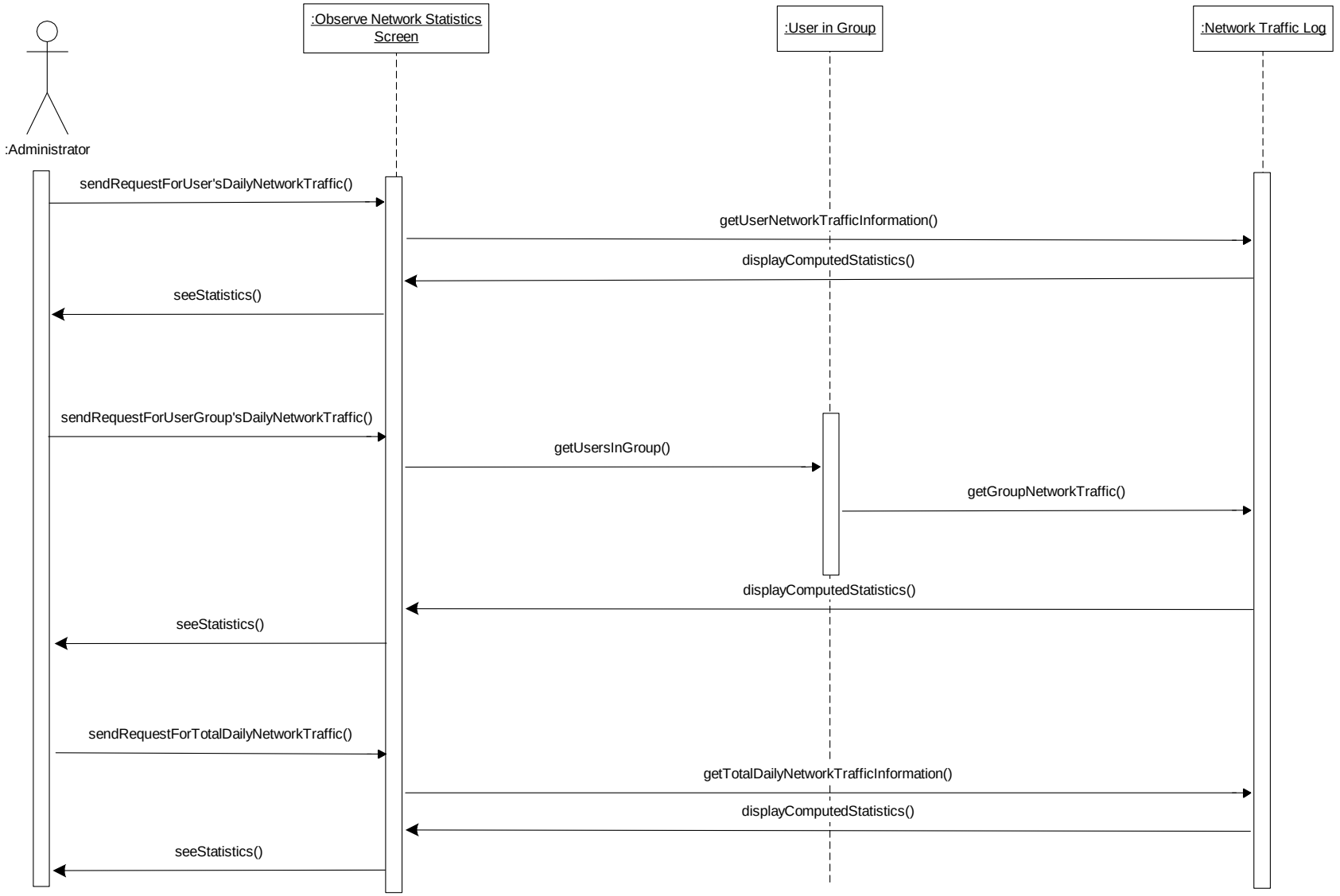


3.3.5.2 Applying URL Access Restriction

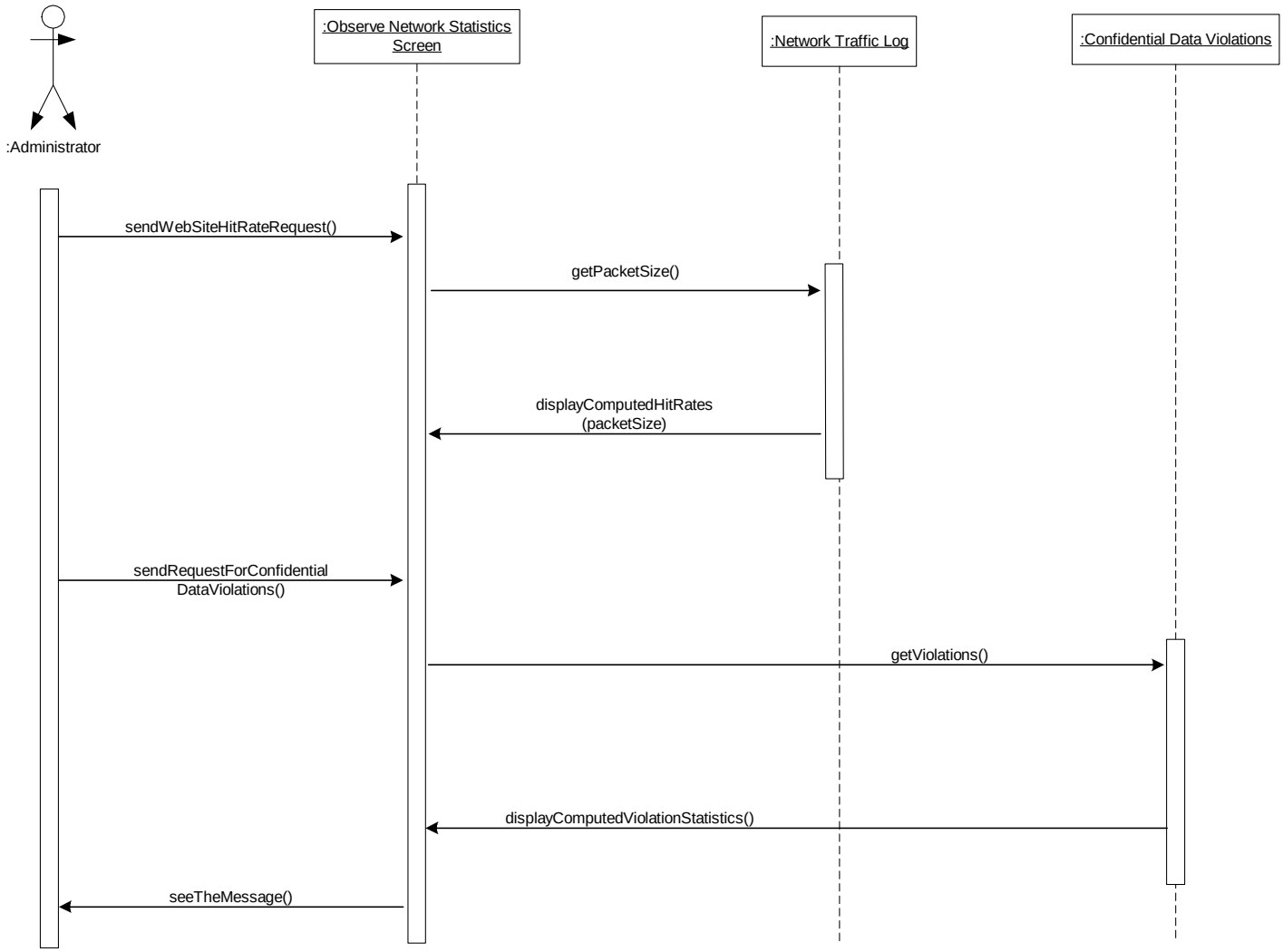


3.3.6 Statistics Module

3.3.6.1 Computing Daily Network Statistics

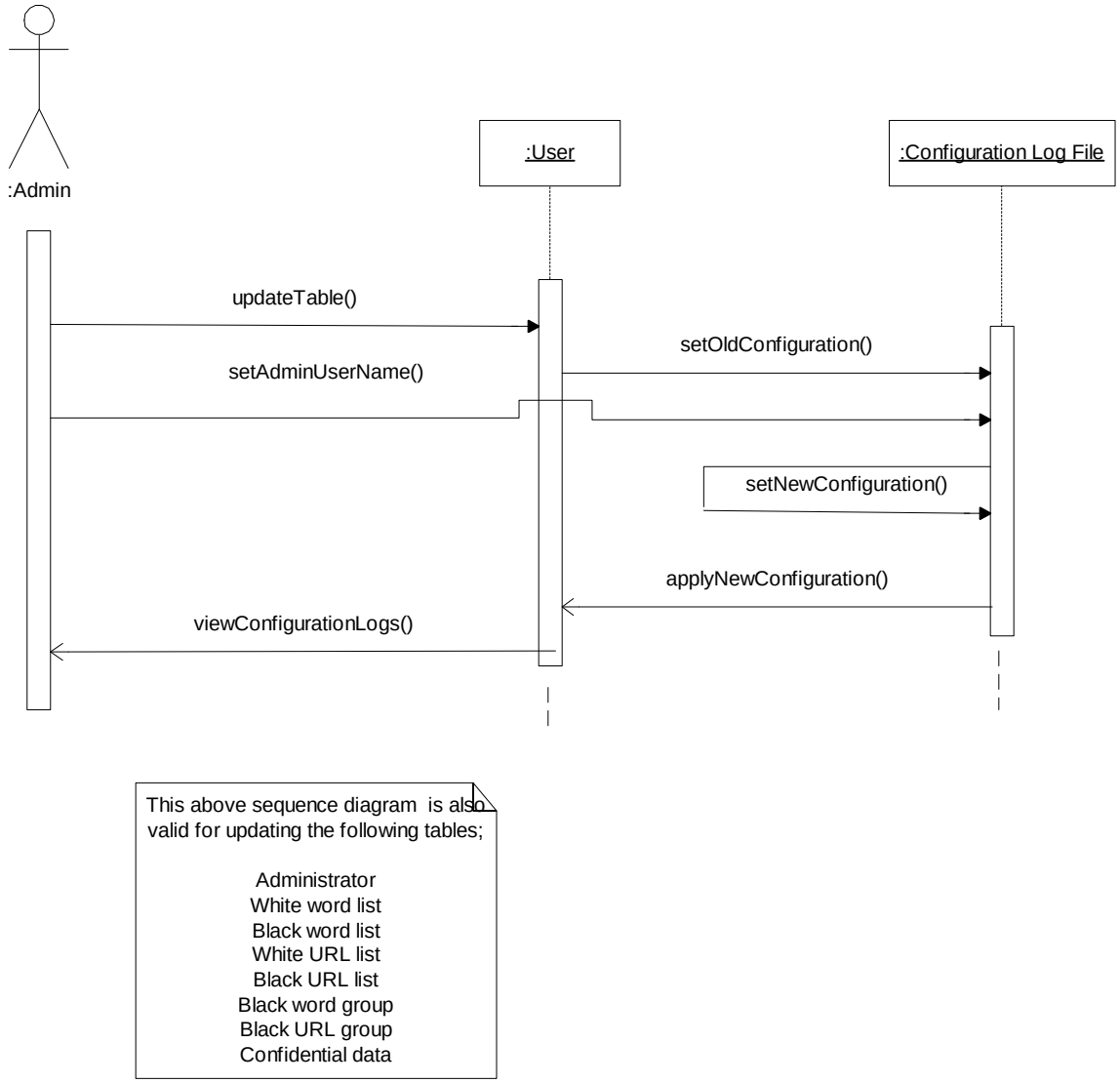


3.3.6.2 Computing Web Site Hit Rates& Confidential Data Violations



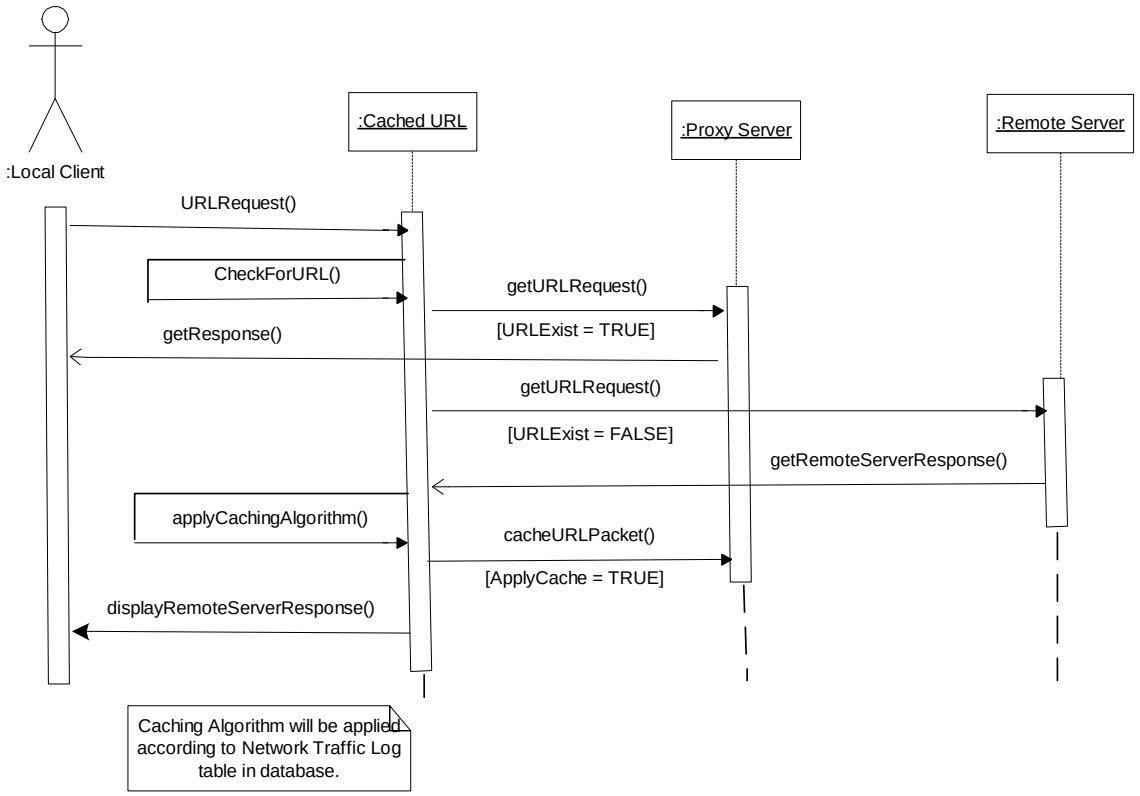
