**Voila Firewall**

# 

**Table of Contents**

**CHAPTER 1**

**INTRODUCTION**

**1.1** **Project Introduction** .................................................................................................... 3

**1.2** **Existing Examples/Solutions**................................................................................... 3

**1.3** **Business Scope**........................................................................................................... 3

**1.4** **Useful Tools and Technologies** ............................................................................... 3

**1.5** **Project Timeline**........................................................................................................... 4

**1.6** **Project Work Breakdown** ........................................................................................... 5

**1.7** **Description**.................................................................................................................... 5

**Chapter 2**........................................................................................................................................ 8

**2.1** **Functional Requirements** .......................................................................................... 8

**2.2** **Non-Functional Requirements** ................................................................................. 9

**2.3** **Selected Requirements** .............................................................................................. 9

**2.4** **System Use Case Modeling** ..................................................................................... 11

**2.5** **Use Case Description**............................................................................................... 12

**2.6** **System Sequence diagram** ..................................................................................... 21

**2.7** **Domain Model** ............................................................................................................. 30

**Chapter 3**...................................................................................................................................... 31

**3.1** **Layer Definition** .......................................................................................................... 31

**3.2** **Layer Diagram** ............................................................................................................ 33

**3.3** **Class Diagram** ............................................................................................................ 34

**3.4** **Sequence Diagram** .................................................................................................... 35

**3.5** **Entity Relationship Diagram**................................................................................... 42

**3.6** **Database Schema** ...................................................................................................... 43

**3.7** **User Interface Design**............................................................................................... 45

**Chapter 4**...................................................................................................................................... 46

**Software Development** .............................................................................................................46

**4.1.** **Coding Standards** ...................................................................................................... 46

**4.2.** **Development Environment** ..................................................................................... 47

*Capital University of Science and Technology, Islamabad*

**4.3.** **Software Description** ................................................................................................ 48

**Chapter 5**.....................................................................................................................................51

**Software Testing** .......................................................................................................................51

**5.1** **Testing Methodology** ................................................................................................ 51

**5.2** **Testing Environment** ................................................................................................ 51

**5.3** **Test Cases** ................................................................................................................... 51

**Chapter 6**.................................................................................................................................... 52

**6.1.** **Installation / Deployment Process Description** ................................................. 52

**Chapter 7**.....................................................................................................................................53

**Project Evaluation** ....................................................................................................................53

**7.1.** **Project Evaluation Report** ....................................................................................... 53

## List of Figures

[Figure 1.1: Project Timeline 4](file:///D:\FYP%20Semester%207th,8th\New%20folder\FYP_Ahsan\Voila%20Firewall.docx#_Toc14242904)

[Figure 2.1: Project Breakdown 5](#_Toc14242905)

[Figure 3.2: Use Case 11](#_Toc14242906)

[Figure 4.2: Add Policy 23](#_Toc14242907)

[Figure 5.2: Editing Policy Rule 24](#_Toc14242908)

[Figure 6.2: Delete Policy Rule 25](#_Toc14242909)

[Figure 7.2: Apply Policy Rule 26](#_Toc14242910)

[Figure 8.2: View Policy Rule 27](#_Toc14242911)

[Figure 9.2: Get Network Interfaces 28](#_Toc14242912)

[Figure 10.2: Open Network Interface 29](#_Toc14242913)

[Figure 11.2: Capture Packets 30](#_Toc14242914)

[Figure 12.2: Create Log 31](#_Toc14242915)

[Figure 13.2: Read Dump File 32](file:///D:\FYP%20Semester%207th,8th\New%20folder\FYP_Ahsan\Voila%20Firewall.docx#_Toc14242916)

[Figure 14.2: Parse Dump File 33](file:///D:\FYP%20Semester%207th,8th\New%20folder\FYP_Ahsan\Voila%20Firewall.docx#_Toc14242917)

[Figure 15.2: Domain Model 34](#_Toc14242918)

[Figure 16.3: Layer Diagram 37](#_Toc14242919)

[Figure 17.3: Class Diagram 38](#_Toc14242920)

[Figure 18.3: Add Policy 39](#_Toc14242921)

[Figure 19.3: Edit Policy 40](#_Toc14242922)

[Figure 20.3: Delete Policy 41](#_Toc14242923)

[Figure 21.3: View Policy Rule 42](#_Toc14242924)

[Figure 22.3: Capture Packets 43](#_Toc14242925)

[Figure 23.3: Get Network Interfaces 44](#_Toc14242926)

[Figure 24.3: Open Network Interface 45](file:///D:\FYP%20Semester%207th,8th\New%20folder\FYP_Ahsan\Voila%20Firewall.docx#_Toc14242927)

[Figure 25.3: Apply Policy 46](#_Toc14242928)

[Figure 26.3: Create Log 47](#_Toc14242929)

[Figure 27.3: Read Dump File 48](#_Toc14242930)

[Figure 28.3: Parse Dump File 49](#_Toc14242931)

[Figure 29.3: ERD 50](#_Toc14242932)

[Figure 30.3: Database Schema 50](file:///D:\FYP%20Semester%207th,8th\New%20folder\FYP_Ahsan\Voila%20Firewall.docx#_Toc14242933)

[Figure 31.3: User Interface Design 51](#_Toc14242934)

[Figure 32.3: User Interface Design 52](#_Toc14242935)

## List of Tables

[Table 1.1: Firewall Features 2](#_Toc14242961)

[Table 2.2: Functional Requirements 8](#_Toc14242962)

[Table 3.2: Non-Functional Requirement 9](#_Toc14242963)

[Table 4.2: Selected Requirements 10](#_Toc14242964)

[Table 5.2: View Policy Rule 12](#_Toc14242965)

[Table 6.2: Add Policy 13](#_Toc14242966)

[Table 7.2: Edit Policy 14](#_Toc14242967)

[Table 8.2: Delete Policy 15](#_Toc14242968)

[Table 9.2: Get Network Interface 16](#_Toc14242969)

[Table 10.2: Apply Policy 17](#_Toc14242970)

[Table 11.2: Open Interface 18](#_Toc14242971)

[Table 12.2: Capture Packets 19](#_Toc14242972)

[Table 13.2: Create Log 20](#_Toc14242973)

[Table 14.2: Read Dump File 21](#_Toc14242974)

[Table 15.2: Parse Dump File 22](#_Toc14242975)

[Table 16.3: Layer Definition 35](#_Toc14242976)

[Table 17.5: Test Case 59](#_Toc14242977)

[Table 18.5:Test Case 59](#_Toc14242978)

[Table 19.5: Test Case 60](#_Toc14242979)

[Table 20.5: Test Case 62](#_Toc14242980)

[Table 21.5: Test Case 63](#_Toc14242981)

[Table 22.5: Test Case 64](#_Toc14242982)

**Chapter 1**

## Introduction

### 1.1 Project Introduction

If a PC is connected to the Internet, it is a potential target to an array of cyber threats and attacks, such as hackers, key loggers, and Trojans. So, to protect a system from these cyber-attacks and threats, firewalls are used.

A firewall works as a barrier, or a shield, between a PC and cyber space. When a computer is connected to the Internet, it is constantly sending and receiving information in small units called packets. The firewall filters these packets to see if they meet certain criteria set by a series of rules and then blocks or allows the data. This way, hackers cannot get inside and steal information such as bank account numbers and passwords from a computer.

Voila firewall is a first-generation firewall. First generation firewall is also referred to as Packet Filtering Firewall. Packet filtering firewall filters the incoming and outgoing packets to and from the network. Voila firewall will capture a packet, decode it, filter it and then allow or halt according to default rule set or defined by end-user. It will be a backdoor free system.

### 1.2 Existing Examples/Solutions

There are some firewalls that provide similar functionality to the Voila Firewall but not all of them provide the complete features. Unlike other highly expensive firewalls, Voila Firewall will be much less costly. Few are few firewalls and their features which are compared to the proposed system:

Table .1: Firewall Features

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Features/Firewalls** | **Zorp** | **Windows**  **7** | **WinGate** | **Untangle** | **Vyatta** |
| **Changing default policy to**  **accept/reject (by issuing a single rule)** | Yes | Yes | Yes | Yes | Yes |
| **IP destination address(es)** | Yes | Yes | Yes | Yes | Yes |
| **IP source address(es)** | Yes | Yes | Yes | Yes | Yes |
| **TCP/UDP destination port(s)** | Yes | Yes | Yes | Yes | Yes |
| **TCP/UDP source**  **port(s)** | Yes | No | Yes | Yes | Yes |
| **Ethernet MAC destination address** | Yes | No | No | No | Yes |
| **Ethernet MAC source address** | No | Yes | No | No | No |
| **Inbound firewall (ingress)** | No | Yes | No | Yes | No |
| **Outbound firewall (egress)** | No | Yes | Yes | Yes | Yes |
| **Cost** | Paid | Included with  windows  XP SP2  and later | Free/Paid | Free/Paid | Free |
| **OS** | Linux | All  Windows  Versions | [Windows](https://en.wikipedia.org/wiki/Windows)based | [Linux](https://en.wikipedia.org/wiki/Linux)[/NanoBSD](https://en.wikipedia.org/wiki/NanoBSD)based appliance  firewall distribution | Debian GNU/Linux |

### 1.3 Business Scope

There are different firewalls available in market but not all of them are cheap or providing all the features in a free version. Mostly, firewalls are not windows based but a major number of end-users use windows operating system. There are also backdoors present in some firewalls [1] [2]. There is not a single vendor in Pakistan who has created a firewall [3]. Voila Firewall will be a custom firewall which will provide all the features of packet filtering. By using these features, end-user will be able to define rules of filtering, block any IP from entering or leaving the system, update or delete default rule set. As it will be a backdoor free firewall, it will be highly useful for general public and defense sector for protecting their systems.

