SMARTECH

NetCheck Project
Initial Design Report

Neslihan Bulut
Kezban Demirtaş
Hande Çelikkanaat
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
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

- Monitor Network Traffic
- Restrict URL Access
- Add/Remove Confidential Data
- Add/Remove Black & White Word List
- Specify the Running Mode of the Program
- Configure Restricted Time Intervals for Specified URLs
- Add/Remove URL Black & White List
- View Configuration Reports
- Define Security Policies
- Get Alerts
- Define Security Policies

Diagram:

Admin:
- Monitor Network Traffic
  - Restrict URL Access
    - Add/Remove Confidential Data
      - Add/Remove Black & White Word List
        - Specify the Running Mode of the Program
          - Configure Restricted Time Intervals for Specified URLs
            - Add/Remove URL Black & White List
              - View Configuration Reports
                - Define Security Policies
                  - Get Alerts
Observe Network Statistics

Interrupt Current Network Traffic via the Web Interface

Observe Persistent Attacks

Observe Daily Network Traffic

Observe Download Information

Observe web Sites with Their Hit Rates and Requesting IPs

Define Access & Download Rights for the Local IPs

Add User

Administrator
3.1.2 Use Case for Local Client and Server

- Local Client
  - Send Request
    - <<include>> Check Access Rights
    - <<extend>> Hide Internal Information
    - Get Response
      - <<include>> Filter Content
      - <<include>> Log Incoming Packet
  - Get Request

- Local Server
  - Send Response
    - <<include>> Hide Internal Information
3.1.3 Use Case for Remote Client and Server

Remote Client

Get Request

Get Response

Remote Server

Send Request

Check Blocked Client List

Detect Intrusion Attempts

<<include>>

<<extend>>
3.2 Class Diagrams

![Class Diagram Image]
packetParser

+parseData()
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

Running Modes are:
- Free mode
- Normal Mode
- Secure Mode
3.3.2.2  Update Users of the System

getUserDetails()  
Message1()  
displayNewUserScreen()  
updateUser()  
getUserDetails()  
getUserGroupDetails()  
getExtraURLsForUser()  
getExtraURLGroupForUser()  
getExtraWordForUser()  
getExtraWordGroupForUser()  
displayUserDetails()  
displayUpdateMessage()  
updateUserDatabase()  
updateUserInGroupDB()  
updateExtraURLDB()  
updateExtraURLGroupDB()  
updateExtraWordDB()  
updateExtraWordGroupDB()  
seeTheMessage()
3.3.2.3 Update User Groups of the System

- Admn: updateUserGroupRequest()
- Update User Screen: updateUserGroup()
- User Group: displayUpdateMessage()
- seetheMessage()
3.3.2.4 **Update Administrators of the System**

- **Update Administrator Screen**: 
  - `updateAdministrator()`
  - `getAdminDetails()`
  - `displayAdminDetails()`
  - `displayNewAdminScreen()`
  - `updateAdminDatabase()`
  - `displayAdminDetails()`
  - `updateBlackURLList()`
  - `updateHavePermissionsDatabase()`
  - `displayUpdateMessage()`
  - `seeTheMessage()`

- **Administrator**: 
  - `updateAdministrator()`
  - `getAdminDetails()`
  - `displayAdminDetails()`
  - `displayNewAdminScreen()`
  - `updateAdminDatabase()`
  - `displayUpdateMessage()`
  - `seeTheMessage()`

- **Have Permissions**: 
  - `updateBlackURLList()`
  - `updateHavePermissionsDatabase()`
  - `displayUpdateMessage()`
  - `seeTheMessage()`

- **Enter Admin Info**: 
  - `enterAdminInfo()`
  - `newAdmin = FALSE`
  - `displayAdminDetails()`
  - `newAdmin = TRUE`
  - `updateAdminDatabase()`
  - `displayUpdateMessage()`

- **Update Permission Request**: 
  - `updatePermissionRequest()`
  - `updatePermission()`
  - `displayUpdateMessage()`
  - `seeTheMessage()`
### 3.3.2.5 Update Black URL List of the System

This is not an method of the administrator class, but makes the system more understandable.

