

2015

Design Document

IPv6 Readiness of servers



IT08 Ltd.

Abdul Sami
Bilawal Mushtaq
Leslie Vundu
Man Fu Lei
Robert Smart

10/23/2015



Version

Version	Implemented By	Revision Date	Reason
1.0	Abdul Sami, Leslie Vundu & Man Fu	04/09/2015	Creation of Document and Formatting
1.1	Man Fu, Leslie Vundu & Abdul Sami	09/09/2015	Redo Document + Edit New Information
1.2	Man Fu	11/09/2015	Edit Document / Formatting
1.25	Robert Smart, Bilawal Mushtaq	13/09/15	Briefs under processes / Edit Document partially
1.3	Abdul Sami, Leslie Vundu	15/09/15	Edit document / error corrections, grammatical mistakes / formatting
2.0	Abdul Sami	22/10/15	Reformatting minor

Table of Contents

Version.....	i
Table of Contents.....	ii
Table of Figures.....	iv
Table of Tables.....	v
Executive summary.....	vi
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Scope.....	1
1.3 Out of scope.....	2
1.4 Description of Intended Audience.....	2
2. Data Design.....	3
2.1 Relational Entity Relationship Diagram.....	3
2.2 Data Structure & operations.....	3
2.3 Data-to-process CRUD Matrix.....	4
3. Process Design.....	7
3.1 Level-1 Data Flow Diagrams.....	7
3.1.1 Level-1.0 Server Probing.....	7
3.1.2 Level-2.0 Database Searching.....	8
3.1.3 Level-3.0 Data Representation.....	8
3.1.4 Level-4.0 Diagnosis.....	9
3.1.5 Level-5.0 Import & Export Database.....	10
3.1.6 Level-6.0 Tables & Charts Export.....	11
3.2 Level-2 Data Flow Diagram.....	12
3.2.1 Level-2 DFD for function 1.3 Probing using IPv4 address.....	12
3.3 Process Description.....	13
3.4 Expanded Use Case Narratives.....	16
3.4.1 Probing Servers.....	16
3.4.2 Searching Data.....	17
3.4.3 Representing Data.....	18
3.4.4 Import & Export Database.....	19
3.4.5 Tables & Charts Export.....	20
3.5 Processing Control & Algorithm.....	20
3.6 Structure Chart.....	22
3.6.1 Server Probing.....	22

3.6.2	Storing Data	23
3.6.3	Searching & Representing Data	24
3.6.4	Diagnosis.....	25
3.6.5	Import & Export Database	26
3.6.6	Tables & Charts Export	27
4.	Software Choices	27
5.	Architecture/Infrastructure	28
5.1	Infrastructure overview	28
5.2	Architectural Diagram	29
5.3	Infrastructural Requirements.....	30
5.4	Alternative Design.....	30
6.	Interface Design	31
6.1	Application Interface Design	31
6.2	User Interface Design.....	32
6.3	Interface Specifications	32
6.3.1	Input	32
6.3.2	Output	33
6.3.3	Formatting.....	34
7.	Conclusion	35
8.	Appendix.....	36
8.1	Appendix A: Deliverable Task Breakdown Statement.....	36
8.2	Appendix B: Data Dictionary.....	37
9.	List of References	40
10.	Glossary.....	41

Table of Figures

Figure 1 - Level-1.0 Server probing	7
Figure 2 - Level-2.0 Database searching	8
Figure 3 - Level-3.0 Data representation	8
Figure 4 - Level-4.0 Diagnosis	9
Figure 5 - Level-5.0 Import Database	10
Figure 6 - Level-5.0 Export Database	10
Figure 7 - Level-6.0 Tables Export	11
Figure 8 - Level-6.0 Charts Export	11
Figure 9 - Level-2 DFD for function 1.3	12
Figure 10 - Process description	15
Figure 11 - Structure diagram: server probing	22
Figure 12 - Structure diagram: storing data	23
Figure 13 - Structure diagram: searching & representing data	24
Figure 14 - Structure diagram: diagnosis	25
Figure 15 - Structure diagram: import & export database	26
Figure 16 - Structure diagram: tables & charts export	27
Figure 17 - Infrastructure overview diagram	28
Figure 18 - Architectural Diagram (Nordgren, 2010)	29
Figure 19 - Application interfaces	31
Figure 20 - User interface	32

Table of Tables

Table 1 - Data-to-process CRUD matrix.....	6
Table 2 - Expanded use case narrative: probing servers.....	16
Table 3 - Expanded use case narrative: Searching data	17
Table 4 - Expanded use case: representing data	18
Table 5 - Expanded use case narrative: import &export database	19
Table 6 - Expanded use case narrative: tables &charts export.....	20
Table 7 - Data dictionary.....	39

Executive summary

The main goal of this document is to provide an architectural overview of a proposed solution based on data provided from the requirements and analysis document. This document describes in detail how the IPv6 prototype software functions. Data design, process design, infrastructural designs and interface design will be well documented here. As to meet the requirements of the client the IPv6 software will measure the performance of internet servers by probing them with different kind of information. IPv6 will run concurrently with IPv4 and the data will be stored in the database. The results will show the different percentages of which country or continent is IPv6 ready. This design document details the reader on how the requirements should be implemented and also provides the programmer with a plan of what needs to be done. This document will help determine if all the necessary requirements have been taken into account. The data design defines the software's characteristic or shows the function involved. The data flow diagrams are used to interpret the function of the IPv6 software at various levels. A data dictionary is provided which will explain the definitions as well as all the data attributes inside the database. A Create Read Update Delete (CRUD) Matrix was used to identify the relationships between different data field and different processes. The process design describes each software component and allows us to view the choices that were made regarding the software selection and provides a brief insight into what features are needed to ensure that the IPv6 software provides the necessary results for comparison. Software choices allow us to expand on our selections and provide reasoning as to why our software choice was best for that particular scenario. Process models such as level 0, context Data Flow Diagram (DFD) and level 2 diagrams are provided. The Architecture/Infrastructure Design part of the document shows the final output in diagram form. The design shows the infrastructure that has been put together and how they will interact with other components. It shows the platform to be used and what capacity the IPv6 software can handle. The main goal of adding an interface design is to ensure that the software behaves exactly like how the programmer wants it to act. The admin would be required to logon to check for if it is functioning properly.

1. Introduction

1.1 Purpose

The purpose of this document is to provide an architectural overview of IPv6 readiness of different servers across different continents. The document describes in details the system as a whole and what we hope to achieve when probing servers for IPv6 readiness. The document will provide technical features of system components and proposes to convey major infrastructural design decisions which have been proposed in preparations for the development of the system.