### 1.4 Useful Tools and Technologies

* Voila Firewall System will be a windows-based desktop application
* Java Standard Edition for implementation of Voila Firewall
* Packet capturing libraries for capturing packets
* NetBeans IDE for development
* Iterative development process will be used
* MySQL DBMS to store the policies
* Packet Builder to build/generate custom packet

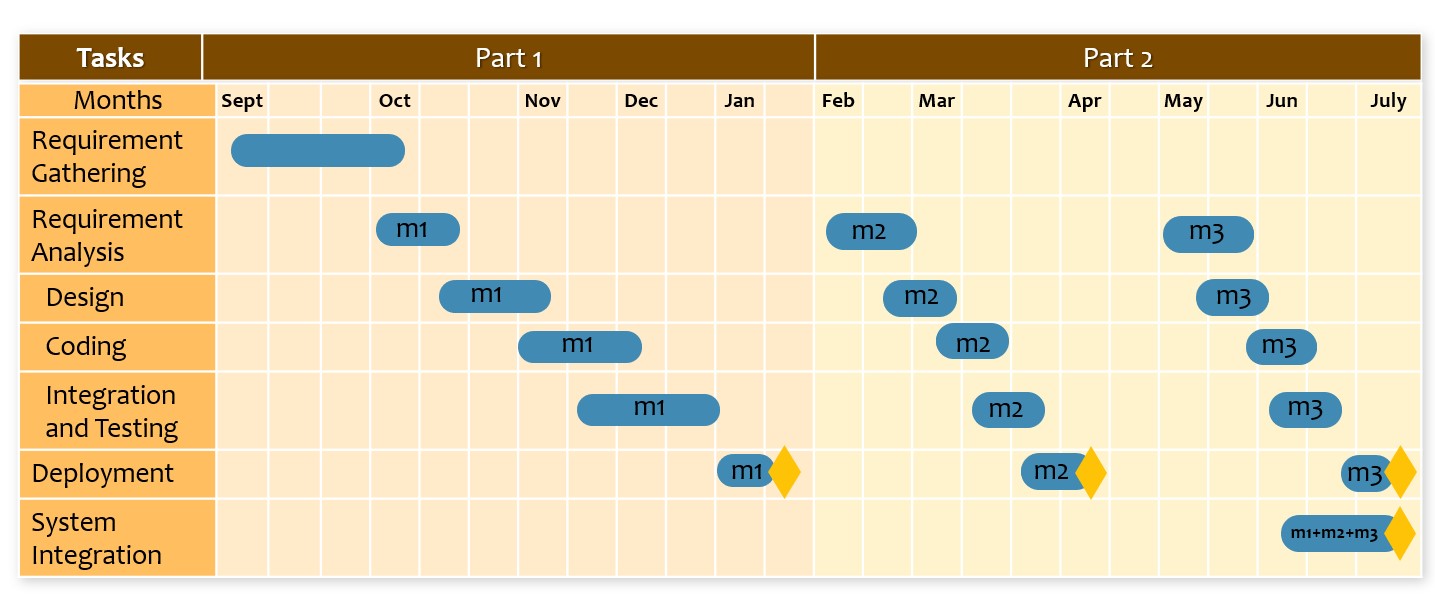
**1.5 Project Timeline**

Figure 1.1: Project Timeline

### 

### 1.6 Project Work Breakdown

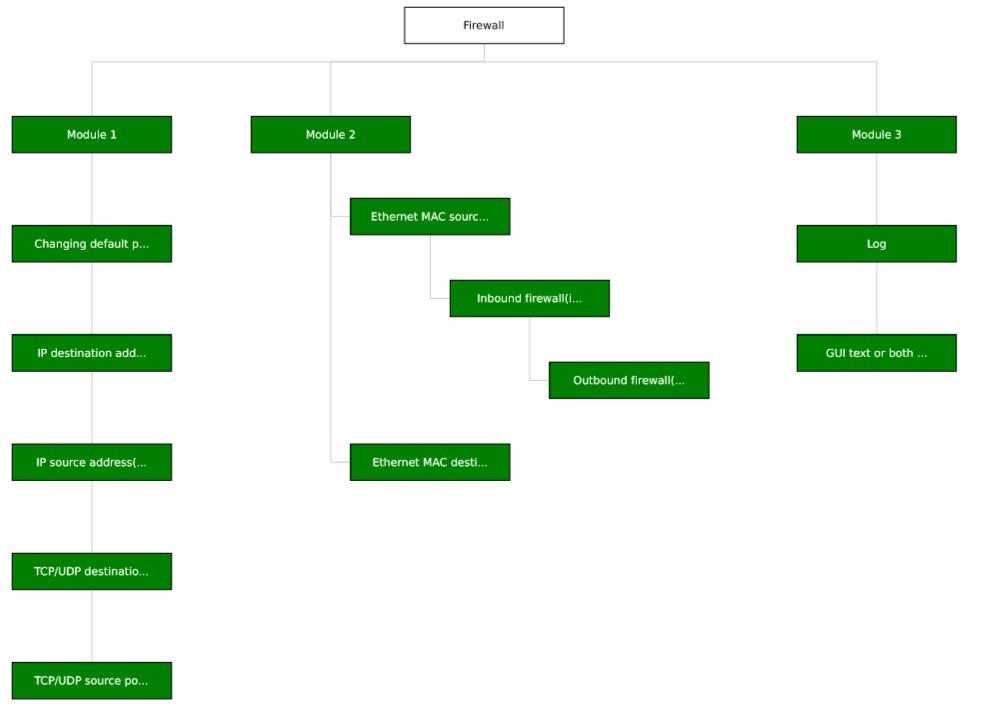


Figure .1: Project Breakdown

### 1.7 Description

**1.7.1 Policies**

**Changing default policy:**

By using Voila, end-user will be able to reject, block or accept any packet, by modifying the default rule set, from entering or leaving from or to the network.

* BLOCK - This feature drops the packet without sending response.
* REJECT - This has the same effect as 'BLOCK', in addition the remote host will get an ICMP error message.
* ACCEPT - It allows the packet to travel through the network.

**IP Destination Address (es):**

IP Destination Address is the destination/IP address of the remote/local host depending upon the network traffic (inbound/outbound) where the network packet is supposed to be transferred.

**IP Source Address (es):**

IP Source Address is the source/IP address of the remote/local host depending upon the network traffic (inbound/outbound) from where the network packet is supposed to be sent.

**TCP/UDP Destination Port(s):**

It is the port number of the remote host depending upon the TCP/UDP on which the data is sent.

**TCP/UDP Source Port(s):**

It is the port number of the local host depending upon the TCP/UDP from which the data is sent.

**1.7.2 Addresses**

**Ethernet MAC Destination Address:**

Ethernet MAC Destination Address is the address of the remote host which is used to identify the physical address of the remote host.

**Ethernet MAC Source Address:**

Ethernet MAC Source Address is the address of the local host which is used to identify the physical address of the local host.

**Inbound firewall (ingress):**

Ingress is a technique which is used to examine the incoming traffic from the network. It allows or denies the packet from entering the machine depending upon the information given in the header of that packet.

**1.7.3 Configuration**

Configuration of a firewall can be in Graphical User Interface mode or text mode. Some firewalls provide both modes which is more effective and efficient.

**Log:**

Firewalls have a logging feature which documents how the firewall handles various type of traffic. The logs can have this information:

* Source and destination IP address
* Port Numbers
* Protocols

## Chapter 2

**Requirements Specification and Analysis**

**2.1 Functional Requirements**

The functional requirements are listed in Table 1.

Table .2: Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.** | **Functional Requirement** | **Type** | **Status** |
| 1 | User can view default policy rules. | Intermediate | Completed |
| 2 | User can add new policy rules. | Intermediate | Completed |
| 3 | User can edit new policy rules. | Intermediate | Completed |
| 4 | User can delete new policy rules. | Intermediate | Completed |
| 5 | User can generate log file. | Intermediate | Completed |
| 6 | System can apply policy rules. | Core | Completed |
| 7 | System can get network interfaces. | Core | Completed |
| 8 | System can open interface. | Core | Completed |
| 9 | System can capture the packets. | Core | Completed |
| 10 | System can open packet. | Core | Completed |
| 11 | System can read the header. | Core | Completed |
| 12 | System can allow incoming packet from entering the machine upon checking header information. | Core | Completed |
| 13 | System can reject incoming packet from entering the machine upon checking header information. | Core | Completed |
| 14 | Read dump file | Core | Completed |
| 15 | Parse dump file | Core | Completed |
| 16 | Select IP ranges | Core | Completed |
| 17 | Select Port ranges | Core | Completed |
| 18 | Access control list | Core | Completed |

**2.2 Non-Functional Requirements**

Non-functional Requirements are given in Table 2.

Table .2: Non-Functional Requirement

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Non-Functional Requirement** | **Type** |
| 1.1 | Policies should be displayed in a list to the user | Intermediate |
| 3.1 | Policy rule details should be provided in a form to edit | Intermediate |
| 5.1 | Log file should be generated on user demand | Intermediate |
| 1.1 | Customize font size | Intermediate |
| 2.1 | Prevention of rule confliction | Intermediate |

**2.3 Selected Requirements**

The selected requirements set for this iteration is listed in Table 3.

Table .2: Selected Requirements

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Functional Requirement** | **Type** |
| 1 | Select IP ranges | Completed |
| 2 | Select Port ranges | Completed |
| 3 | Access control list | Completed |
| 4 | Prevention of rule confliction | Completed |

**2.4 System Use Case Modeling**

**2.4.1 Use Case Diagram**



Figure .2: Use Case

**2.5 Use Case Description**

The use case description for the above use case model is provided in the below section.