3.3.7 Logging Module

3.3.7.1 Saving the Configuration Logs



3.3.8 Caching Module

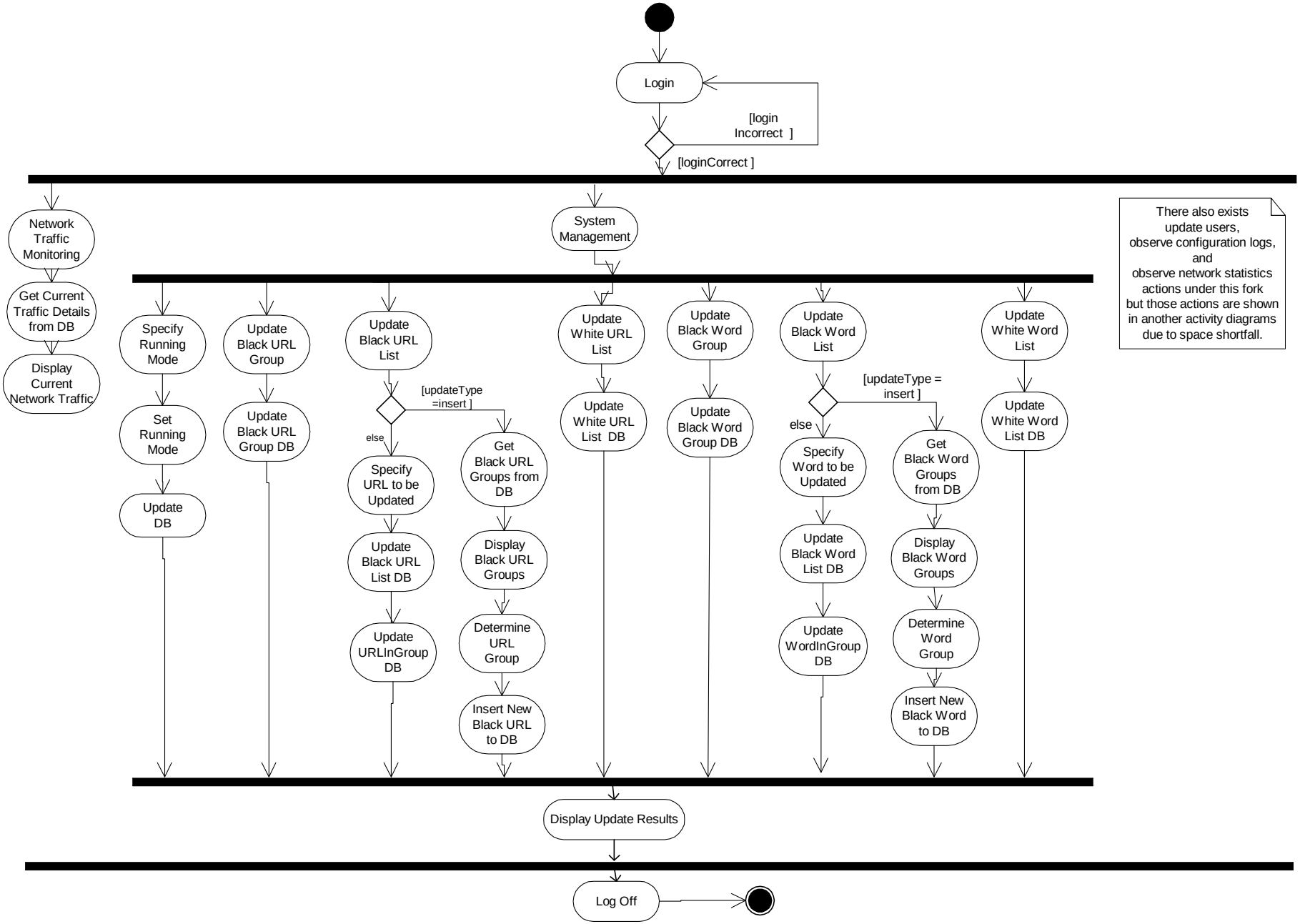
3.3.8.1 Applying the Caching Mechanism



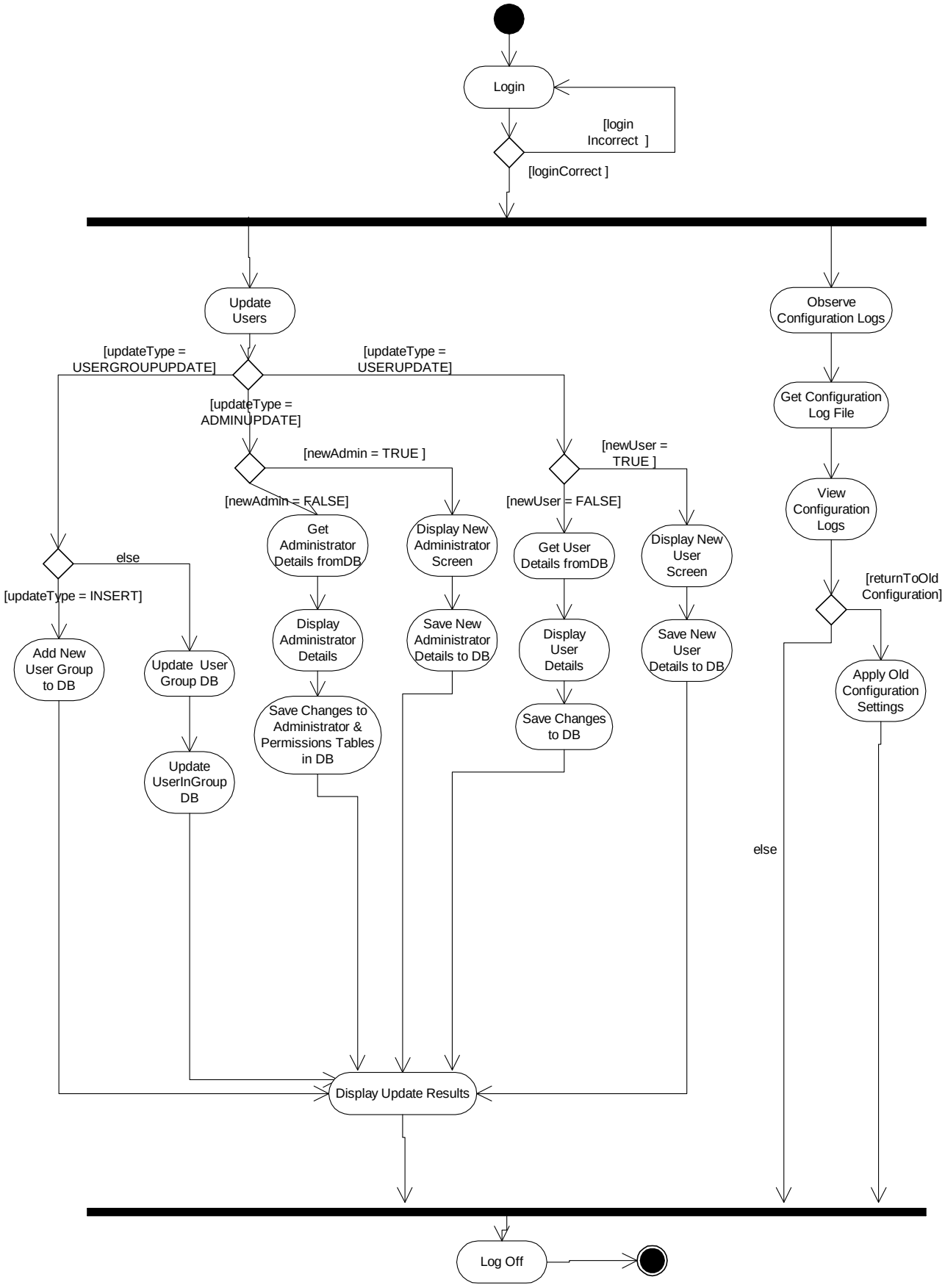
3.4 Activity Diagrams

3.4.1 Web Module

Activity diagram of the web module shows the flow of all controls that are applied by the administrator via the web interface. The web interface display functions are held by the web module and for those displays the web interface module has to interact with system management, network traffic monitoring, statistics, and logging modules. These modules are integrated into the web module in the activity diagrams for the better understandability of the system.