This document is based on the Requirement and Analysis document version 3.1 submitted earlier, as the client provided the project requirements which were outlined to identify the business process needs for the proposed solution. After thorough research on the defined functional requirements, the document was developed to define methods that will allow the IPv6 software to do the following:

- Probing servers across the internet
- Processing the raw data and storing them into database
- Searching the database and graphically representing the data
- Diagnosis toward IPv6 unavailable networks
- Provide means for end users to use it
- Provide clear and concise technical documentation.

As a result of the Design Document we will be able to develop a test plan or a development plan which then specifies how, when and what services were tested.

1.2 Scope

The scope of this document is intended to be a detailed design of the proposed solution to IT08 Ltd. The document will provide the following:

- Data design
- Process design
- Architecture design
- User interface

1.3 Out of scope

This document will not provide designs or descriptions of the following:

- Source code of the program
- Any topics not discussed above from the table of contents

1.4 Description of Intended Audience

The intended audience of this document are the group members and all stakeholders for IT08 Ltd. This document is mainly used to provide a more in-depth look into the designing of the prototype software that needs to be completed for the client.

2. Data Design

This sections goal is to provide an overview of the flow of data involving the processes in the system.

2.1 Relational Entity Relationship Diagram

The following is the E-R diagram that has been normalized to 3rd degree:

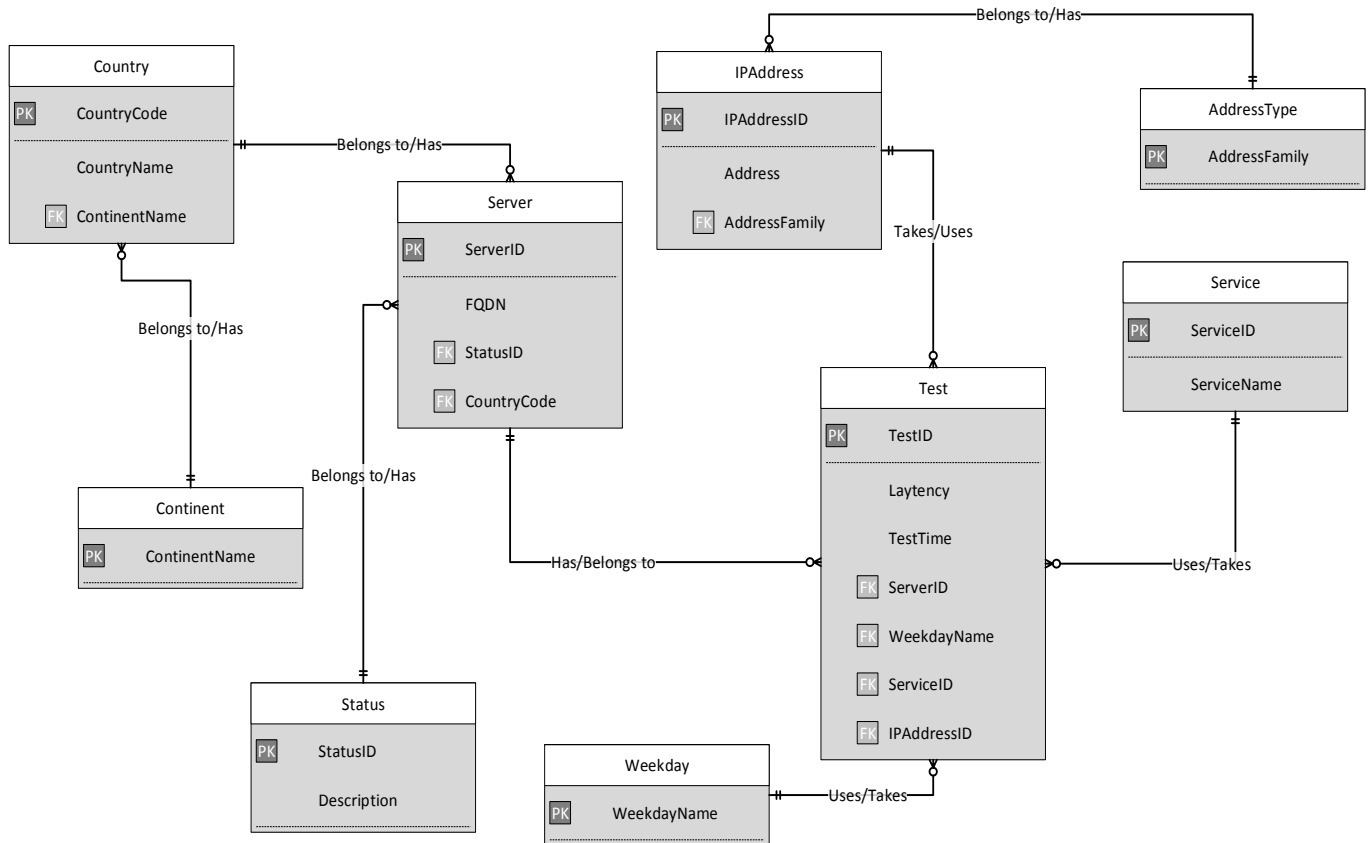


Figure 1 - Relational E-R Diagram

2.2 Data Structure & operations

All the data will be able to be read, created and updated. The deletion action is not implemented in this software.

The data dictionary is a document that explains the definition as well as all kinds of data attributes inside the database. These attributes are also called metadata. The data dictionary is in **Appendix B**.

2.3 Data-to-process CRUD Matrix

The matrix that identifies the relationships between different data fields and different processes is as follows:

Entity. Attribute	Initialization	Probing Servers	Storing Data	Searching Data	Data Representation	Server Diagnosis	Import & Export Database	Tables & Charts Export
Continent	C		U	R	R		C,R,U	R
.ContinentName	C		U	R	R		C,R,U	R
Country	C		U	R	R		C,R,U	R
.CountryCode	C		U	R	R		C,R,U	R
.CountryName	C		U	R	R		C,R,U	R
Server	C		U	R	R		C,R,U	R
.ServerID	C		U	R	R		C,R,U	R
.FQDN	C		U	R	R		C,R,U	R
IPAddress	C		U	R	R		C,R,U	R

.IPAddressID	C		U	R	R		C,R,U	R
.Address	C		U	R	R		C,R,U	R
AddressType	C		U	R	R		C,R,U	R
.AddressFamily	C		U	R	R		C,R,U	R
Service	C		U	R	R		C,R,U	R
.ServiceID	C		U	R	R		C,R,U	R
.ServiceName	C		U	R	R		C,R,U	R
Status	C		U	R	R	U	C,R,U	R
.StatusID	C		U	R	R	U	C,R,U	R
.Description	C		U	R	R	U	C,R,U	R
Test	C		U		R		C,R,U	R
.TestID	C		U		R		C,R,U	R
.Latency	C		U		R		C,R,U	R

.TestTime	C		U		R		C,R,U	R
Weekday	C		U		R		C,R,U	R
.WeekdayName	C		U		R		C,R,U	R
C=create, R=read, U=update, D=delete								

Table 1 - Data-to-process CRUD matrix

3. Process Design

This section includes the Data flow diagrams from level 1 onwards.