**View Policy Rules:**

Table .2: View Policy Rule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC01 | | | |
| **Use Case Name:** | View Policy Rules | | | |
| **Created By:** | Ahsan Zaheer | Last Updated By: | | Asim Ejaz |
| **Date Created:** | 10/30/2018 | Last Revision Date: | | 11/01/2018 |
| **Actors:** | User | | | |
| **Description:** | View default Policy rule use case will show the policy rules which are defined by default to the user. | | | |
| **Trigger:** | On Start | | | |
| **Preconditions:** | User must install the software. | | | |
| **Post conditions:** | User will be able to see all the default policy rules of the software. | | | |
| **Normal Flow:** | **User** | | **System** | |
| 1: User will start the program | | 2: System will fetch the policy rules from database and show them to the user | |
| **Alternative Flows:** | User terminates the program. | | | |
| **Exceptions:** | 1. Database is not responding. 2. System is not responding. | | | |

**Add Policy:**

Table .2: Add Policy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC02 | | | |
| **Use Case Name:** | Add Policy | | | |
| **Created By:** | Ahsan Zaheer | Last Updated By: | | Asim Ejaz |
| **Date Created:** | 10/30/2018 | Last Revision Date: | | 11/01/2018 |
| **Actors:** | User | | | |
| **Description:** | Add Policy Rule use case will let user define new policy rule  as per user’s requirement | | | |
| **Trigger:** | User clicks on Add Policy Button | | | |
| **Preconditions:** | Users starts the program | | | |
| **Post conditions:** | User can see the new policy rule added in the list of policy rules | | | |
| **Normal Flow:** | **User** | | **System** | |
|  | 1: User adds new policy | | 2: System will show form to add new policy rule | |
|  | 3: User will request the system to add new policy | | 4: System will facilitate the user and add new policy rule | |
| **Alternative Flows:** | 1: User provides invalid input | | 2: System asks the user to provide valid input | |
| **Exceptions:** | 1. Database is not responding. 2. System is not responding. | | | |

**Edit Policy:**

Table .2: Edit Policy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC03 | | | |
| **Use Case Name:** | Edit Policy | | | |
| **Created By:** | Ahsan Zaheer | Last Updated By: | | Asim Ejaz |
| **Date Created:** | 10/30/2018 | Last Revision Date: | | 11/01/2018 |
| **Actors:** | User | | | |
| **Description:** | User will edit policy rule which is defined by default or defined by user from the policy rule list. | | | |
| **Trigger:** | User clicks on Edit Policy Button | | | |
| **Preconditions:** | User starts the program | | | |
| **Post conditions:** | Policy rule will be updated and user can view the updated policy rule in the list which is displayed on main dashboard | | | |
| **Normal Flow:** | **User** | | **System** | |
|  | 1: User chooses the policy rule to edit    3: User edits the policy | | 2: System displays the details of chosen policy  4: System displays the success message | |
| **Alternative Flows:** | 1: User does not choose any policy | | 2: System displays the policy list | |
| **Exceptions:** | 1. Database is not responding. 2. System is not responding. | | | |

**Delete Policy:**

Table .2: Delete Policy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC04 | | | |
| **Use Case Name:** | Delete Policy | | | |
| **Created By:** | Ahsan Zaheer | Last Updated By: | | Asim Ejaz |
| **Date Created:** | 10/30/2018 | Last Revision Date: | | 11/01/2018 |
| **Actors:** | User | | | |
| **Description:** | User will choose policy to delete from the policy list which will be displayed as the user starts the program | | | |
| **Trigger:** | User clicks on delete button | | | |
| **Preconditions:** | User starts the program and chooses the policy to delete | | | |
| **Post conditions:** | Chosen policy will be deleted from the list of policy rules | | | |
| **Normal Flow:** | **User** | | **System** | |
|  | 1: Users chooses policy to delete | | 2: System displays the details of that policy | |
|  | 3: User deleted the policy | | 4: System removes the  deleted policy from the list and database | |
| **Alternative Flows:** | 1: User does not choose any policy | | 2: System displays the policy list already defined | |
| **Exceptions:** | 1. Database is not responding. 2. System is not responding. | | | |

**Get Network Interface:**

Table .2: Get Network Interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC05 | | | |
| **Use Case Name:** | Get Network Interface | | | |
| **Created By:** | Ahsan Zaheer | Last Updated By: | | Asim Ejaz |
| **Date Created:** | 10/30/2018 | Last Revision Date: | | 11/01/2018 |
| **Actors:** | System | | | |
| **Description:** | System will get all the network interfaces that are available in the machine. | | | |
| **Trigger:** | As the program starts, it will detect all the network interfaces automatically. | | | |
| **Preconditions:** | Network interfaces should be available in the machine. | | | |
| **Post conditions:** | System will provide the list of all the interfaces available in the machine. | | | |
| **Normal Flow:** | **System** | | **System** | |
| 1: System will detect the interfaces. | | 2: System provides the network interfaces | |
| **Alternative Flows:** | There is no network interface available in the machine | | | |
| **Exceptions:** | 1. System is unable to get any network interfaces. | | | |

**Apply Policy:**

Table .2: Apply Policy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC06 | | | |
| **Use Case Name:** | Apply Policy | | | |
| **Created By:** | Ahsan Zaheer | Last Updated By: | | Asim Ejaz |
| **Date Created:** | 10/30/2018 | Last Revision Date: | | 11/01/2018 |
| **Actors:** | System | | | |
| **Description:** | System will apply the policy rules on incoming and outgoing traffic which are defined by the user. | | | |
| **Trigger:** | When system start receiving packets. | | | |
| **Preconditions:** | Internet should be connected so that policy rules can be applied on captured packets added by user or must be defined by default for filtering the packets. | | | |
| **Post conditions:** | System will allow or block packets on the basis of defined policy rules. | | | |
| **Normal Flow:** | **System** | | **System** | |
|  | 1: Network packet arrives | | 2: System fetch the policy rules from database | |
|  | 3: System filters the packet according to policy rules | | 4: System will allow or block the packet after checking the  policy rules | |
| **Alternative Flows:** | 1: Internet connection is disabled | | 2: System shows the list of policy rules which are already defined | |
| **Exceptions:** | 1. System is not responding while getting the packets. 2. Database is not responding. | | | |

**Open Interface:**

Table .2: Open Interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC07 | | | |
| **Use Case Name:** | Open Interface | | | |
| **Created By:** | Ahsan Zaheer | Last Updated By: | | Asim Ejaz |
| **Date Created:** | 10/30/2018 | Last Revision Date: | | 11/01/2018 |
| **Actors:** | System | | | |
| **Description:** | System will detect the network interfaces which are available in the machine. System will open the network interface that is selected by the user for the first time. | | | |
| **Trigger:** | User will choose the interface on the start up | | | |
| **Preconditions:** | User must select a network interface before system can opening it | | | |
| **Post conditions:** | System will open the network interface chose by user. | | | |
| **Normal Flow:** | **System** | | **System** | |
|  | 1: System will show the list of network interfaces that are available in the machine. Then user will choose one of them. | | 2: System will open the device and display  the details of that interface. | |
| **Alternative Flows:** | System does not detect network interfaces | | | |
| **Exceptions:** | 1. System is not detecting any network interfaces. 2. System is unable to open any interface. | | | |

**Capture Packets:**

Table .2: Capture Packets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC08 | | | |
| **Use Case Name:** | Capture Packets | | | |
| **Created By:** | Ahsan Zaheer | Last Updated By: | | Asim Ejaz |
| **Date Created:** | 10/30/2018 | Last Revision Date: | | 11/01/2018 |
| **Actors:** | System | | | |
| **Description:** | System will capture incoming and outgoing packets | | | |
| **Trigger:** | Filter button should be enabled | | | |
| **Preconditions:** | Network interface must be opened before capturing. | | | |
| **Post conditions:** | System will capture all the packets successfully. | | | |
| **Normal Flow:** | **System** | | **System** | |
| 1: System will open the device | | 2: System will start capturing packets. | |
| **Alternative Flows:** | Internet connection is disabled | | | |
| **Exceptions:** | 1. Internet Connection is disabled 2. System is not responding | | | |

**Create Log:**

Table .2: Create Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC09 | | | |
| **Use Case Name:** | Create Log | | | |
| **Created By:** | Ahsan Zaheer | Last Updated By: | | Ahsan Zaheer |
| **Date Created:** | 01/30/2019 | Last Revision Date: | | 01/30/2019 |
| **Actors:** | System | | | |
| **Description:** | System will write the packet into the file. | | | |
| **Trigger:** | As the system receives a packets. | | | |
| **Preconditions:** | Log file must exist. If not, then the system should create the log file. | | | |
| **Post conditions:** | System will close the file after writing. | | | |
| **Normal Flow:** | **System** | | **System** | |
| 1: User will select the checkbox. | | 2: System will detect if the checkbox is selected or not and then it stores the packet into the file. | |
| **Alternative Flows:** | Log file does not exist. | | | |
| **Exceptions:** | 1. System is unable to write any packets. | | | |

**Read Dump File:**

Table .2: Read Dump File

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC10 | | | |
| **Use Case Name:** | Read Dump File | | | |
| **Created By:** | Ahsan Zaheer | Last Updated By: | | Ahsan Zaheer |
| **Date Created:** | 09/04/2019 | Last Revision Date: | | 12/04/2019 |
| **Actors:** | User | | | |
| **Description:** | System will read the dump file. | | | |
| **Trigger:** | After creating the dump file. | | | |
| **Preconditions:** | The dump file must be created before reading. | | | |
| **Post conditions:** | System will read the file and then parse it. | | | |
| **Normal Flow:** | **User** | | **System** | |
| 1: User will select the dump file for reading. | | 2: System will read the dump file. | |
| **Alternative Flows:** | Dump file does not exist. | | | |
| **Exceptions:** | 1. System is unable to create the dump file. | | | |

**Parse Dump File:**

Table .2: Parse Dump File

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC11 | | | |
| **Use Case Name:** | Parse Dump File | | | |
| **Created By:** | Ahsan Zaheer | Last Updated By: | | Ahsan Zaheer |
| **Date Created:** | 09/04/2019 | Last Revision Date: | | 12/04/2019 |
| **Actors:** | System | | | |
| **Description:** | System will parse the dump file. | | | |
| **Trigger:** | After reading the dump file. | | | |
| **Preconditions:** | The dump file must be read. | | | |
| **Post conditions:** | System will be able to parse the dump file. | | | |
| **Normal Flow:** | **System** | | **System** | |
| 1: System will read the dump file. | | 2: System will be able to parse the dump file. | |
| **Alternative Flows:** | System cannot read the dump file. | | | |
| **Exceptions:** | 1. System is unable to read the dump file. | | | |

**2.6 System Sequence diagram**

**2.6.1 Add Policy Rule**

**sd System Sequence Diagram**

User



:

System

Policy Added Successfully

addPolicyRule()

Figure .2: Add Policy

**Description:**

The SSD in figure 4.2 is about adding policy rule. User will be able to add policy rules in the system.