Update method includes update & delete functionalities.
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

**System Management Screen**

- insertBlackWord()
- seeBlackWordGroups()
- determineBlackWordGroup()
- [adminSpecifiedGroup = TRUE]
  - determineBlackWordGroup()
  - updateBlackWordList()
- [adminSpecifiedGroup = FALSE]
  - seeTheMessage()
- displayUpdateMessage()

**Black Word Group**

- getBlackWordGroups()
- displayBlackWordGroups()

**Black Word List**

- updateBlackWordList()
- displayUpdateMessage()

**Administrator**

- updateUserBlackWord()
- updateWordInGroupTable()
- updateBlackWordList()
- displayUpdateMessage()

Update method includes update & delete functionalities.
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

Current traffic means recent 10 minutes traffic.
3.3.4 Content Filtering Module

3.3.4.1 Applying Content Filtering

- Remote Server
- User Groups
- Black Word Group
- Black Word List
- Extra Word Group
- Extra Word
- White Word List
- User

=getUserGroup();

URLRequest();

getBlackWordGroup(User Group);

checkBlackWords(Packet Data, User Group);

checkExtraWordGroup(User);

checkExtraWord(Packet Data);

checkBlackWords(Packet Data);

[EntriesFound = TRUE]

checkWhiteWords(Packet Data);

BlackWordFound = FALSE

skipFiltering();

[checkInWhiteWords = TRUE]

filter();

[checkInWhiteWords = FALSE]

getPacket();

BlackWordFound = TRUE

getPacket();
3.3.4.2 Applying Confidential Data Filtering

![Diagram showing the process of applying confidential data filtering]

- Local Client
  - sendRequestPacket()
  - filterConfidentialData()
  - addViolation()
  - [FILTERED = TRUE]
  - sendFilteredPacket()
  - getResponse()

- Remote Server
  - filterConfidentialData()
  - [FILTERED = TRUE]
  - sendResponsePacket()

- Remote Client
  - sendRequest()
  - filterConfidentialData()
  - addViolation()
  - [FILTERED = TRUE]
  - getResponse()
3.3.5 Restriction Module

3.3.5.1 Applying Download Restriction

```
sendURLRequest()

getUserResponse()

checkRemainingDownloadSize()

[Remaining Download Size > 0]

sendURLRequest()

[Remaining Download Size = 0]
```
3.3.5.2 Applying URL Access Restriction

This URL request first passes through Download Check, Module, Hide Confidential Data and Logging Modules.

getUserGroup()

getUserGroup()

[WhiteList = FALSE]

getBlackURLGroup

checkInBlackURLList(URL Request)

getRestrictedAccessMessage()

[CheckInBlackURLList = TRUE]

checkExtraURLGroup(user)

[checkInBlackURLList = FALSE]

sendURLRequest()

getServerResponse()

[NoEntriesFound(user) = TRUE]

checkInBlackURLList(URLRequest)

[NoEntriesFound(user) = FALSE]

getRestrictedAccessMessage()

[checkInBlackList = TRUE]

checkExtraUser(user)

[CheckInBlackURLList = FALSE]

sendURLRequest()

getServerResponse()

[CheckExtraURL = FALSE]

[CheckExtraURL = TRUE]
3.3.6 Statistics Module

3.3.6.1 Computing Daily Network Statistics

Administrator

Observe Network Statistics

Screen

User in Group

Network Traffic Log

sendRequestForUser'sDailyNetworkTraffic()

seeStatistics()

getUserNetworkTrafficInformation()

displayComputedStatistics()

getUsersInGroup()

getGroupNetworkTraffic()

displayComputedStatistics()

sendRequestForUserGroup'sDailyNetworkTraffic()

seeStatistics()

sendRequestForTotalDailyNetworkTraffic()

seeStatistics()

getUserNetworkTrafficInformation()

displayComputedStatistics()

getGroupNetworkTraffic()

displayComputedStatistics()
3.3.6.2 Computing Web Site Hit Rates & Confidential Data Violations

Administrator

- sendWebSiteHitRateRequest()
- getPacketSize()
- displayComputedHitRates(packetSize)
- displayComputedViolationStatistics()
- seeTheMessage()
- sendRequestForConfidentialDataViolations()
- getViolations()
3.3.7 Logging Module