3.1 Level-1 Data Flow Diagrams

Level-1 DFD comes from level-0 diagram which doesn't contain the explicit symbols of the sources or sinks except where the data flow needs to be clarified. Level-1 DFDs have the numbering convention of X.Y (two digits). X indicates the function in level-0 diagram and Y numbers the processes in this diagram.

3.1.1 Level-1.0 Server Probing

The level-1 server probing DFD shown below:

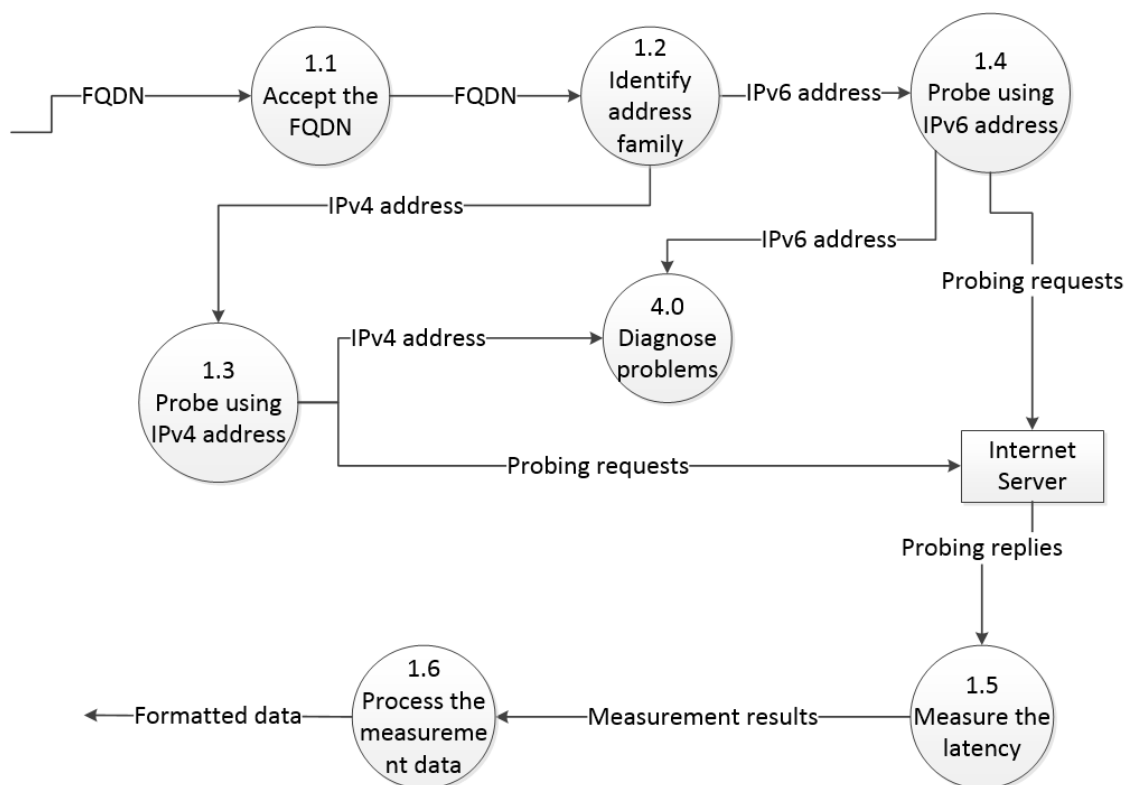


Figure 1 - Level-1.0 Server probing

The software accepts the Fully Qualified Domain Name (FQDN) and then identifies its address family. IPv4 and IPv6 addresses will be probed separately. If the server can't be reached, it will be passed onto the diagnosis function 4.0. The probing requests will be sent to the Internet server, the replies will be measured and the results will be stored into database after being processed as well as formatted.

3.1.2 Level-2.0 Database Searching

The level-2.0 database searching diagram is like following:

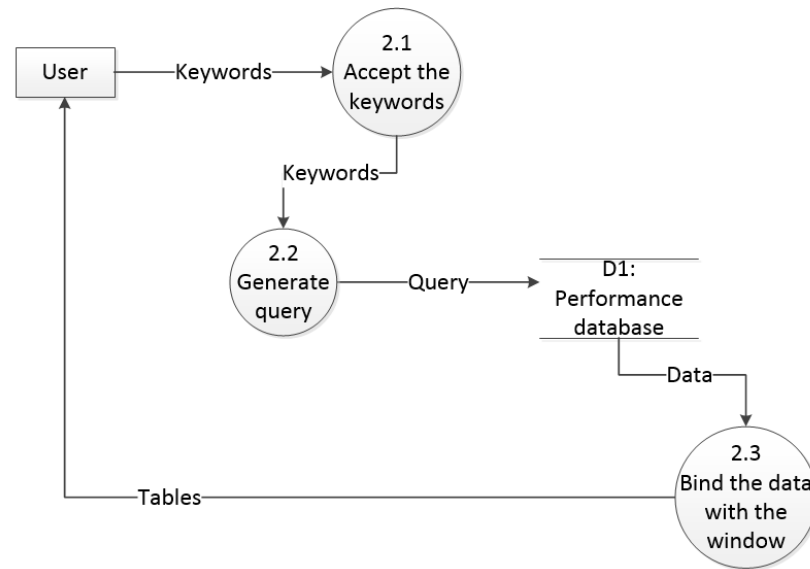


Figure 2 - Level-2.0 Database searching

The user inputs the keywords to search the database. The software generates queries which then enquire the database for related information and bind the data to the window that will display the table.