**2.6.2 Edit Policy Rule**

**sd System Sequence Diagram**

User



:

System

Policy Updated Successfully

editPolicyRule()

Figure .2: Editing Policy Rule

**Description:**

The SSD in figure 5.2 is about editing policy rule. User will be able to edit policy rules in the system.

**2.6.3 Delete Policy Rule**

**sd System Sequence Diagram**

User



:

System

Policy Deleted Successfully

deletePolicyRule()

Figure .2: Delete Policy Rule

**Description:**

The SSD in figure 6.2 is about deleting policy rule. User will be able to edit policy rules in the system.

**2.6.5 Apply Policy Rule**

**sd System Sequence Diagram**

System



:

System

Policy Applied Successfully

ApplyPolicyRule()

Figure .2: Apply Policy Rule

**Description:**

The SSD in figure 7.2 is about applying policy rules. System will apply all the policies on every packet.

**2.6.6 View Policy Rules**

**sd System Sequence Diagram**

User



:

System

Displaying Policy Rules

viewPolicyRules()

Figure .2: View Policy Rule

**Description:**

The SSD in figure 8.2 is about viewing default policy rule. In this user will be able to see all the default policy rule that is defined by the developers.

**2.6.7 Get Network Interfaces**

**sd Sequence**



System

:

System

Getting Network Interface Successfully

getNetworkInterface()

Figure .2: Get Network Interfaces

**Description:**

The SSD in figure 9.2 is about getting network interfaces. System will detect all the interfaces which are connected to the system.

**2.6.8 Open Network Interface**

**sd System Sequence Diagram**

System



:

System

Opened Successfully

openInterface()

Figure .2: Open Network Interface

**Description:**

The SSD in figure 10.2 is about opening interface. System will open any interface automatically.

**2.6.9 Capture Packets**

**sd System Sequence Diagram**

System



:

System

Captured Successfully

startCapturing()

Figure .2: Capture Packets

**Description:**

The SSD in figure 11.2 is about capturing packets. After opening the interface, the system will start capturing packets.

**2.6.10 Create Log**

**sd System Sequence Diagram**

System



:

System

Created Successfully

createLog()

Figure .2: Create Log

**Description:**

The SSD in figure 12.2 is about creating the log. After capturing the packets system will create a log file and write the packet in it.

**2.6.11 Read Dump File**

****

Figure 13.2: Read Dump File

**Description:**

The SSD in figure 13.2 is about reading the dump file. After capturing the packets system will create a log file and then read it.

**2.6.12 Parse Dump File**

****

Figure 14.2: Parse Dump File

**Description:**

The SSD in figure 14.2 is about parsing dump file. After capturing the packets system will create a dump file and then read it, and then parse that dump file.

**2.7 Domain Model**



Figure .2: Domain Model

**Description:**

* Firewall has one to many relationship with “Network Interface”. It means our firewall will have one or many network interfaces.
* Firewall has one to many relationship with Policy Rules. It means our firewall can have one or multiple policies in it.
* Network interface has one to many relationship with packet. It means network interfaces can have zero or multiple packets in it.

## Chapter 3

System Design

System design is a process that defines the components, features, modules and their interfaces and data for the system. It is used to design system that satisfies the requirements for business needs or an organization. System design can be a theory to product development. We are using top-down approach to develop the system design.

**3.1 Layer Definition**

The table 3.1 enlists the layers of the system.

Table .3: Layer Definition

|  |  |
| --- | --- |
| **Layers** | **Description** |
| Presentation Layer | This layer will be used for the interaction with the user through a graphical user interface. |
| Business Logic Layer | This layer contains the business logic. All the constraints and majority of the functions reside under this layer. |
| Database Layer | This layer contains the database of the application being developed. |

**3.1.1 Presentation Layer:**

This layer occupies the top level and displays information related to packets. Packet information is classified and then displayed to user in the GUI. This tier communicates with other tiers by receiving results from other components and displaying it and sending information to other tiers in the application.

**3.1.2 Business Logic Layer:**

Application Layer also called the middle tier, logic tier, business logic or logic tier, this tier is pulled from the presentation tier. The major functionality of the system falls under this layer. Selecting adapters to capturing packets and opening headers is responsibility of this layer. It gets rules from the presentation layer and allows packets or rejects them. After processing the header of packet, it sends data to presentation layer to be displayed to the user.

**3.1.3 Database Layer:**

In this layer database server is used to store and retrieve information. Database server used in the system is MySQL. Allowed IP addresses in whitelist and blocked IP addresses in black list are stored in the database. The database server is independent of the application.

**3.2 Layer Diagram**

**pkg sf arch**



**Presentation Layer**



**GUI**



**Bussiness Layer**



**Firewall**



**DataBase Layer**



**DataBase**



**External Service**

«use»

«use»

«use»

Figure .3: Layer Diagram

**3.3 Class Diagram**

The class diagram describes the attributes and operations of a class and the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.



Figure .3: Class Diagram

**3.4 Sequence Diagram**

Sequence Diagram model the flow of logic within your system in a visual manner enabling you both to document and validate your logic and are commonly used for both analysis and design purposes.

**Add Policy:**

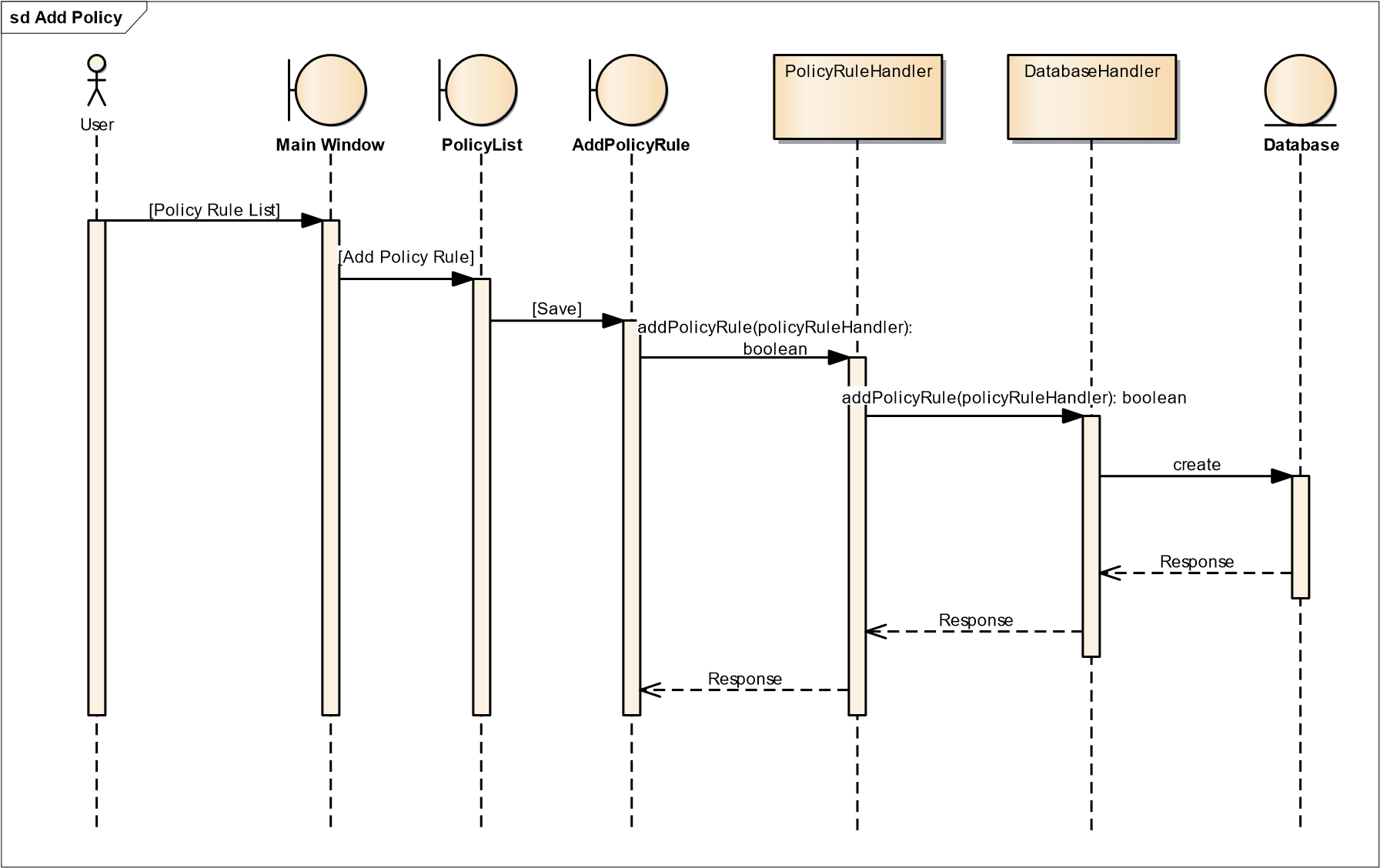


Figure .3: Add Policy

**Description:**

The SD in figure 18.3 is about adding policy rules. Users will be able to add policy rules in the system.