3.3.7.1 Saving the Configuration Logs

This above sequence diagram is also valid for updating the following tables:

- Administrator
- White word list
- Black word list
- White URL list
- Black URL list
- Black word group
- Black URL group
- Confidential data
3.3.8  Caching Module

3.3.8.1  Applying the Caching Mechanism

Caching Algorithm will be applied according to Network Traffic Log table in database.
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.
Network Traffic Monitoring
- Get Current Traffic Details from DB
- Display Current Network Traffic

System Management
- Login
  - [login Incorrect ]
  - [loginCorrect ]
- Specify Running Mode
- Update Black URL Group
- Update Black URL List
  - [updateType = insert ]
  - else Specify URL to be Updated
- Update Black URL Group DB
- Update URL In Group DB
- Update DB
- Display Update Results
- Log Off

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.
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.

1. **Check Restricted URLs for User**
2. **Check White URL List**
   - **[URLInWhiteList]**
   - **else**
     1. **Get User Group from DB**
     2. **Get Restricted URL Groups for User Group**
     3. **Check if Requested URL is in Restricted URL List**
        - **[YES]**
          1. **Get Extra Restricted URL Groups for User**
          2. **Check if Requested URL is in Extra Restricted URL List of the User**
             - **[YES]**
             - **[NO]**
        - **[NO]**
     - **[NO]**
3. **Display Requested URL to User**
4. **Display Error Message to User**
This diagram shows the internal actions for the content filtering activity.

1. Content Filtering
2. Get User Group from DB
3. Get Black Word Groups for the User Group
4. Get Black Words in Black Word Groups
5. Get Extra Black Word Groups for the User
6. Get Black Words in Extra Black Word Group
7. Get Extra Black Words for the User
8. Find the Words in the Packet to be Filtered
9. Check in White Word List
10. Filter the Words which are not in White Word List
11. 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>Char(15)</td>
<td>Primary Key</td>
</tr>
<tr>
<td>name</td>
<td>VARCHAR(30)</td>
<td>Not Null</td>
</tr>
<tr>
<td>permitted Download Size</td>
<td>Float</td>
<td></td>
</tr>
<tr>
<td>remaining Download Size</td>
<td>Float</td>
<td></td>
</tr>
</tbody>
</table>

4.1.2 NetworkTrafficLog

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>Char(15)</td>
<td>Primary Key</td>
</tr>
<tr>
<td>name</td>
<td>VARCHAR(30)</td>
<td>Not Null</td>
</tr>
<tr>
<td>permitted Download Size</td>
<td>Float</td>
<td></td>
</tr>
<tr>
<td>remaining Download Size</td>
<td>Float</td>
<td></td>
</tr>
</tbody>
</table>

4.1.3 BlackWordList

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Serial</td>
<td>Primary Key</td>
</tr>
<tr>
<td>word</td>
<td>VARCHAR(20)</td>
<td>Not Null</td>
</tr>
<tr>
<td>isActive</td>
<td>boolean</td>
<td></td>
</tr>
</tbody>
</table>

4.1.4 BlackWordGroup

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Serial</td>
<td>Primary Key</td>
</tr>
<tr>
<td>wordGroup</td>
<td>VARCHAR(20)</td>
<td>Not Null</td>
</tr>
<tr>
<td>isActive</td>
<td>boolean</td>
<td></td>
</tr>
</tbody>
</table>
### 4.1.5 WhiteWordList

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Serial</td>
<td>Primary Key</td>
</tr>
<tr>
<td>word</td>
<td>VARCHAR(20)</td>
<td></td>
</tr>
<tr>
<td>isActive</td>
<td>boolean</td>
<td></td>
</tr>
</tbody>
</table>

### 4.1.6 WordInGroup

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>wordID</td>
<td>Integer</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( BlackWordList(ID) )</td>
</tr>
<tr>
<td>wordGroupID</td>
<td>Integer</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( BlackWordGroup(ID) )</td>
</tr>
</tbody>
</table>

### 4.1.7 ConfidentialData

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Serial</td>
<td>Primary Key</td>
</tr>
<tr>
<td>data</td>
<td>VARCHAR(30)</td>
<td></td>
</tr>
</tbody>
</table>