3.1.3 Level-3.0 Data Representation

The level-3.0 data representation diagram is as follows:

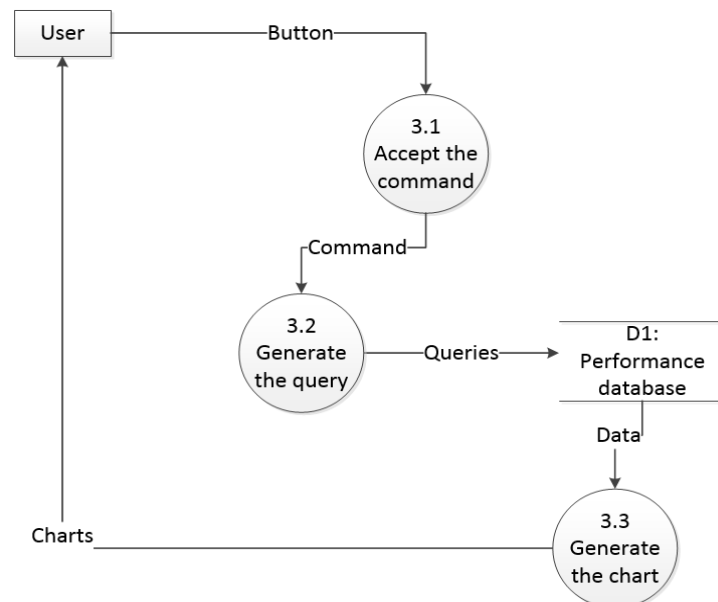


Figure 3 - Level-3.0 Data representation

The user clicks on a button and the software then generates the query corresponding to that function. The query is then used to enquire the database for related data which will be bound to the charts that are going to be displayed.

3.1.4 Level-4.0 Diagnosis

The level-4.0 diagnosis diagram is as follows:

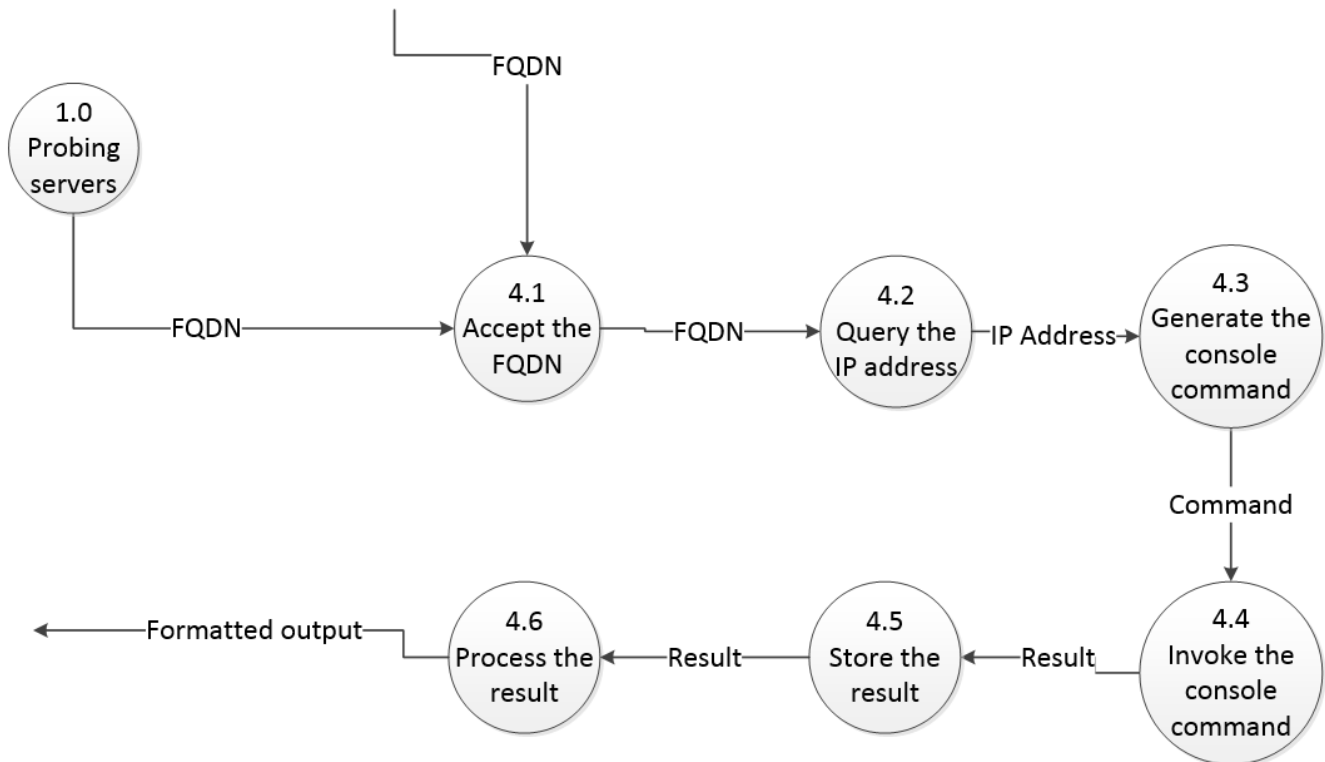


Figure 4 - Level-4.0 Diagnosis

When the server can't be reached in probing module, the FQDN will be transferred here (to the diagnosis function) and the address will be resolved to generate the console command that can be used to diagnose the server's status. The result of the diagnosis will be stored into the software which then processes and formats the results and stores them into the database.

3.1.5 Level-5.0 Import & Export Database

The level-5.0 import database diagram is like following:

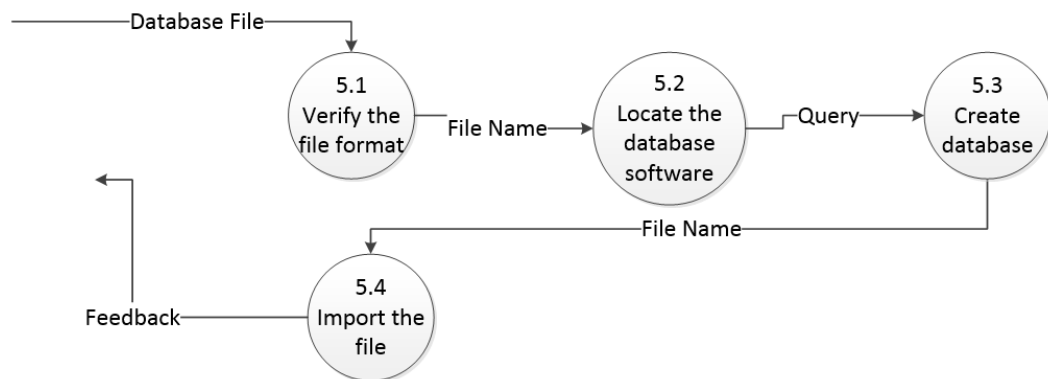


Figure 5 - Level-5.0 Import Database

The user supplies the database file which will be verified that it has the correct format. Then the database software's location is read from the configuration file and the query is generated to create the database first in order to import from the file. Feedback is generated to tell the user that if the import is successful or the file format is incorrect.