Login
[login Incorrect]
[loginCorrect]
Network Traffic Monitoring
Get Current Traffic Details from DB
Display Current Network Traffic
System Management
Specify Running Mode
Set Running Mode
Update DB
Update Black URL Group
Update Black URL Group DB
Update Black URL List
[updateType =insert]
else
Specify URL to be Updated
Update Black URL List DB
Update URLInGroup DB
Get Black URL Groups from DB
Display Black URL Groups
Determine URL Group
Insert New Black URL to DB
Update White URL List
Update White URL List DB
Update Black Word Group
Update Black Word Group DB
Update Black Word List
[updateType = insert]
else
Specify Word to be Updated
Update Black Word List DB
Update WordInGroup DB
Get Black Word Groups from DB
Display Black Word Groups
Determine Word Group
Insert New Black Word to DB
Update White Word List
Update White Word List DB
There also exists update users, observe configuration logs, and observe network statistics actions under this fork but those actions are shown in another activity diagrams due to space shortfall.
Display Update Results
Log Off

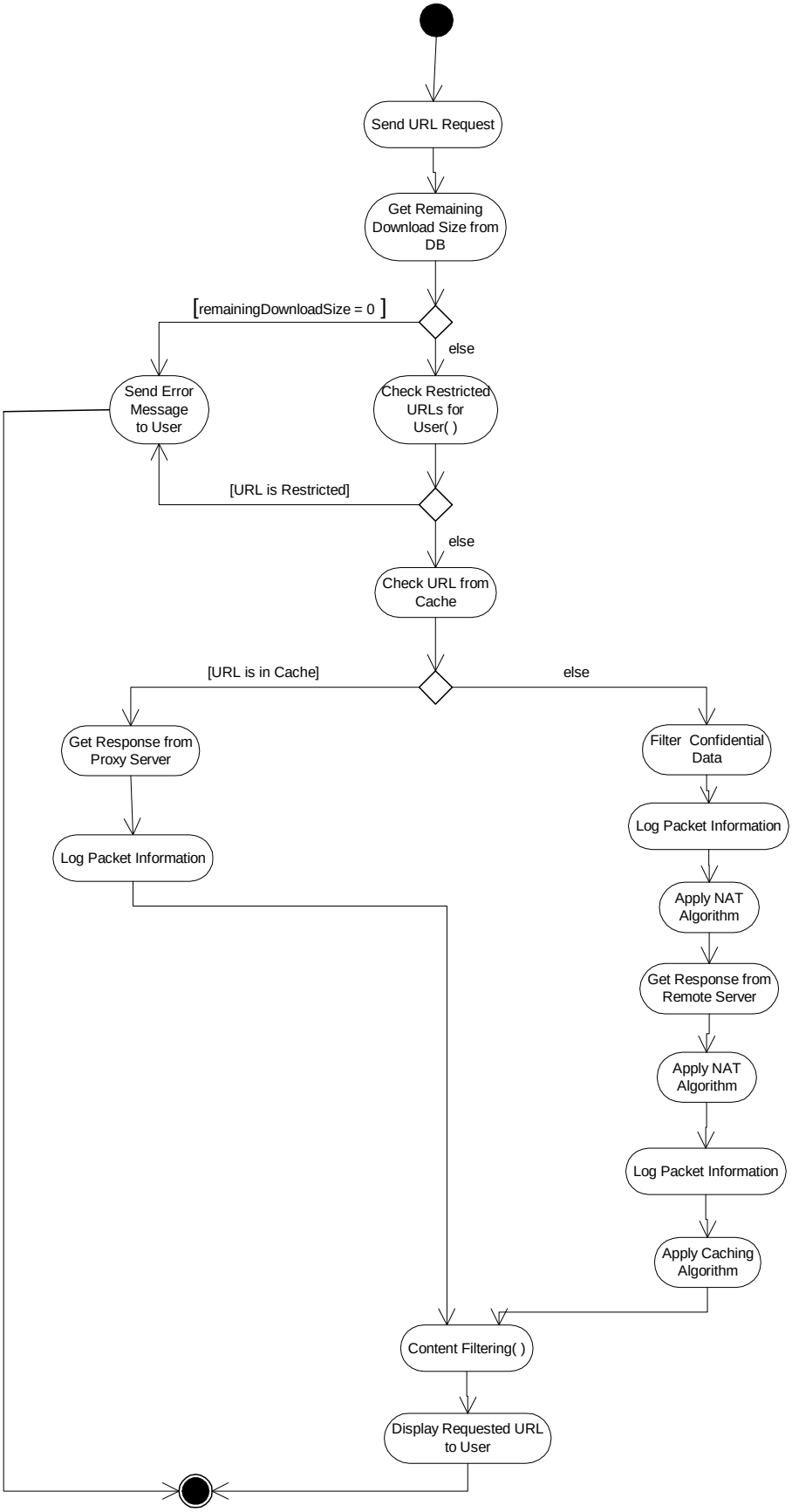

Login
[login Incorrect]
[loginCorrect]
Observe Network Statistics
Observe Statistics About Confidential Data Violations
Observe Web Site Hit Rates
Observe Network Traffic Statistics
Get Number of Violations for Users from the DB
Get Number of Requests for the URL from Network Traffic Logs
Observe Statistics for User
Observe Statistics for User Group
Observe Statistics Related to Total Daily Network Traffic
Compute Statistics
Compute Hit Rates
Get User 's Network Traffic Information from Network Traffic Logs
Get Users in the Specified Group
Get Network Traffic Information from Network Traffic Logs
Get Group's Network Traffic Information from DB
Compute Statistics
Display Statistics
Log Off