### 4.1.8 BlackURLList

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Serial</td>
<td>Primary Key</td>
</tr>
<tr>
<td>URL</td>
<td>VARCHAR(150)</td>
<td>Not Null</td>
</tr>
<tr>
<td>isActive</td>
<td>boolean</td>
<td></td>
</tr>
</tbody>
</table>

### 4.1.9 BlackURLGroup

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Serial</td>
<td>Primary Key</td>
</tr>
<tr>
<td>URLGroup</td>
<td>VARCHAR(20)</td>
<td>Not Null</td>
</tr>
<tr>
<td>isActive</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 4.1.10 URLInGroup

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>URLID</td>
<td>Integer</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( BlackURLList(ID) )</td>
</tr>
<tr>
<td>URLGroupID</td>
<td>Integer</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( BlackURLGoup(ID) )</td>
</tr>
</tbody>
</table>

### 4.1.11 WhiteURLList

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Serial</td>
<td>Primary Key</td>
</tr>
<tr>
<td>URL</td>
<td>VARCHAR(150)</td>
<td></td>
</tr>
<tr>
<td>isActive</td>
<td>boolean</td>
<td></td>
</tr>
</tbody>
</table>

### 4.1.12 Administrator

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>userName</td>
<td>VARCHAR(30)</td>
<td>Primary Key</td>
</tr>
<tr>
<td>password</td>
<td>VARCHAR(20)</td>
<td></td>
</tr>
<tr>
<td>IP</td>
<td>Char(15)</td>
<td></td>
</tr>
<tr>
<td>fullName</td>
<td>VARCHAR(30)</td>
<td></td>
</tr>
<tr>
<td>email</td>
<td>VARCHAR(30)</td>
<td></td>
</tr>
<tr>
<td>GSM</td>
<td>Char(11)</td>
<td></td>
</tr>
</tbody>
</table>

### 4.1.13 Permissions

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Integer</td>
<td>Primary Key</td>
</tr>
<tr>
<td>type</td>
<td>Char(20)</td>
<td></td>
</tr>
</tbody>
</table>

### 4.1.14 HavePermissions

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>userName</td>
<td>Integer</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( Administrator(username) )</td>
</tr>
<tr>
<td>permissionID</td>
<td>Char(20)</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( Permissions(ID) )</td>
</tr>
</tbody>
</table>
### 4.1.15 LocalUserGroup

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Serial</td>
<td>Primary Key</td>
</tr>
<tr>
<td>groupName</td>
<td>VARCHAR(20)</td>
<td></td>
</tr>
<tr>
<td>permittedDownloadSize</td>
<td>Float</td>
<td></td>
</tr>
</tbody>
</table>

### 4.1.16 RestrictedURLforUserGroup

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>userGroupID</td>
<td>Integer</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( LocalUserGroup(ID) )</td>
</tr>
<tr>
<td>blackURLGroupID</td>
<td>Integer</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( BlackURLGroup(ID) )</td>
</tr>
</tbody>
</table>

### 4.1.17 RestrictedWordforUserGroup

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>userGroupID</td>
<td>Integer</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( LocalUserGroup(ID) )</td>
</tr>
<tr>
<td>blackWordGroupID</td>
<td>Integer</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( BlackWordGroup(ID) )</td>
</tr>
</tbody>
</table>

### 4.1.18 UserInGroup

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>userIP</td>
<td>Char(15)</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( LocalUser(IP) )</td>
</tr>
<tr>
<td>userGroupID</td>
<td>Integer</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( LocalUserGroup(ID) )</td>
</tr>
</tbody>
</table>

### 4.1.19 ExtraURLforUser

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>userIP</td>
<td>Char(15)</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( LocalUser(IP) )</td>
</tr>
<tr>
<td>blackURLID</td>
<td>Integer</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( BlackURLList(ID) )</td>
</tr>
</tbody>
</table>
### 4.1.20 ExtraWordforUser

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>userIP</td>
<td>Char(15)</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(LocalUser(IP))</td>
</tr>
<tr>
<td>blackWordID</td>
<td>Integer</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(BlackWordList(ID))</td>
</tr>
</tbody>
</table>