The level-5.0 Export database diagram is like following:

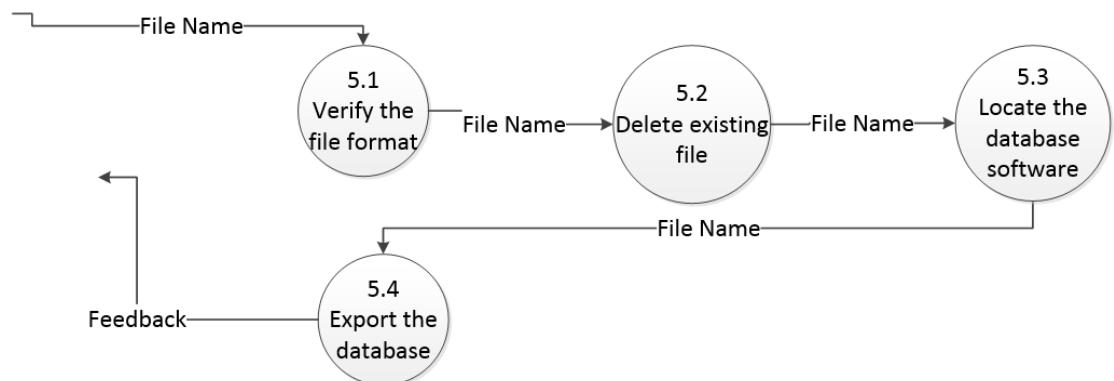


Figure 6 - Level-5.0 Export Database

The user supplies the file name whose extension will be examined first. Then the program will delete the existing file if it exists, locate the database software and export the database to the specified file.

3.1.6 Level-6.0 Tables & Charts Export

The level-6.0 tables export diagram is like following:

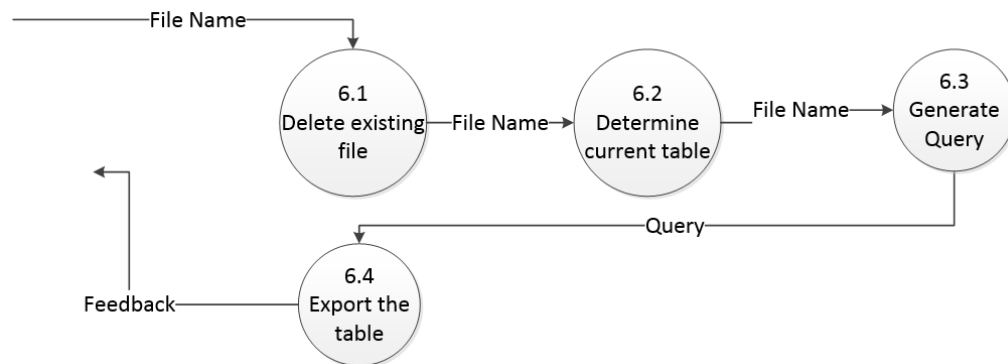


Figure 7 - Level-6.0 Tables Export

The user supplies the file name and if the file already exists it will be deleted. Then the program determines what table is being displayed and generates the corresponding query to the database to export the table.

The level-6.0 charts export diagram is like following:

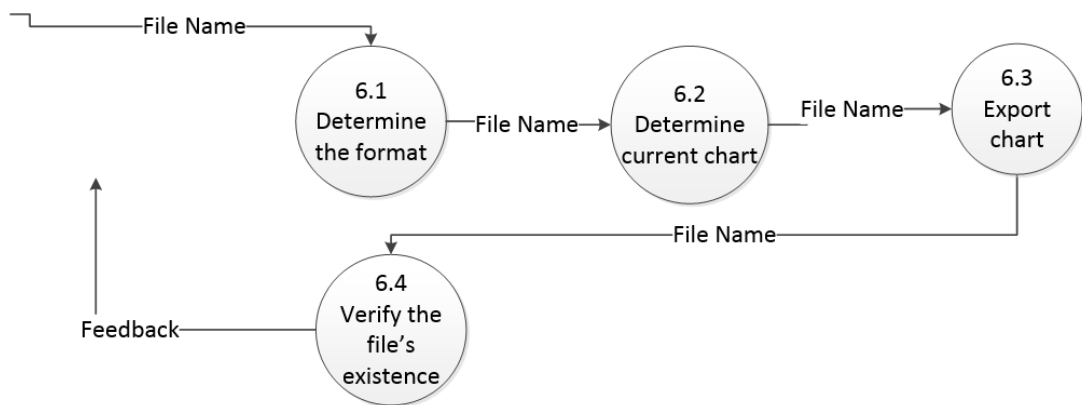


Figure 8 - Level-6.0 Charts Export

The user supplies the file name and the program will determine the format and the chart that is being displayed. It will then export the chart and verify if the chart has been exported successfully.

3.2 Level-2 Data Flow Diagram

Level-2 DFD follows from level-1 diagram which expands the level-1 diagram to a more detailed view of specific functions. It has the numbering format of X.Y.Z representing the functions indicated in level-0, level-1 and level-2 diagrams. The following diagram provides the level-2 DFD of function 1.3 in section 4.1. Function 1.4 in section 4.1 is almost the same as this so the author omitted it.

3.2.1 Level-2 DFD for function 1.3 Probing using IPv4 address

The Level-2 DFD for function 1.3 diagram is as follows:

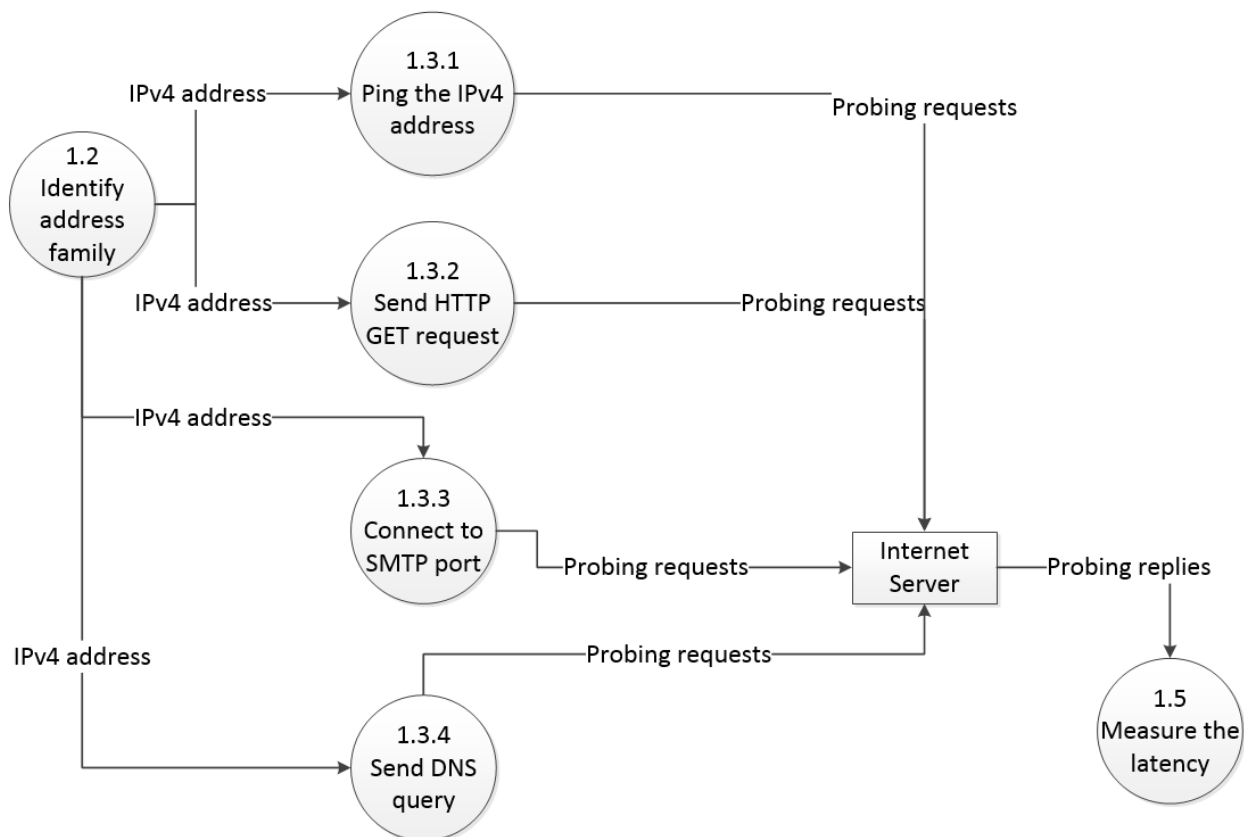


Figure 9 - Level-2 DFD for function 1.3

After the address family is identified, the addresses can be pinged using a "Ping" function, sending Hyper Text Transfer Protocol (HTTP) GET request, connecting with Simple Mail Transfer Protocol (SMTP) port and sending with Domain Name Service (DNS) queries to probe the Internet Servers. The reply from the probing function will then be measured for latencies in function 1.5.

3.3 Process Description

The process description will be narrated using structured English to represent the logic as follows:

Process 1.0: Probe the servers

DO

 GET List-of-servers

 READ next-line

 GET server-information

 GET IP-address-family

 BEGIN IF

 IF IP-address-family is IPv4

 THEN GENERATE IPv4-requests

 ELSE IF IP-address-family is IPv6

 THEN GENERATE IPv6-requests

 IF every IPv6-reply is timeout

 THEN GOTO Process-4.0

 END IF

 GET elapsed-time

 IF elapsed-time is equal to maximum-time

 SET latency TO 0

 ELSE SET latency TO elapsed-time

 ADD server-information TO database

 ADD latency TO database

UNTIL end-of-list

Process 2.0: Search the database

DO

 GET textbox-content

<p> READ textbox-string GENERATE query BEGIN IF IF database contains keywords READ server record GENERATE table ELSE RETURN error-message UNTILend-of-program </p>
<p> Process 3.0: Present the data DO GET button-type GENERATE query SEND query TO database READ result SET chart-parameters EQUAL TO result SET chart VISIBLE UNTILend-of-program </p>
<p> Process 4.0: Diagnose DO GET IP-address GENERATE console-command CASE: output CONTAINS "network unreachable" SET status "network unsupported" CASE: output CONTAINS "host unreachable" SET status "host misconfigured" CASE: output CONTAINS "could not find host" </p>

<p>SET status "host unsupported"</p> <p>DEFAULT: SET status "host supports IPv6"</p> <p>ADD status TO database</p> <p>UNTILend-of-program</p>
<p>Process 5.0: Import & Export database</p> <p>DO</p> <p> GET file name</p> <p> GET database server's path</p> <p> GENERATE query</p> <p> SEND query TO database</p> <p>UNTILend-of-program</p>
<p>Process 6.0: Tables export</p> <p>DO</p> <p> GET file name</p> <p> GET database server's path</p> <p> GENERATE query</p> <p> SEND query TO database</p> <p>UNTILend-of-program</p>
<p>Process 6.0: Charts export</p> <p>DO</p> <p> GET file name</p> <p> GET chart type</p> <p> EXPORT chart</p> <p> VERIFY file</p> <p>UNTILend-of-program</p>

Figure 10 - Process description

3.4 Expanded Use Case Narratives

The expanded use case narrative captures the sequence of the messages sent from the actor in the use case to the system and the system's response to these messages. It also reveals the structure and content of these messages. (Stumpf & Teague, 2005)

3.4.1 Probing Servers

The expanded use case narrative for probing servers is as follows:

Use case:	Probing Servers
Actors:	Server, User
Purpose:	Probe the server for measuring network latencies using various methods.
Overview:	A sample subset of Internet servers will be probed using both IPv4 and IPv6 addresses to measure their performances when responding to IPv4 and IPv6 requests in different countries as well as of different service types.
Type:	Essential
Preconditions:	IPv6 connection must be available on local machine. Server's domain name must be known to be resolved.
Post-conditions:	The server record must be able to be updated or inserted into the database.
Special Requirements:	Complete probing for one server should be finished in 10 seconds.
Flow of Events	
ACTOR ACTION	SYSTEM RESPONSE
1. This use case begins when a user clicks on the "Probe" button or "Probe List" button.	
2. The user provides the server's domain name or the list of servers that need to be probed.	3. Probe these servers and provide information on the screen. If the server can't be reached, the diagnose function will be performed. If "Probe List" is chosen, the results are automatically stored into database.
4. The user decides whether to store the individual probing results by clicking on the "Insert" button or not.	5. The system will perform the update action if record already exists or insert action if it doesn't. Result will be displayed.
6. The servers have been probed and results are stored.	
Alternative Flow of Events	
Line1: Blank input will make the program probe local machine. Jump to Step 3. Inputting the addresses directly has no negative effect to the program. Jump to Step 3. Servers that can't be resolved will indicate error message and jump to Step 3. Line3: The thread will be aborted if request times out. Jump to Step 4.	