**Edit Policy:**

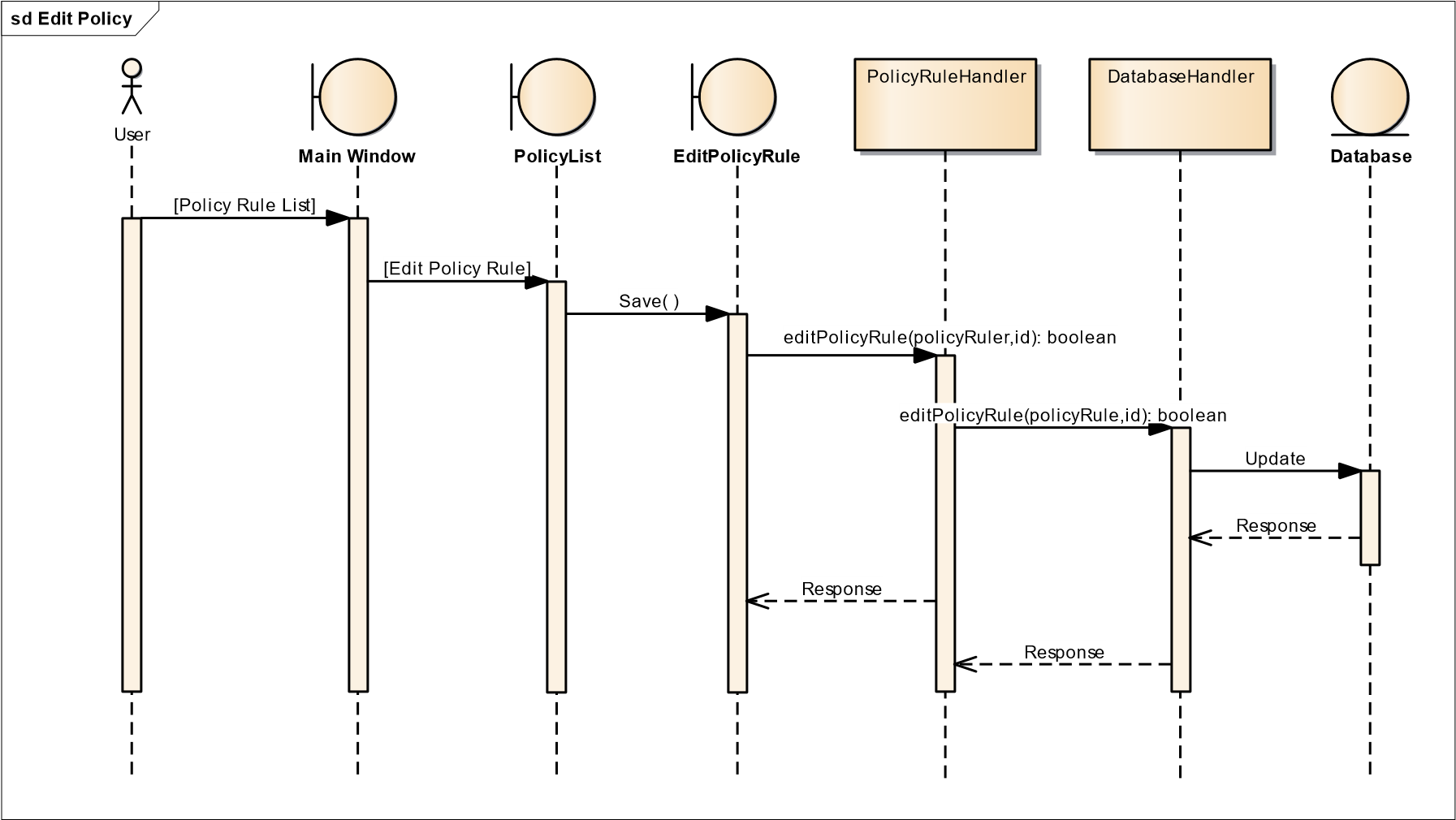


Figure .3: Edit Policy

**Description:**

The SD in figure 19.3 is about editing policy rules. Users will be able to edit policy rules in the system.

**Delete Policy:**

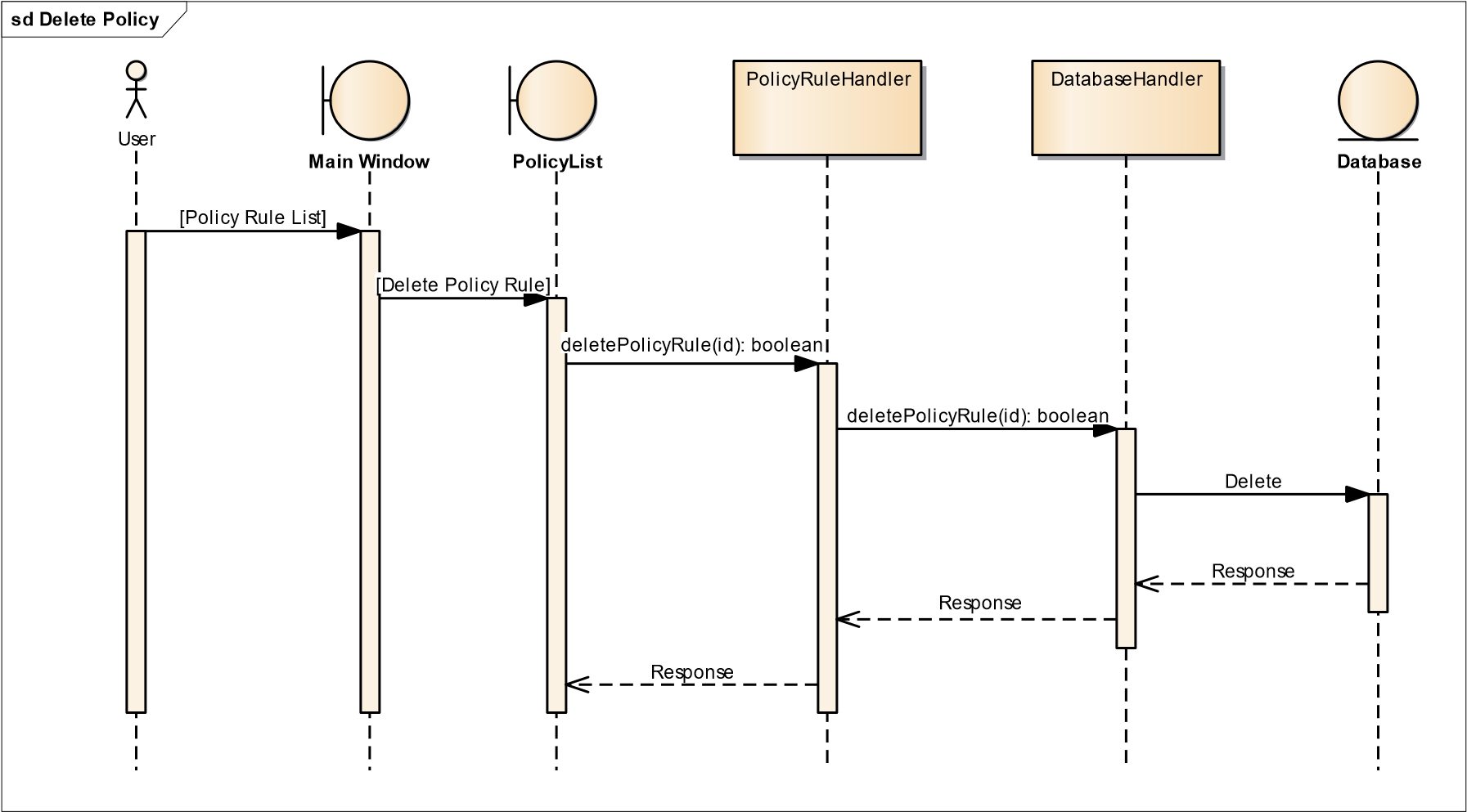


Figure .3: Delete Policy

**Description:**

The SD in figure 20.3 is about delete policy rules. Users will be able to delete policy rules in the system.

**View PolicyRule:**



Figure .3: View Policy Rule

**Description:**

The SD in figure 21.3 is about viewing policy rules. Users will be able to view all the policies which they created.