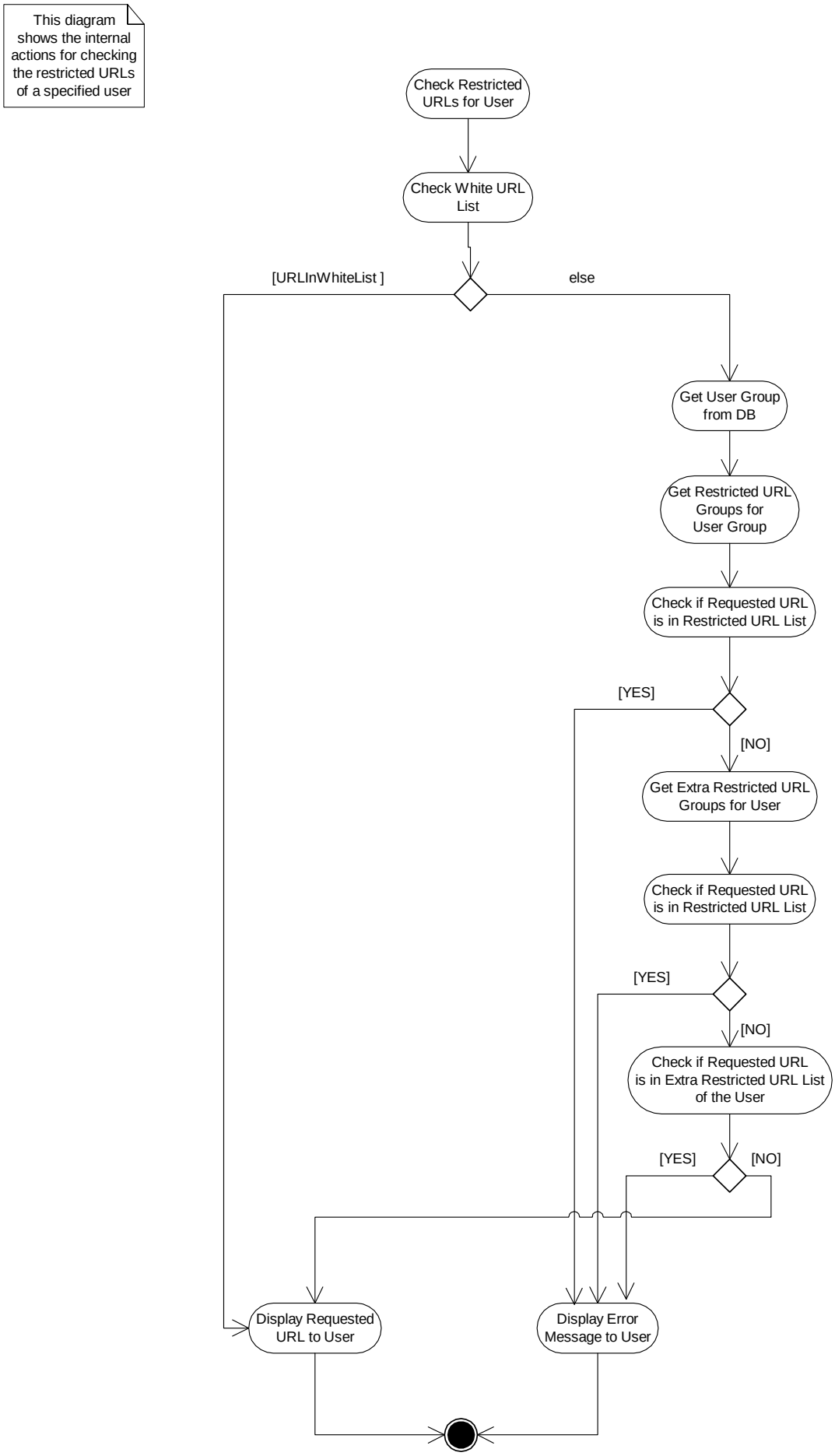

Login
[login Incorrect]
[loginCorrect]
Update Users
Observe Configuration Logs
[updateType = USERGROUPUPDATE]
[updateType = USERUPDATE]
[updateType = ADMINUPDATE]
Get Configuration Log File
[newAdmin = TRUE]
[newUser = TRUE]
View Configuration Logs
[newAdmin = FALSE]
[newUser = FALSE]
Get Administrator Details fromDB
Display New Administrator Screen
Get User Details fromDB
Display New User Screen
else
[updateType = INSERT]
[returnToOld Configuration]
Add New User Group to DB
Update User Group DB
Display Administrator Details
Save New Administrator Details to DB
Display User Details
Save New User Details to DB
Apply Old Configuration Settings
Save Changes to Administrator & Permissions Tables in DB
Save Changes to DB
Update UserInGroup DB
else
Display Update Results
Log Off

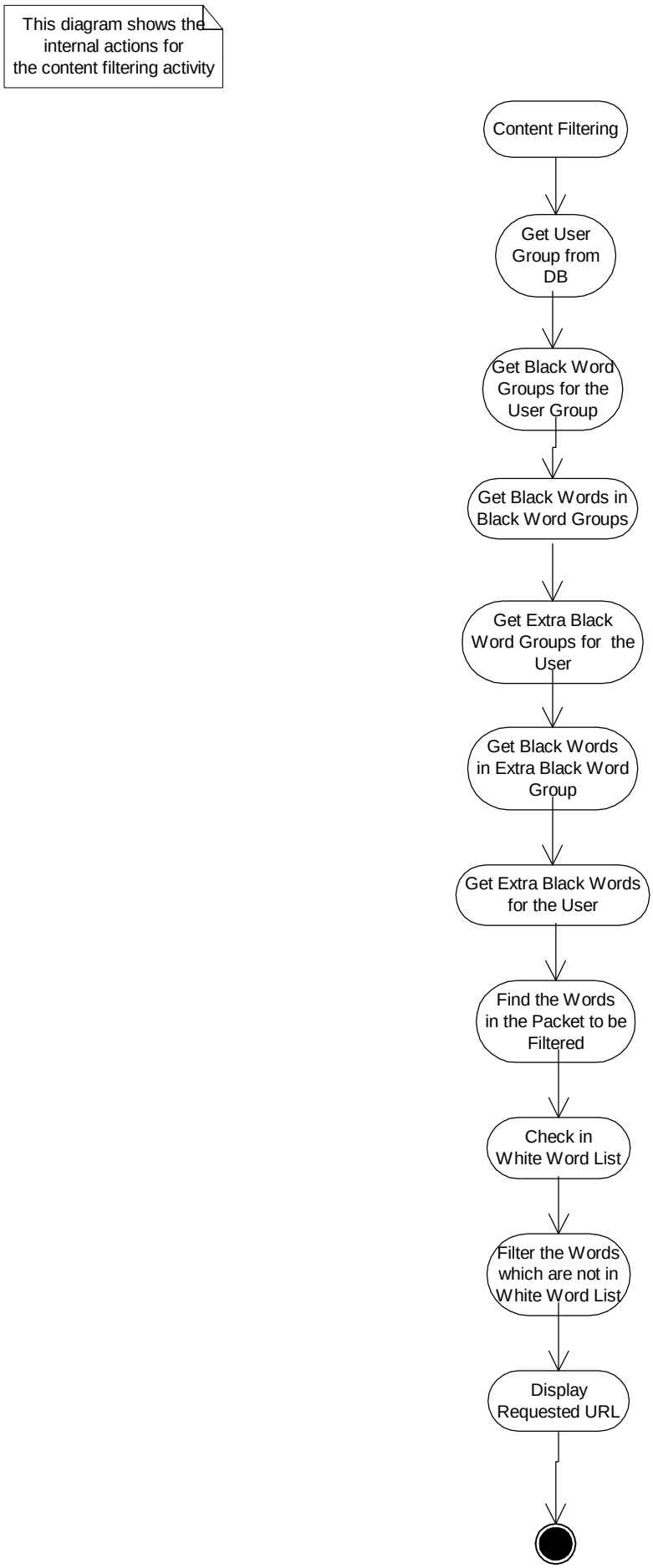
3.4.2 Restriction, Content Filtering, Caching, and Logging Modules

The activity diagram for restriction, content filtering, caching, and logging modules are also integrated again for the better understandability issues.

1.




This diagram shows the internal actions for checking the restricted URLs of a specified user
Check Restricted URLs for User
Check White URL List
[URLInWhiteList]
else
Get User Group from DB
Get Restricted URL Groups for User Group
Check if Requested URL is in Restricted URL List
[YES]
[NO]
Get Extra Restricted URL Groups for User
Check if Requested URL is in Restricted URL List
[YES]
[NO]
Check if Requested URL is in Extra Restricted URL List of the User
[YES]
[NO]
Display Requested URL to User
Display Error Message to User


This diagram shows the internal actions for the content filtering activity
Content Filtering
Get User Group from DB
Get Black Word Groups for the User Group
Get Black Words in Black Word Groups
Get Extra Black Word Groups for the User
Get Black Words in Extra Black Word Group
Get Extra Black Words for the User
Find the Words in the Packet to be Filtered
Check in White Word List
Filter the Words which are not in White Word List
Display Requested URL

4 DATABASE DESIGN

Construction of the database is the key concept in the development of our project. This is because all other modules of our system are dependent on the data modeling. If the database design is made in a complete manner, the rest of the system will also be designed concretely.

4.1 Database Table Specifications

The following table specifications explain our project's database tables in a detailed manner.