Table 2 - Expanded use case narrative: probing servers

3.4.2 Searching Data

The expanded use case narrative for searching data is as follows:

Use case:	Searching Data
Actors:	User
Purpose:	Search the data inside the database to retrieve related records.
Overview:	The user requests that records in the database about related server's information need to be retrieved and displayed on the screen using table format.
Type:	Essential
Preconditions:	The servers have been probed completely and measurement data were stored into the database. The database connection must be opened.
Post-conditions:	The data retrieved can be sorted again by columns and copied.
Special Requirements:	Data searching should be completed in 1 second in normal condition.
Flow of Events	
ACTOR ACTION	SYSTEM RESPONSE
1. This use case begins when a user clicks on the "Search" button.	
2. The user provides the search keywords and chooses the range of search.	3. Displays the related results from the database.
4. The user types in additional words in the textbox.	5. Display only records containing the new keywords from current table.
6. The user clicks on the column names.	
	7. The records are sorted according to the direction of the arrow next to the column header.
8. The data retrieving is complete.	
Alternative Flow of Events	
Line2: The user doesn't input anything but chooses the ranges of continent or country, all records belonging to the continent or country will be displayed. Jump to Step 3.	
Line3: If no record is found the program displays a blank table with only the column headers. Jump to 4.	

Table 3 - Expanded use case narrative: Searching data

3.4.3 Representing Data

The expanded use case narrative for representing data is as follows:

Use case:	Representing Data
Actors:	User
Purpose:	Represent the data as diagrams.
Overview:	The user requests that the data in the database are displayed as diagrams in the chart area to indicate the percentages of IPv6 available servers in the database in different countries and continents. The performance of single server can also be displayed.
Type:	Essential
Preconditions:	The servers have been probed completely and measurement data were stored into the database. The database connection must be opened.
Post-conditions:	The data represented are easily understood in the chart.
Special Requirements:	Data representation should be completed in 1 second in normal condition.
Flow of Events	
ACTOR ACTION	SYSTEM RESPONSE
1. This use case begins when a user clicks on the buttons requesting for different representation of the servers.	
2. The user chooses the countries that are to be compared or enters the server name for performance representation.	3. Displays the respective charts after calculating the data from database.
4. The data representation is complete.	
Alternative Flow of Events	
Line2: The user doesn't input anything and clicks on the "Compare this server" button. The data will be calculated based on the performance of local machine. Jump to Step 3. The user chooses same countries to compare. The program will display only data for one country. Jump to Step 3.	
Line3: If no record is found the program returns feedback message. Return to Step 1.	

Table 4 - Expanded use case: representing data

3.4.4 Import & Export Database

The expanded use case narrative for import & export database is as follows:

Use case:	Import & Export database
Actors:	User
Purpose:	Import the database from a file and export the database to a file.
Overview:	Database needs to be exported to a file to prepare for backup. It also needs the import function to restore the database.
Type:	Optional
Preconditions:	The database connection must be opened.
Post-conditions:	The database will be exported or imported.
Special Requirements:	N/A
Flow of Events	
ACTOR ACTION	SYSTEM RESPONSE
1. This use case begins when a user clicks on the "Import database" button or "Export database" button.	
2. The user inputs the name of the file to be exported to or import from.	3. Import/export the database to the specified file.
4. The database import/export function is complete.	
Alternative Flow of Events	
Line2: The user inputs a file name with wrong extension. Repeat Step 2.	

Table 5 - Expanded use case narrative: import & export database

3.4.5 Tables & Charts Export

The expanded use case narrative for tables & charts export is as follows:

Use case:	Tables & Charts export
Actors:	User
Purpose:	Export the displayed tables and charts to files.
Overview:	Tables and charts being displayed can be exported to files with correct formats.
Type:	Optional
Preconditions:	The database connection must be opened. Users clicked on a button that can display a table or chart or both.
Post-conditions:	The table or chart will be exported to a file.
Special Requirements:	Charts should be able to be exported to EMF format files.
Flow of Events	
ACTOR ACTION	SYSTEM RESPONSE
1. This use case begins when a user clicks on the "Export table" button or "Export" button next to the chart.	
2. The user inputs the name of the file to be exported to.	3. Export the table or the chart to the specified file.
4. The table/chart export function is complete.	
Alternative Flow of Events	
Line2: The user inputs a file name with wrong extension. Continue to Step 3.	

Table 6 - Expanded use case narrative: tables & charts export

3.5 Processing Control & Algorithm

The program has six main functions: server probing, data storing and searching, data representation, diagnosis, import & export database and tables & charts export. In the probing function, the control of the system is passed down to the sub-processes that call the file dialog to read the list of servers. The servers are interpreted as Fully Qualified Domain Name and transferred to the main function. The function then calls the Domain Name System (DNS) library to resolve the address. Both IPv4 and IPv6 addresses will be resolved if any of them exists. The address will be passed to the IPv4 and IPv6 functions to call sub-processes to complete different kinds of probing tasks. The replies of these sub-processes will be passed to the calculation sub-process to calculate the delays which will be passed onto the main function. This process recurs iteratively until the end of the list of servers has been reached.

In the data storing and searching function, the server data will be passed from the probing function to this main function. If data already exists, the data inside the database will be updated by invoking the sub-process to execute the queries to update the record. If the data doesn't exist, a new record will be inserted into the database. During this process, the control of the thread will be passed onto the sub-processes which will then return it to the main function. The application can then capture the input from the user and search the database by invoking sub-processes to read the data inside the database. When the data is read, the control is passed to the user interface thread to display the data on the screen.

In the diagnosis function the data is transferred from the server probing module to this main function. Alternatively the input from the user can be captured by sub-process too. The main function passes control to the "Diagnose" process, to determine the cause of server unavailability. The diagnosis output is extracted and is passed to the main function which will then transfer the output to the interpretation process in order for it to compare the output with predefined criteria and store the result of this comparison into the database. Individual diagnosis results will also be displayed on the screen by invoking the user interface thread.

In the import & export database function the file name is transferred from the file dialogue module to the main function. Depending on the type of operation the file name is passed to either the import module or export module. These module will generate respective queries that will be executed by the execute query module. When the operation is finished, the control is passed back to the main function.