**Capture Packets:**

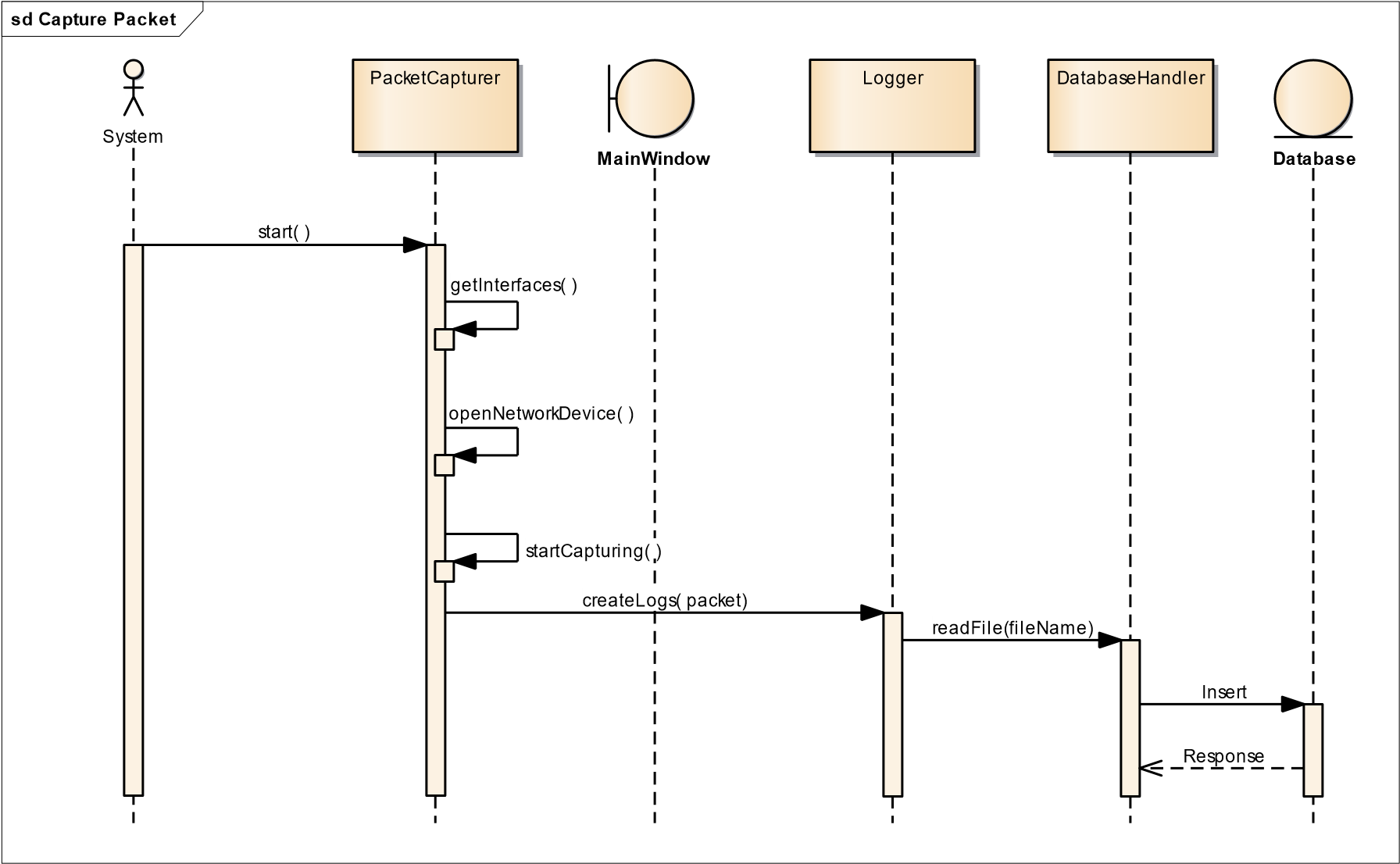


Figure .3: Capture Packets

**Description:**

The SD in figure 22.3 is about capturing packets. After opening the interface, the system will start capturing packets.

**Get Network Interfaces:**



Figure .3: Get Network Interfaces

**Description:**

The SD in figure 23.3 is about getting network interfaces. System will detect all the interfaces which are connected to the system.

**Open Network Interface:**



Figure 24.3: Open Network Interface

**Description:**

The SD in figure 24.3 is about opening interface. System will open any interface automatically.

**Apply Policy:**

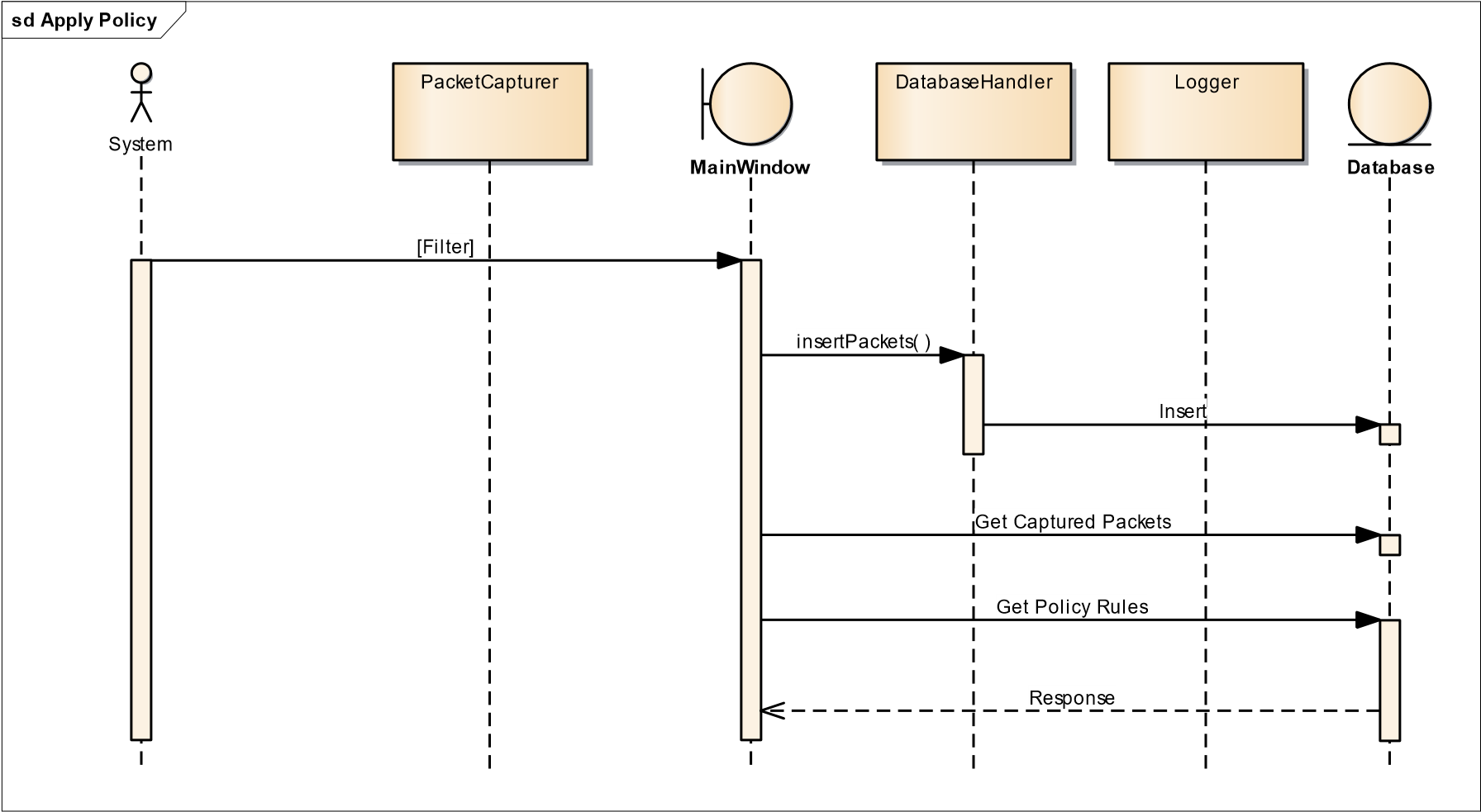


Figure .3: Apply Policy

**Description:**

The SD in figure 25.3 is about applying policy rules. System will apply all the policies on every packet which are added by user.

**Create Log:**



Figure .3: Create Log

**Description:**

The SD in figure 26.3 is about creating log file. System will create a log file and then write the packets in it.

**Read Dump File:**



Figure .3: Read Dump File

**Description:**

The SD in figure 27.3 is about reading dump file. System will read the file and then parse it.

**Parse Dump File:**



Figure .3: Parse Dump File

**Description:**

The SD in figure 28.3 is about parsing dump file. System will read the file and then parse it.

**3.5 Entity Relationship Diagram**

Entity Relation Diagram is provided for the above system:



Figure .3: ERD

**3.6 Database Schema**

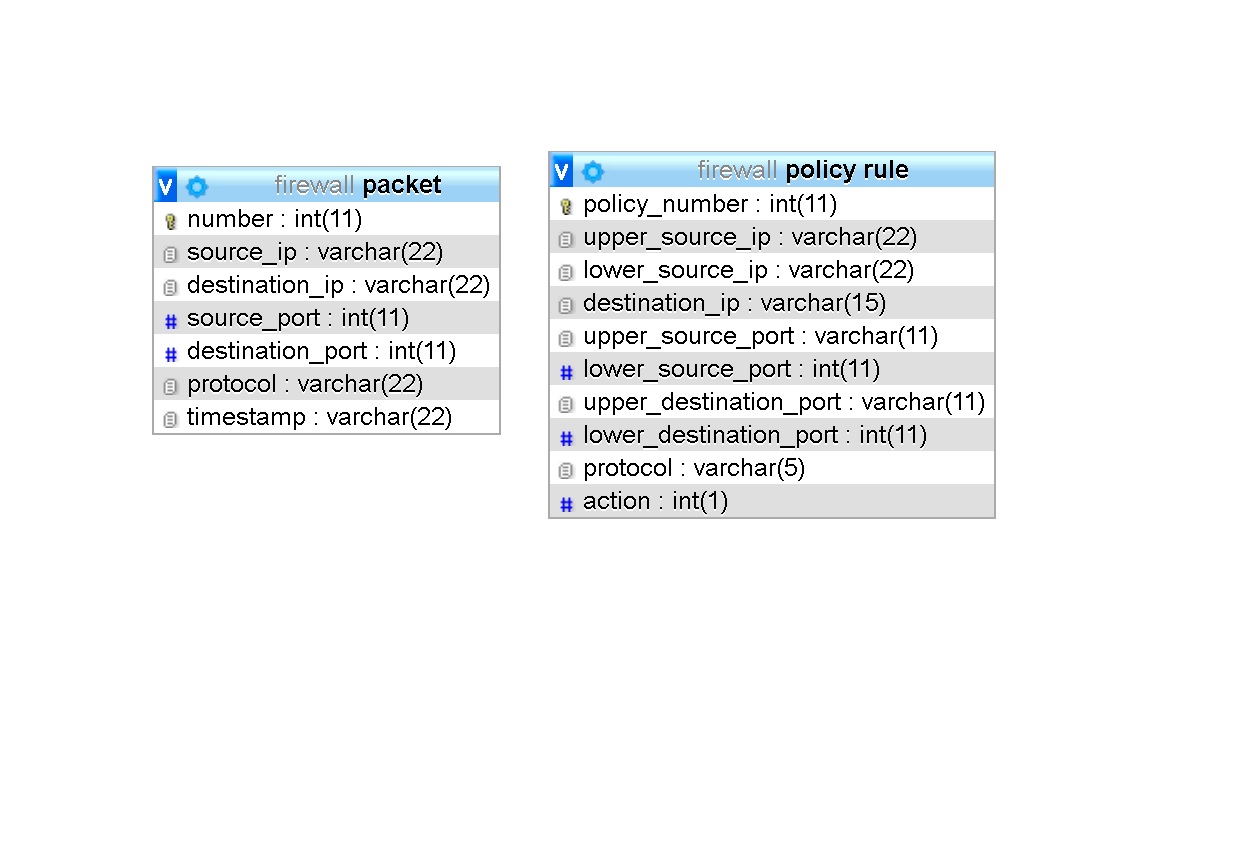
A database schema represents the logical configuration of all or part of a relational database. It can exist both as a visual representation and as a set of rules known as integrity constraints that govern a database. These rules are expressed in a data definition language, such as SQL.