4.1.1 LocalUser

Name	Content Description	Supplementary Info.
IP	Char(15)	Primary Key
name	VARCHAR(30)	Not Null
permitted Download Size	Float	
remaining Download Size	Float	

4.1.2 NetworkTrafficLog

Name	Content Description	Supplementary Info.
IP	Char(15)	Primary Key
name	VARCHAR(30)	Not Null
permitted Download Size	Float	
remaining Download Size	Float	

4.1.3 BlackWordList

Name	Content Description	Supplementary Info.
ID	Serial	Primary Key
word	VARCHAR(20)	Not Null
isActive	boolean	

4.1.4 BlackWordGroup

Name	Content Description	Supplementary Info.
ID	Serial	Primary Key
wordGroup	VARCHAR(20)	Not Null
isActive	boolean	

4.1.5 WhiteWordList

Name	Content Description	Supplementary Info.
ID	Serial	Primary Key
word	VARCHAR(20)	
isActive	boolean	

4.1.6 WordInGroup

Name	Content Description	Supplementary Info.
wordID	Integer	Foreign Key (BlackWordList(ID))
wordGroupID	Integer	Foreign Key (BlackWordGroup(ID))

4.1.7 ConfidentialData

Name	Content Description	Supplementary Info.
ID	Serial	Primary Key
data	VARCHAR(30)	

4.1.8 BlackURLList

Name	Content Description	Supplementary Info.
ID	Serial	Primary Key
URL	VARCHAR(150)	Not Null
isActive	boolean	

4.1.9 BlackURLGroup

Name	Content Description	Supplementary Info.
ID	Serial	Primary Key
URLGroup	VARCHAR(20)	Not Null
isActive		

4.1.10 URLInGroup

Name	Content Description	Supplementary Info.
URLID	Integer	Foreign Key (BlackURLList(ID))
URLGroupID	Integer	Foreign Key (BlackURLGoup(ID))

4.1.11 WhiteURLList

Name	Content Description	Supplementary Info.
ID	Serial	Primary Key
URL	VARCHAR(150)	
isActive	boolean	

4.1.12 Administrator

Name	Content Description	Supplementary Info.
userName	VARCHAR(30)	Primary Key
password	VARCHAR(20)	
IP	Char(15)	
fullName	VARCHAR(30)	
email	VARCHAR(30)	
GSM	Char(11)	

4.1.13 Permissions

Name	Content Description	Supplementary Info.
ID	Integer	Primary Key
type	Char(20)	

4.1.14 HavePermissions

Name	Content Description	Supplementary Info.
userName	Integer	Foreign Key (Administrator(username))
permissionID	Char(20)	Foreign Key (Permissions(ID))

4.1.15 LocalUserGroup

Name	Content Description	Supplementary Info.
ID	Serial	Primary Key
groupName	VARCHAR(20)	
permittedDownloadSize	Float	

4.1.16 RestrictedURLforUserGroup

Name	Content Description	Supplementary Info.
userGroupID	Integer	Foreign Key (LocalUserGroup(ID))
blackURLGroupID	Integer	Foreign Key (BlackURLGroup(ID))

4.1.17 RestrictedWordforUserGroup

Name	Content Description	Supplementary Info.
userGroupID	Integer	Foreign Key (LocalUserGroup(ID))
blackWordGroupID	Integer	Foreign Key (BlackWordGroup(ID))

4.1.18 UserInGroup

Name	Content Description	Supplementary Info.
userIP	Char(15)	Foreign Key (LocalUser(IP))
userGroupID	Integer	Foreign Key (LocalUserGroup(ID))

4.1.19 ExtraURLforUser

Name	Content Description	Supplementary Info.
userIP	Char(15)	Foreign Key (LocalUser(IP))
blackURLID	Integer	Foreign Key (BlackURLList(ID))

4.1.20 ExtraWordforUser

Name	Content Description	Supplementary Info.
userIP	Char(15)	Foreign Key (LocalUser(IP))
blackWordID	Integer	Foreign Key (BlackWordList(ID))

4.1.21 ExtraURLGroupforUser

Name	Content Description	Supplementary Info.
userIP	Char(15)	Foreign Key (LocalUser(IP))
blackURLGroupID	Integer	Foreign Key (BlackURLGroup(ID))

4.1.22 ExtraWordGroupforUser

Name	Content Description	Supplementary Info.
userIP	Char(15)	Foreign Key (LocalUser(IP))
blackWordGroupID	Integer	Foreign Key (BlackWordGroup(ID))

4.1.23 ConfidentialDataViolations

Name	Content Description	Supplementary Info.
userIP	Char(15)	Foreign Key (LocalUser(IP))
violatedRuleID	Serial	Foreign Key (ConfidentialData(ID))
time	Timestamp	

4.1.24 CachedURL

Name	Content Description	Supplementary Info.
URLID	Integer	Primary Key
URL	VARCHAR(150)	

4.1.25 RunningMode

Name	Content Description	Supplementary Info.
modeID	Integer	Primary Key
isActive	boolean	

4.2 Database Table SQL's

After the table specifications, we have also constructed the database of our project in PostgreSQL with the following sql create commands:

```
create LocalUser(
        IP char(15),
        name varchar(30) not null,
        permittedDownloadSize float,
        remainingDownloadSize float,
        primary key(IP)
);

create table NetworkTrafficLog(
        communicationID serial primary key,
        sourceIP char(15) not null,
        destinationIP char(15) not null,
        destinationURL varchar(150),
        packetSize float,
        time timestamp
);

create table BlackWordList (
        ID serial primary key,
        word varchar(20) not null,
        isActive boolean
);

create table BlackWordGroup (
        ID serial primary key,
        wordGroup varchar(20) not null,
        isActive boolean
);

create table WhiteWordList (
        ID serial primary key,
        word varchar(20) not null,
        isActive boolean
);

create table WordInGroup (
        wordID integer,
        wordGroupID integer,
        primary key(wordID, wordGroupID),
        foreign key (wordID) references blackWordList on delete cascade,
        foreign key (wordGroupID) references blackWordGroup on delete cascade
);
```

```
create table ConfidentialData (
        ID serial primary key,
        data varchar(30)
);

create table BlackURLList (
        ID serial primary key,
        URL varchar(150) not null,
        isActive boolean
);

create table BlackURLGroup (
        ID serial primary key,
        URLGroup varchar(20) not null,
        isActive boolean
);

create table URLInGroup (
        URLID integer,
        URLGroupID integer,
        primary key(URLID, URLGroupID),
        foreign key (URLID) references blackURLList on delete cascade,
        foreign key (URLGroupID) references blackURLGroup on delete cascade
);

create table WhiteURLList (
        ID serial primary key,
        URL varchar(150) not null,
        isActive boolean
);

create table Administrator(
        userName varchar(30),
        password varchar(20),
        IP char(15),
        fullName varchar(30),
        email varchar(30),
        GSM char(11),
        primary key (userName)
);

create table Permissions (
        ID integer,
        type char(20),
        primary key (ID)
);
```

```
create table HavePermissions (
        userName varchar(30),
        permissionID integer,
        primary key (userName, permissionID) ,
        foreign key (userName) references administrator on delete cascade on
                update cascade,
        foreign key (permissionID) references permissions(ID) on delete
                cascade on update cascade
);

create table LocalUserGroup(
        ID serial primary key,
        groupName varchar(20),
        permittedDownloadSize float
);

create table RestrictedURLforUserGroup(
        userGroupID integer,
        blackURLGroupID integer,
        primary key(userGroupID,blackURLGroupID),
        foreign key(userGroupID) references LocalUserGroup(ID) on delete
                cascade,
        foreign key(blackURLGroupID) references blackURLGroup(ID) on delete
                cascade
);

create table RestrictedWordforUserGroup(
        userGroupID integer,
        blackWordGroupID integer,
        primary key(userGroupID,blackWordGroupID),
        foreign key(userGroupID) references LocalUserGroup(ID) on delete
                cascade,
        foreign key(blackWordGroupID) references blackWordGroup(ID) on delete
                cascade
);

create table UserInGroup(
        userIP char(15),
        userGroupID integer,
        primary key(userIP,userGroupID),
        foreign key (userIP) references LocalUser(IP) on delete cascade on
                update cascade,
        foreign key(userGroupID) references LocalUserGroup(ID) on delete
                cascade
);
```

```
create table ExtraURLforUser(
        userIP char(15),
        blackURLId integer,
        primary key(userIP,blackURLId),
        foreign key (userIP) references LocalUser(IP) on delete cascade on
                update cascade,
        foreign key (blackURLId) references blackURLList(ID) on delete
                cascade
);

create table ExtraWordforUser(
        userIP char(15),
        blackWordId integer,
        primary key(userIP,blackWordId),
        foreign key (userIP) references LocalUser(IP) on delete cascade on
                update cascade,
        foreign key (blackWordId) references blackWordList(ID) on delete
                cascade
);

create table ExtraURLGroupforUser(
        userIP char(15),
        blackURLGroupId integer,
        primary key(userIP,blackURLGroupId),
        foreign key (userIP) references LocalUser(IP) on delete cascade on
                update cascade,
        foreign key (blackURLGroupId) references blackURLGroup(ID) on delete
                cascade
);

create table ExtraWordGroupforUser(
        userIP char(15),
        blackWordGroupId integer,
        primary key(userIP,blackWordGroupId),
        foreign key (userIP) references LocalUser(IP) on delete cascade on
                update cascade,
        foreign key (blackWordGroupId) references blackWordGroup(ID) on
                delete cascade
);

create table ConfidentialDataViolations(
        userIP Char(15),
        violatedRuleID serial,
        time timestamp,
        primary key(userIP,violatedRuleID),
        foreign key (userIP) references LocalUser(IP) on delete cascade on
                update cascade,
        foreign key (violatedRuleID) references confidentialData(ID) on
                delete cascade
);
```

```
create table CachedURL (
        URLID integer,
        URL varchar(150),
        primary key (URLID)
);

create table RunningMode (
        ID integer,
        isActive boolean,
        primary key (ID)
);
```