### 4.1.21 ExtraURLGroupforUser

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>userIP</td>
<td>Char(15)</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(LocalUser(IP))</td>
</tr>
<tr>
<td>blackURLGroupID</td>
<td>Integer</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(BlackURLGroup(ID))</td>
</tr>
</tbody>
</table>

### 4.1.22 ExtraWordGroupforUser

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>userIP</td>
<td>Char(15)</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(LocalUser(IP))</td>
</tr>
<tr>
<td>blackWordGroupID</td>
<td>Integer</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(BlackWordGroup(ID))</td>
</tr>
</tbody>
</table>

### 4.1.23 ConfidentialDataViolations

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>userIP</td>
<td>Char(15)</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(LocalUser(IP))</td>
</tr>
<tr>
<td>violatedRuleID</td>
<td>Serial</td>
<td>Foreign Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(ConfidentialData(ID))</td>
</tr>
<tr>
<td>time</td>
<td>Timestamp</td>
<td></td>
</tr>
</tbody>
</table>

### 4.1.24 CachedURL

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>URLID</td>
<td>Integer</td>
<td>Primary Key</td>
</tr>
<tr>
<td>URL</td>
<td>VARCHAR(150)</td>
<td></td>
</tr>
</tbody>
</table>
### 4.1.25 RunningMode

<table>
<thead>
<tr>
<th>Name</th>
<th>Content Description</th>
<th>Supplementary Info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>modeID</td>
<td>Integer</td>
<td>Primary Key</td>
</tr>
<tr>
<td>isActive</td>
<td>boolean</td>
<td></td>
</tr>
</tbody>
</table>
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:

```sql
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(
    userID char(15),
    blackURLId integer,
    primary key(userID,blackURLId),
    foreign key (userID) references LocalUser(IP) on delete cascade on update cascade,
    foreign key (blackURLId) references blackURLList(ID) on delete cascade
);

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

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

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

create table ConfidentialDataViolations(
    userID Char(15),
    violatedRuleID serial,
    time timestamp,
    primary key(userID,violatedRuleID),
    foreign key (userID) 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;