Figure 30.3: Database Schema

**3.7 User Interface Design**

User interface design for the system is provided:

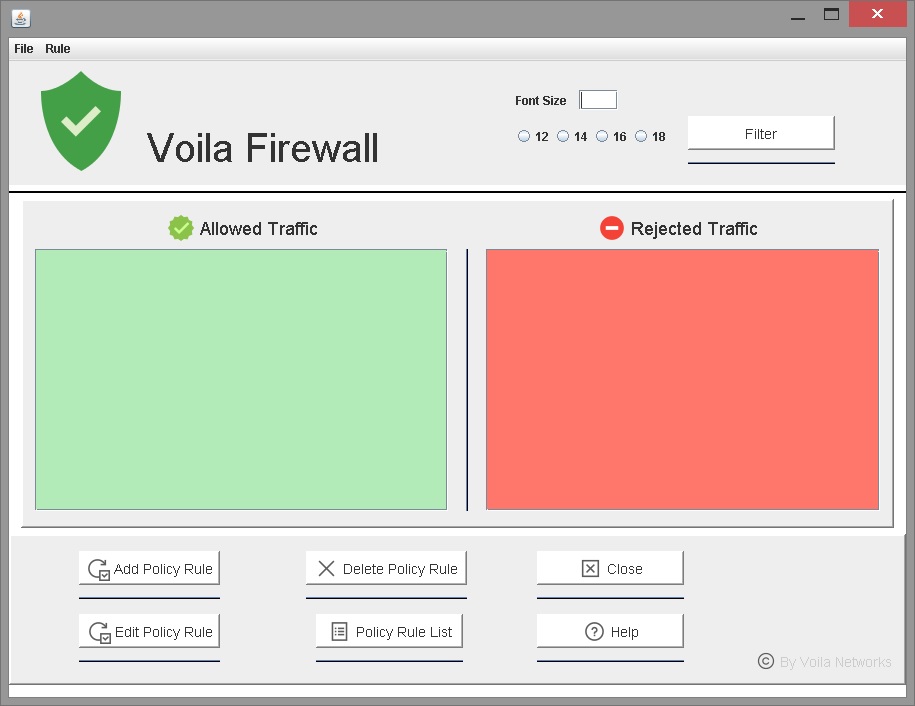


Figure .3: User Interface Design

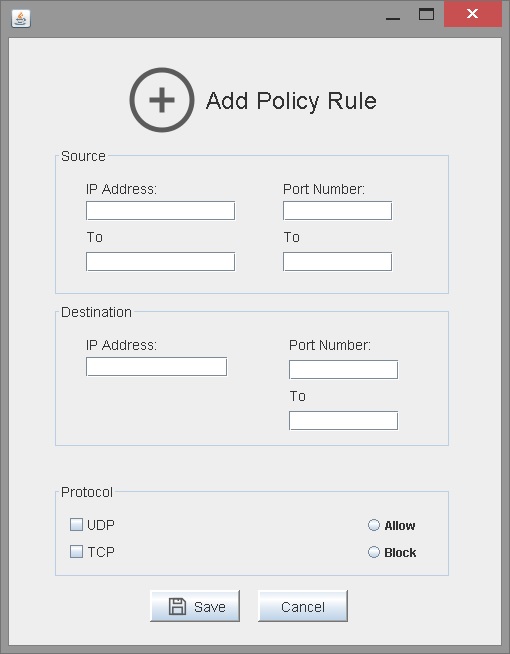


Figure .3: User Interface Design

## Chapter 4

### Software Development

We encountered many problems while developing our software. The main problem was lack of external libraries, finding suitable library was difficult. We also face difficulties involving existing software like its complexity, lack of documentation and lack of suitable supporting software. Common problems are:

* Integration with GUI
* Database Connectivity
* External libraries Integration

GUI integration issue was resolved by modifying our logic. Database connectivity issues were removed by changing the drivers. External libraries issue was resolved by setting configurations and class paths.

### 4.1. Coding Standards

**4.1.1. Indentation**

Eight spaces are used while indenting. The indentation pattern is consistently followed throughout.

**4.1.2. Declaration**

One declaration per line is mostly used in our code. The order and position of declaration is as follows:

* Main class is the only static class.
* Instance variables are placed in a consistent way because all the classes are public.
* package level with no access modifier.
* Class constructors are declared.
* Class methods are grouped by functionality rather than by scope or accessibility to make reading and understanding the code easier.
* Declarations for local variables are only at the beginning of blocks e.g. at the beginning of a try/catch construct.

**4.1.3. Statement Standards**

Most of the lines contains at most one statement. While compound statements are statements that contain lists of statements enclosed in braces. The enclosed statements are indented one more level than the normal statement. Braces are used around all statements, even single statements, when they are part of a control structure.

**4.1.4. Naming Convention**

Following conventions are followed while naming a class or a member:

* + We used camel case convention. For example, use of names like addPolicyRule, editPolicyRule, and dumpFileReader instead of names like a, b.
  + Terminology applicable to the domain is used.
  + Mixed case is used to make names readable with lower case letters in general capitalizing the first letter of class names and interface names.

### 4.2. Development Environment

We are using NetBeans (IDE) Version 8.2. The reason for using NetBeans is that it is easy to use and it is currently the best IDE for desktop application development. The designing of graphical user interface is very efficient and attractive.

### 4.3. Software Description

In this section we will provide our second module and the class diagram of this module as well. The subheadings of our module are:

* **Packet Capturing**

**Description**: This function utilizes Jpcap library to capture packet in Packet type class object. Then we have an infinite loop that captures the packet and print its data.

**Snippet 1**

public void startCapturing() throws IOException

{

Logger createlog = new Logger();

Packet packet = null;

String captured\_packet;

while (true) {

packet = captor.getPacket();

if (packet != null) {

captured\_packet = packet.toString(); char ch = captured\_packet.charAt(0);

if (ch != 'A') {

for(int i=0;i<1000;i++)

{

mw.allowedArea.setText("Hello"); mw.blockedArea.setText("Bye");

}

System.out.println(captured\_packet);

createlog.createlogs(packet);

}

* **Logging**

**Description**: This function gets packet from packet capturer class. It places the packet data in the file and waits for its call again.

#### Snippet 2

void createlogs(Packet packet) throws FileNotFoundException, IOException // creates logs

{

file = new File("test1.txt");

if (!file.exists()) {

try {

file.createNewFile();

System.out.println(" File created in the current directory");

} catch (IOException ex) {

java.util.logging.Logger.getLogger(Logger.class.getName()).log(Level.SEVERE, null, ex);

}

}

String data = packet.toString();

try {

bw = new BufferedWriter(new FileWriter(file, true));

bw.write(data);

bw.newLine();

}

catch (IOException ioe)

{

System.err.println("IOException: " + ioe.getMessage());

}

bw.close();

}

* **Implementing Policy Rules**

**Description**: This function gets policy Rule handler object and through that we can access the values of policy rule to add into database. This functions inserts the user policy into the database.

#### Snippet 3

public boolean addPolicyRule(PolicyRuleHandler policyRuleHandler) throws

SQLException, ClassNotFoundException {

String query = "INSERT INTO `policy rule`(`source\_ip`, `destination\_ip`,

`source\_port`, `destination\_port`, `action` ) VALUES(?,?,?,?,?)";

PreparedStatement ps = connection.prepareStatement(query);

//policyRuleHandler.policyRule.getDestinationIP(); ps.setString(1, "192.168.1.2"); ps.setString(2, "192.168.100.1");

ps.setInt(3, 5353); ps.setInt(4, 5353); ps.setInt(5, 1); int i = ps.executeUpdate(); if (i == 1) {

connection.close(); return true;

} else {

connection.close(); return false;

}

}

## Chapter 5

**Software Testing**

### 5.1 Testing Methodology

We used black box testing for our software. Eclipse IDE is used for generation of test cases.

### 5.2 Testing Environment

Netbeans IDE is used to test the test cases.

### 5.3 Test Cases

* **Test Case 1**
* Test case description

This test case is about user authentication. It checks if the password entered is correct or not.

* How test case was generated

We generated this test case by entering wrong password in the password field to check the output result.

* Expected result of the test case

Incorrect password

* Actual result of the test case

Incorrect password

Table .5: Test Case

|  |  |
| --- | --- |
| Date: 06 July 2019 |  |
| *System:* Voila Firewall |  |
| *Objective:* To test the user authentication | *Test ID:*1 |
| *Version:*1 | *Test Type:* Usability |
| *Input:*  Numeric Values  Instance: 123 |  |
| *Expected Result:* Incorrect Password |  |
| *Actual Result:* Incorrect Password |  |

* **Test Case 2**
* Test case description

This test case is about input validation of IP addresses.

* How test case was generated

We generated this test case by entering invalid format of IP address in source and destination IP’s fields.

* Expected result of the test case

Invalid IP address

* Actual result of the test case

Invalid IP address

Table .5:Test Case

|  |  |
| --- | --- |
| Date: 06 July 2019 |  |
| *System:* Voila Firewall |  |
| *Objective:* To validate IP addresses | *Test ID:*2 |
| *Version:*1 | *Test Type:* Functional |
| *Input:*  Alpha Numeric Values  Instance: Abc.192.168.10.1 |  |
| *Expected Result:* Invalid IP address |  |
| *Actual Result:* Invalid IP address |  |

* **Test Case 3**
* Test case description

This test case is about input validation of port numbers.

* How test case was generated

We generated this test case by entering invalid format of port numbers in source and destination port fields.