5 SYNTAX SPECIFICATION

As a part of standardization process for rapid development of our project, we have decided to develop special naming conventions. In addition, we have decided the syntax of descriptive comments that will be used for understandability and maintainability of our product. The details of the specifications are described below.

5.1 Naming Conventions

We have decided that all names should be comprehensible. For names that are composed of more than one word, lower case/upper case characters will be used to distinguish between consecutive words.

Naming the Classes:

All classes will have names beginning with a capital letter. The classes with more than one word will have a capital letter at the beginning of each word. For instance, “NetworkTrafficLog” is a valid class name for our project.

Naming the Class Attributes:

Attributes will begin with a lower case letter. In case there are more words, they will be distinguished by capital letters at the beginning. “groupName” and “word” are valid class attribute examples.

Naming the Class Methods:

Methods will have the same convention with the class attributes.

Naming the Database Table:

Names of the tables in the database will begin with capital letters and will continue with a capital letter for each consecutive word. Attributes of the tables will follow the naming convention for the class attributes; that is, will begin with lower case letters and continue with upper case letters for each new word.

Naming the Files:

Files that include the source code and header for a class will be named as the following respectively:

<class_name><.cpp>

<class_name><.h>

Naming the Global and Local Variables:

We will try to avoid using global variables as an appropriate software engineering principle. However, in case of any necessity, global variables will be prefixed with "g_", since usage of global variables significantly decrease the understandability of the source code. Likewise, pointers will be prefixed by "p_". Since we will implement our project in C++, we will need extensive use of pointers. For local variables, we have decided to use a convention that will help differentiate the type of the variable, such as "an_int" for a variable of integer type.

5.2 Commenting Conventions

In order to increase the understandability of our source code appropriate commenting is an important concern. We are intending to use comments for file descriptions, for function definitions, and for not easily understood variables. Commenting style for our project is described as follows;

Commenting the Files:

Files should be described at the beginning according to the following format;

```
/* ------------------------------------------------------------------------------
/* File name:
/* Created by:
/* Created at: ( Date:DD.MM.YY – Time: HH:MM:SS)
/* Modified by:
/* Modified at: ( Date:DD.MM.YY – Time: HH:MM:SS)
/* Version:
/* Description:
------------------------------------------------------------------------------*/
```

Commenting the Functions:

For the description of the functions we have specified the following format;

```
/*-------------------------------------------------------------------------------
/* Function Signature: <return_type> <function_name> (<param_1>,<param_2>,..)
/* Parameters: <parameter_name> <parameter_description>
/* Return value: <return_value> <return_value_description>
/* Function Description:
-------------------------------------------------------------------------------*/
```

Commenting the Variables:

At the point of variable declaration a brief description could be added as follows;

```
<variable_type><variable_name> // variable description
```

6 HARDWARE AND SOFTWARE SPECIFICATIONS

6.1 Software Specifications

The system should provide a Linux operating system with the following facilities for our program to run:

- Apache web server,
- PostgreSQL as the Database Management System,
- A firewall (Iptables),
- A web browser for the administrative purposes,
- GNU C++ compiler.

6.2 Hardware Specifications

The following hardware should be provided for our program to run appropriately:

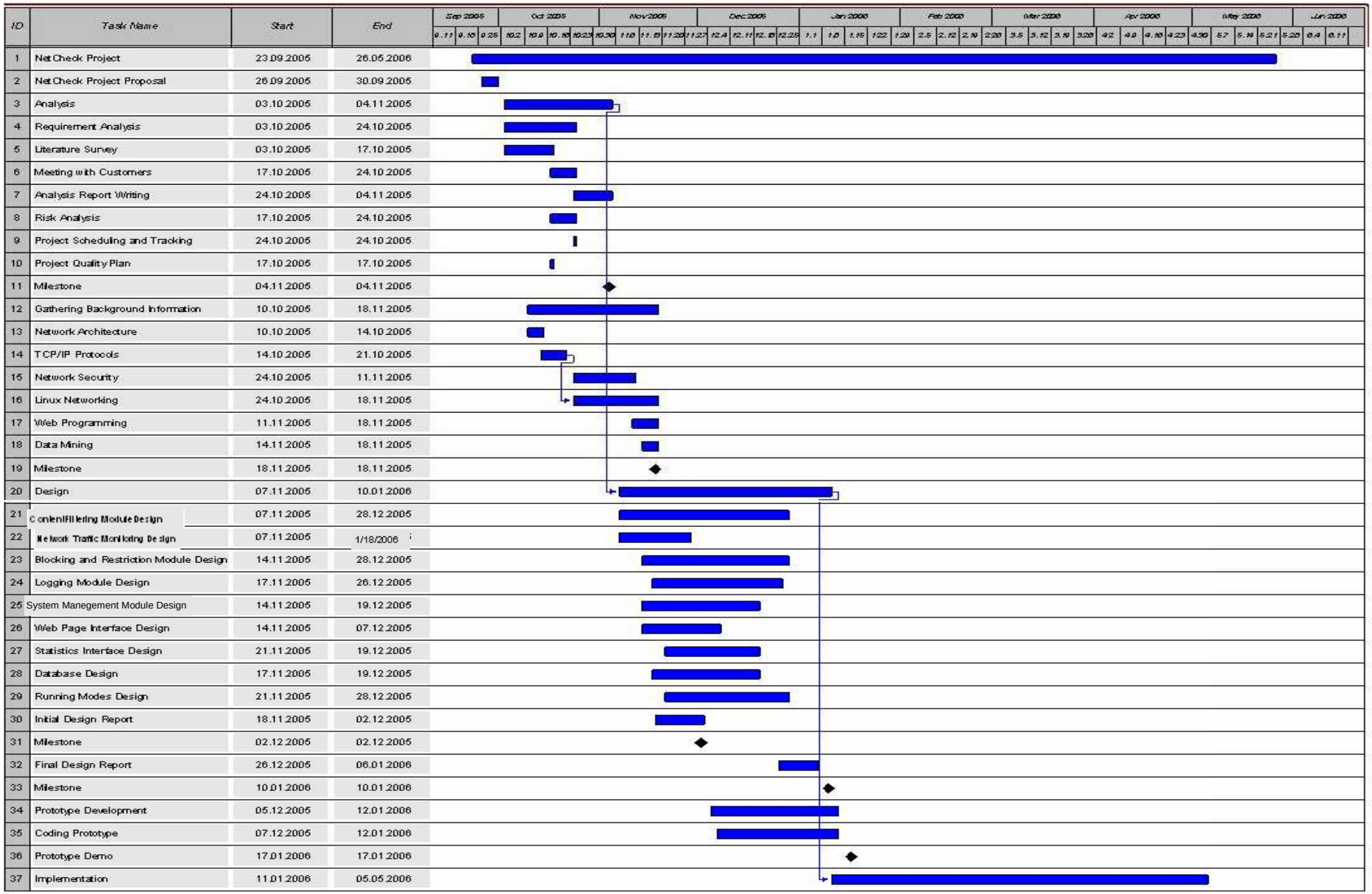
- Minimum 512 MB RAM,
- Minimum 5 GB of free disk space, for database storage,
- A Pentium IV processor,
- Minimum two network interface cards.