```c
/*---------------------------------------
/* 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;

```c
<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.
<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NetCheck Project</td>
<td>22.01.2006</td>
<td>25.05.2006</td>
</tr>
<tr>
<td>2</td>
<td>NetCheck Project Proposal</td>
<td>22.02.2006</td>
<td>30.09.2006</td>
</tr>
<tr>
<td>3</td>
<td>Analysis</td>
<td>05.10.2005</td>
<td>04.11.2005</td>
</tr>
<tr>
<td>4</td>
<td>Requirement Analysis</td>
<td>05.10.2005</td>
<td>24.10.2005</td>
</tr>
<tr>
<td>5</td>
<td>Literature Survey</td>
<td>03.10.2005</td>
<td>17.10.2005</td>
</tr>
<tr>
<td>6</td>
<td>Meeting with Customers</td>
<td>17.10.2005</td>
<td>24.10.2005</td>
</tr>
<tr>
<td>7</td>
<td>Analyzer Report Writing</td>
<td>24.10.2005</td>
<td>04.11.2005</td>
</tr>
<tr>
<td>8</td>
<td>Risk Analysis</td>
<td>17.10.2005</td>
<td>24.10.2005</td>
</tr>
<tr>
<td>9</td>
<td>Project Scheduling and Training</td>
<td>24.10.2005</td>
<td>24.10.2005</td>
</tr>
<tr>
<td>10</td>
<td>Project Quality Plan</td>
<td>17.10.2005</td>
<td>17.10.2005</td>
</tr>
<tr>
<td>11</td>
<td>Milestone</td>
<td>04.11.2005</td>
<td>04.11.2005</td>
</tr>
<tr>
<td>12</td>
<td>Gathering Background Information</td>
<td>16.10.2005</td>
<td>16.11.2005</td>
</tr>
<tr>
<td>16</td>
<td>Linux Networking</td>
<td>24.10.2005</td>
<td>15.11.2005</td>
</tr>
<tr>
<td>17</td>
<td>Web Programming</td>
<td>11.11.2005</td>
<td>15.11.2005</td>
</tr>
<tr>
<td>18</td>
<td>Database</td>
<td>14.11.2005</td>
<td>15.11.2005</td>
</tr>
<tr>
<td>19</td>
<td>Milestone</td>
<td>16.11.2005</td>
<td>16.11.2005</td>
</tr>
<tr>
<td>20</td>
<td>Design</td>
<td>07.11.2005</td>
<td>07.12.2005</td>
</tr>
<tr>
<td>22</td>
<td>Content Distribution Design</td>
<td>07.11.2005</td>
<td>07.09.2005</td>
</tr>
<tr>
<td>31</td>
<td>Milestone</td>
<td>02.12.2005</td>
<td>02.12.2005</td>
</tr>
<tr>
<td>33</td>
<td>Milestone</td>
<td>16.01.2006</td>
<td>10.01.2006</td>
</tr>
<tr>
<td>34</td>
<td>Prototype Development</td>
<td>05.12.2005</td>
<td>12.01.2006</td>
</tr>
<tr>
<td>35</td>
<td>Coding Prototype</td>
<td>07.12.2005</td>
<td>12.01.2006</td>
</tr>
<tr>
<td>36</td>
<td>Prototype Design</td>
<td>17.01.2006</td>
<td>17.01.2006</td>
</tr>
<tr>
<td>37</td>
<td>Implementation</td>
<td>11.01.2006</td>
<td>05.05.2006</td>
</tr>
<tr>
<td>ID</td>
<td>Task Name</td>
<td>Start</td>
<td>End</td>
</tr>
<tr>
<td>----</td>
<td>-----------------------------------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>38</td>
<td>Filtering Module Implementation</td>
<td>06.02.2006</td>
<td>03.03.2006</td>
</tr>
<tr>
<td>39</td>
<td>Monitoring Module Implementation</td>
<td>24.01.2006</td>
<td>14.02.2006</td>
</tr>
<tr>
<td>40</td>
<td>Security and Protection Module Implementation</td>
<td>02.02.2006</td>
<td>28.02.2006</td>
</tr>
<tr>
<td>41</td>
<td>Logging Module Implementation</td>
<td>02.02.2006</td>
<td>23.02.2006</td>
</tr>
<tr>
<td>42</td>
<td>Management Module Implementation</td>
<td>09.03.2006</td>
<td>15.04.2006</td>
</tr>
<tr>
<td>43</td>
<td>Web Page Interface Implementation</td>
<td>14.04.2006</td>
<td>02.06.2006</td>
</tr>
<tr>
<td>45</td>
<td>Content Filtering Module Implementation</td>
<td>10.01.2006</td>
<td>31.01.2006</td>
</tr>
<tr>
<td>46</td>
<td>Network Traffic Monitoring Module Implementation</td>
<td>07.02.2006</td>
<td>02.02.2006</td>
</tr>
<tr>
<td>47</td>
<td>Milestone</td>
<td>06.05.2006</td>
<td>04.06.2006</td>
</tr>
<tr>
<td>48</td>
<td>Testing</td>
<td>07.06.2005</td>
<td>18.06.2006</td>
</tr>
<tr>
<td>49</td>
<td>System Management Module Implementation</td>
<td>07.03.2006</td>
<td>04.05.2006</td>
</tr>
<tr>
<td>50</td>
<td>Integration Testing</td>
<td>20.04.2006</td>
<td>10.06.2006</td>
</tr>
<tr>
<td>51</td>
<td>Milestone</td>
<td>22.05.2006</td>
<td>22.05.2006</td>
</tr>
<tr>
<td>52</td>
<td>Project Finalization</td>
<td>12.05.2006</td>
<td>20.05.2006</td>
</tr>
<tr>
<td>53</td>
<td>Application Setup Development</td>
<td>10.05.2006</td>
<td>25.06.2006</td>
</tr>
<tr>
<td>54</td>
<td>User Manual Preparation</td>
<td>12.05.2006</td>
<td>18.05.2006</td>
</tr>
<tr>
<td>55</td>
<td>Milestone</td>
<td>20.05.2006</td>
<td>30.05.2006</td>
</tr>
</tbody>
</table>