* Expected result of the test case

Invalid port number

* Actual result of the test case

Invalid port number

Table .5: Test Case

|  |  |
| --- | --- |
| Date: 06 July 2019 |  |
| *System:* Voila Firewall |  |
| *Objective:* To validate Port numbers | *Test ID:*3 |
| *Version:*1 | *Test Type:* Functional |
| *Input:*  *Alpha Numeric Values*  Instance: 20a1 |  |
| *Expected Result:* Invalid port number |  |
| *Actual Result:* Invalid port number |  |

* **Test Case 4**
* Test case description

This test case is about analyzing the policy which allows the packet enters the host system.

* How test case was generated

We generated this test case by entering IP address, port number and protocol, and action. We took packets from file to apply this policy rule.

* Expected result of the test case

Allowed

* Actual result of the test case

Allowed

Table .5: Test Case

|  |  |
| --- | --- |
| Date: 06 July 2019 |  |
| *System:* Voila Firewall |  |
| *Objective:* To examine the allow packet policy | *Test ID:*4 |
| *Version:*1 | *Test Type:* Functional |
| *Input:*  *Numeric Values*  Instance: 192.168.10.1 |  |
| *Expected Result:* Allowed |  |
| *Actual Result:* Allowed |  |

* **Test Case 5**
* Test case description

This test case is about analyzing the policy which blocks the packet enters the host system.

* How test case was generated

We generated this test case by entering IP address, port number and protocol, and action. We took packets from file to apply this policy rule.

* Expected result of the test case

Blocked

* Actual result of the test case

Blocked

Table .5: Test Case

|  |  |
| --- | --- |
| Date: 06 July 2019 |  |
| *System:* Voila Firewall |  |
| *Objective:* To examine the block packet policy | *Test ID:*5 |
| *Version:*1 | *Test Type:* Functional |
| *Input:*  *Numeric Values*  Instance: 192.168.10.2 |  |
| *Expected Result:* Blocked |  |
| *Actual Result:* Blocked |  |

* **Test Case 6**
* Test case description

This test case is about blocking the wildcard input of IP address and port number.

* How test case was generated

We generated this test case by entering 0.0.0.0 as IP address and 0 as port number.

* Expected result of the test case

Blocked

* Actual result of the test case

Blocked

Table .5: Test Case

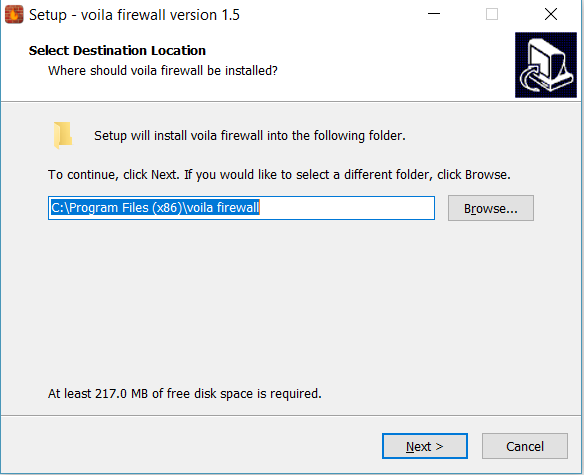
|  |  |
| --- | --- |
| Date: 06 July 2019 |  |
| *System:* Voila Firewall |  |
| *Objective:* To block all IP addresses and port numbers | *Test ID:*6 |
| *Version:*1 | *Test Type:* Functional |
| *Input:*  *Numeric Values*  Instance: 0.0.0.0 |  |
| *Expected Result:* Blocked |  |
| *Actual Result:* Blocked |  |

**Chapter 6**

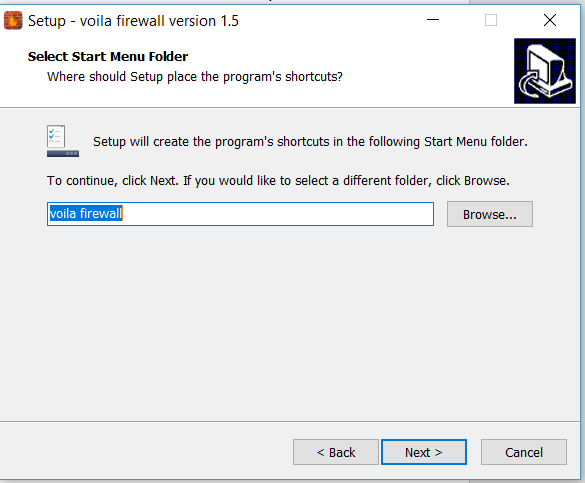
## Software Deployment

**6.1. Installation / Deployment Process Description**

We have a setup file of our software. User will simply start the setup, and a wizard will open as shown below:

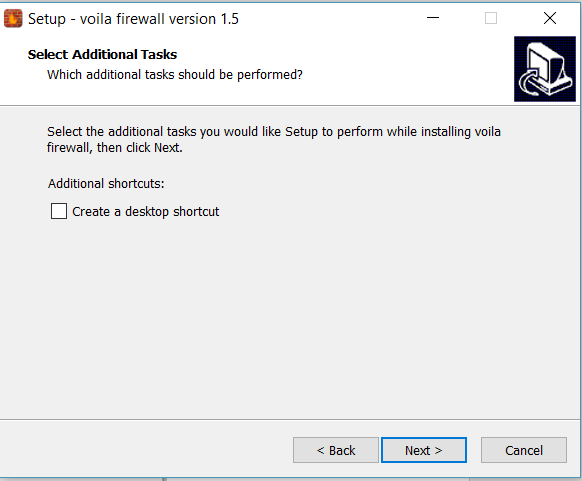


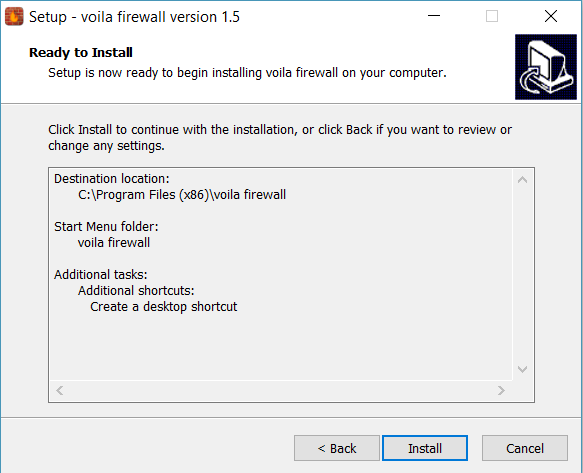
users can choose the path where they want to install this software. And press Next button



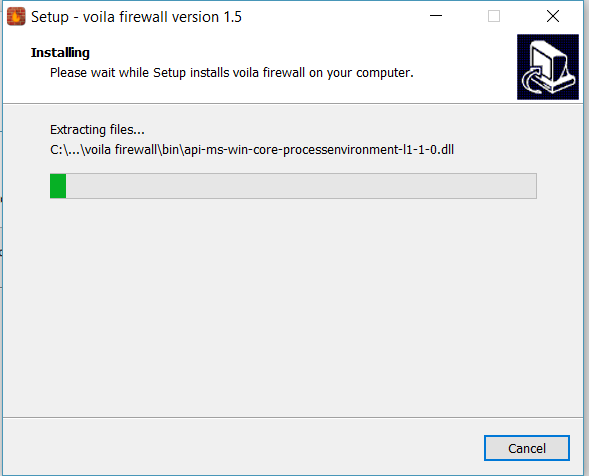
User can select where to create program’s shortcut then press Next button to continue the process

Then if you want to create desktop shortcut check the box as shown below:

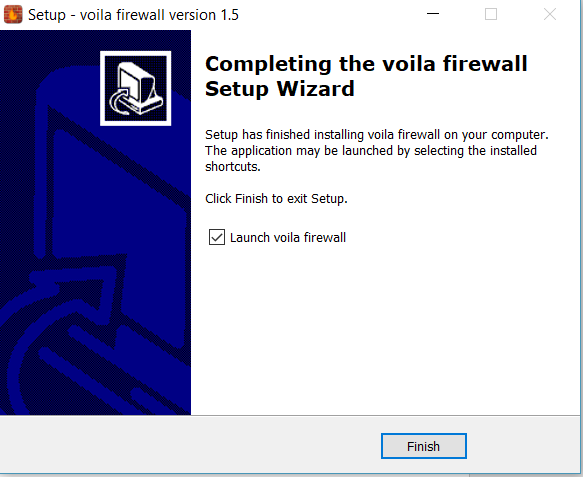




Now the program is ready to install, and all the configurations are displayed in text area once ready click the Install button to start the installing process



This will take couple of seconds to install and once it is done following window will open



Press the finish button to complete the setup and firewall will start.

## Chapter 7

### Project Evaluation

This chapter includes the examiners evaluation report, including the points to be revised/included along with the selected requirements in the next iteration.

### 7.1. Project Evaluation Report

|  |  |
| --- | --- |
| **Examiner Name:** | |
| **S. No.** | **Suggestion** |
| 1 | Entity relation should be labeled. |
| 2 | Sequence diagram should be split. |
| 3 | English sentence structure should be improved. |
| 4 | Formatting should be improved. |
| 5 | Naming conventions are not followed in code artifacts. |
| 6 | Indention rules are not followed as described in chapter 4 |
| 7 | Test cases for the incorrect behavior to be included. |
| 8 | Test cases should be added to test remaining functionalities. |

**Other Comments (If any):**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature**

### References

1. "NSA spied on Pakistani civil-military leadership," Sunday 08 2016. [Online]. Available: https://tribune.com.pk/story/1166854/nsa-spied-pakistani-civilmilitary-leadership/. [Accessed Saturday 09 2018].
2. "Snowden documents show NSA leak is real," Sunday 08 2016. [Online]. Available: https://tribune.com.pk/story/1166392/snowden-documents-shownsa-leak-real-report/. [Accessed Saturday 09 2018].
3. "List of firewall vendors," [Online]. Available: http://firewall.firm.in/tag/list-offirewall-vendors/. [Accessed Friday 09 2018].