6.3 Tool Specifications

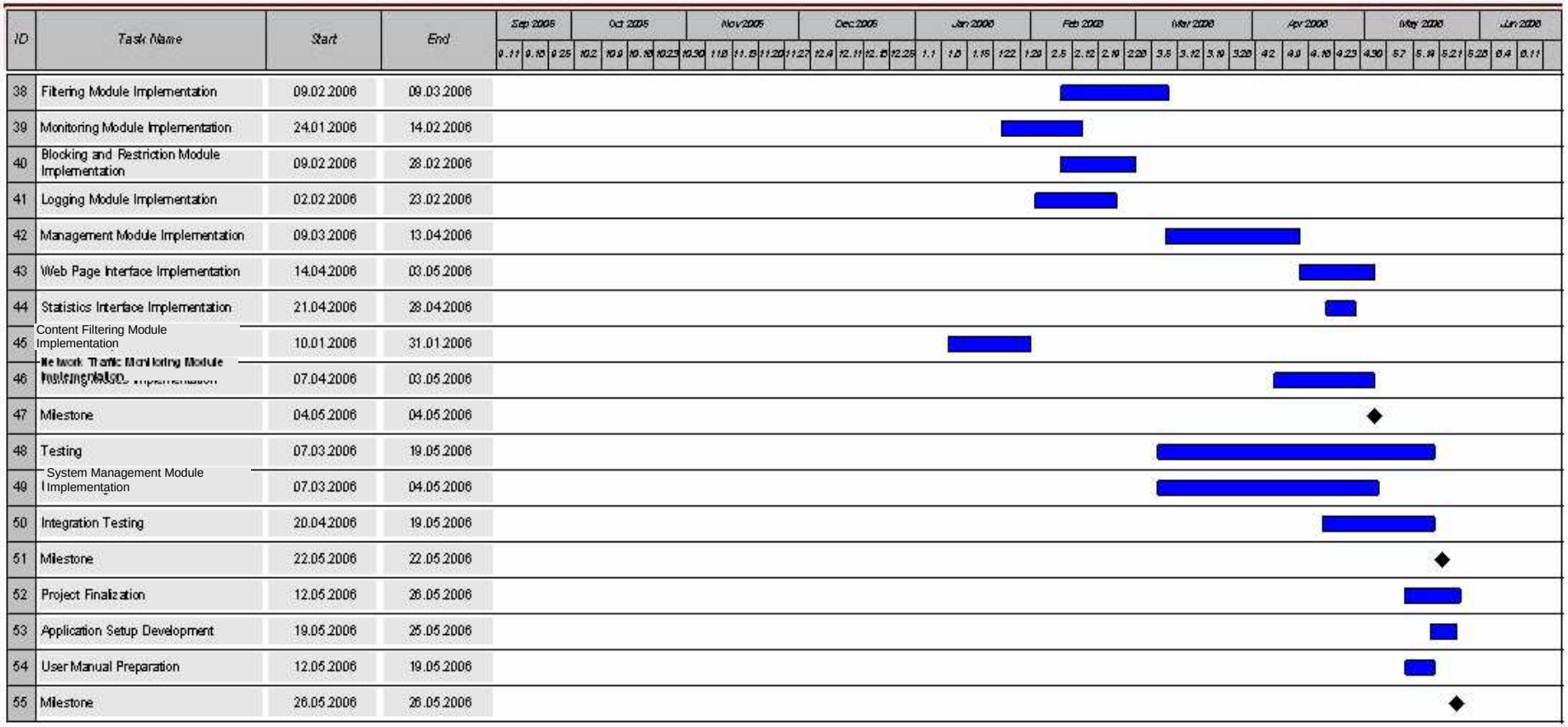
In the implementation phase of our project, we have determined to use the following tools:

- Linux operating system as the development platform,
- C++ programming language,
- GNU C++ compiler,
- PostgreSQL Database Management System,
- PHP and Apache web server.

7 UPDATED GANNT CHART



ID	Task Name	Start	End
1	NetCheck Project	23.09.2005	26.05.2006
2	NetCheck Project Proposal	26.09.2005	30.09.2005
3	Analysis	03.10.2005	04.11.2005
4	Requirement Analysis	03.10.2005	24.10.2005
5	Literature Survey	03.10.2005	17.10.2005
6	Meeting with Customers	17.10.2005	24.10.2005
7	Analysis Report Writing	24.10.2005	04.11.2005
8	Risk Analysis	17.10.2005	24.10.2005
9	Project Scheduling and Tracking	24.10.2005	24.10.2005
10	Project Quality Plan	17.10.2005	17.10.2005
11	Milestone	04.11.2005	04.11.2005
12	Gathering Background Information	10.10.2005	18.11.2005
13	Network Architecture	10.10.2005	14.10.2005
14	TCP/IP Protocols	14.10.2005	21.10.2005
15	Network Security	24.10.2005	11.11.2005
16	Linux Networking	24.10.2005	18.11.2005
17	Web Programming	11.11.2005	18.11.2005
18	Data Mining	14.11.2005	18.11.2005
19	Milestone	18.11.2005	18.11.2005
20	Design	07.11.2005	10.01.2006
21	ContentFiltering Module Design	07.11.2005	28.12.2005
22	Network Traffic Monitoring Design	07.11.2005	1/18/2006
23	Blocking and Restriction Module Design	14.11.2005	28.12.2005
24	Logging Module Design	17.11.2005	26.12.2005
25	System Manegement Module Design	14.11.2005	19.12.2005
26	Web Page Interface Design	14.11.2005	07.12.2005
27	Statistics Interface Design	21.11.2005	19.12.2005
28	Database Design	17.11.2005	19.12.2005
29	Running Modes Design	21.11.2005	28.12.2005
30	Initial Design Report	18.11.2005	02.12.2005
31	Milestone	02.12.2005	02.12.2005
32	Final Design Report	26.12.2005	06.01.2006
33	Milestone	10.01.2006	10.01.2006
34	Prototype Development	05.12.2005	12.01.2006
35	Coding Prototype	07.12.2005	12.01.2006
36	Prototype Demo	17.01.2006	17.01.2006
37	Implementation	11.01.2006	05.05.2006



ID	Task Name	Start	End
38	Filtering Module Implementation	09.02.2006	09.03.2006
39	Monitoring Module Implementation	24.01.2006	14.02.2006
40	Blocking and Restriction Module Implementation	09.02.2006	28.02.2006
41	Logging Module Implementation	02.02.2006	23.02.2006
42	Management Module Implementation	09.03.2006	13.04.2006
43	Web Page Interface Implementation	14.04.2006	03.05.2006
44	Statistics Interface Implementation	21.04.2006	28.04.2006
45	Content Filtering Module Implementation	10.01.2006	31.01.2006
46	Network Traffic Monitoring Module Implementation	07.04.2006	03.05.2006
47	Milestone	04.05.2006	04.05.2006
48	Testing	07.03.2006	19.05.2006
49	System Management Module Implementation	07.03.2006	04.05.2006
50	Integration Testing	20.04.2006	19.05.2006
51	Milestone	22.05.2006	22.05.2006
52	Project Finalization	12.05.2006	26.05.2006
53	Application Setup Development	19.05.2006	25.05.2006
54	User Manual Preparation	12.05.2006	19.05.2006
55	Milestone	26.05.2006	26.05.2006