In the tables & charts export function the file name is transferred to the main function. In the case of exporting tables, the file name will be passed to a sub module which generates the query according to the table types passed from the representation processes. The query will then be passed to the execute query module and executed. In the case of exporting the charts, the type of chart is also passed from the representation processes. The chart will be exported according to both the chart type and the file extension. When the operation is finished, the control is passed back to the main function.

3.6 Structure Chart

3.6.1 Server Probing

The structure diagram for server probing is as follows:

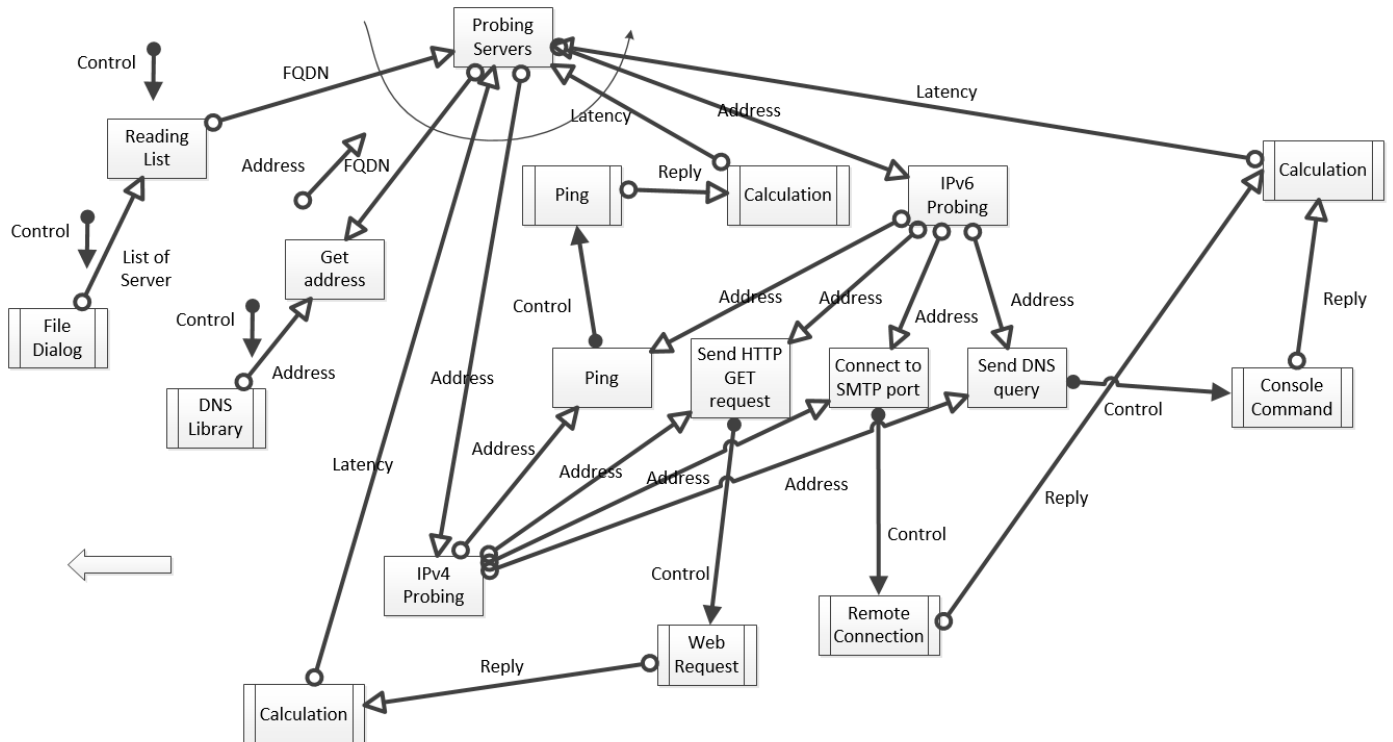


Figure 11 - Structure diagram: server probing

The file dialog reads the list of servers along with their names, passing them onto the main function (Probing Servers). The controls (which consists of File Dialog, DNS Library, Remote connection etc.) of these components are passed back to it. The DNS library then resolves the names into addresses which will be passed back to main function. The main function then passes these addresses to different sub-processes to go through Ping, Sending HTTP GET requests, connecting to SMTP port and Sending DNS queries. The latencies for these processes will be calculated and sent back to main function. This is an iterative process.

3.6.2 Storing Data

The structure diagram for storing data is as follows:

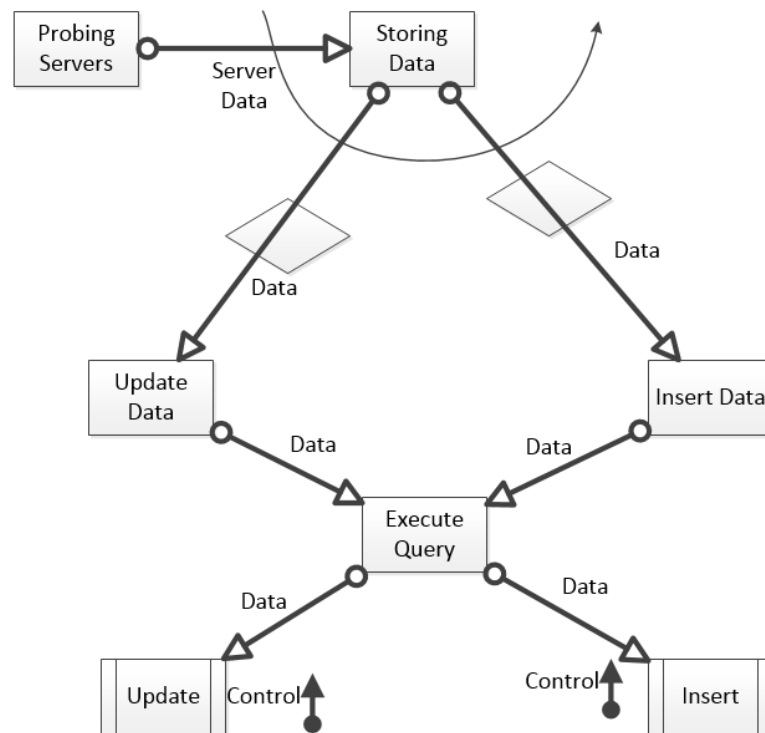


Figure 12 - Structure diagram: storing data

The data from probing servers are transferred into the main function (Storing Data) which decides whether the database has the records; if the records exist, the data will be updated. If the database doesn't contain the records, the data will then be inserted into the database. The data is passed down to the queries and the control is passed back to the main function after the completion. This is an iterative process.

3.6.3 Searching & Representing Data

The structure diagram for searching data is as follows:

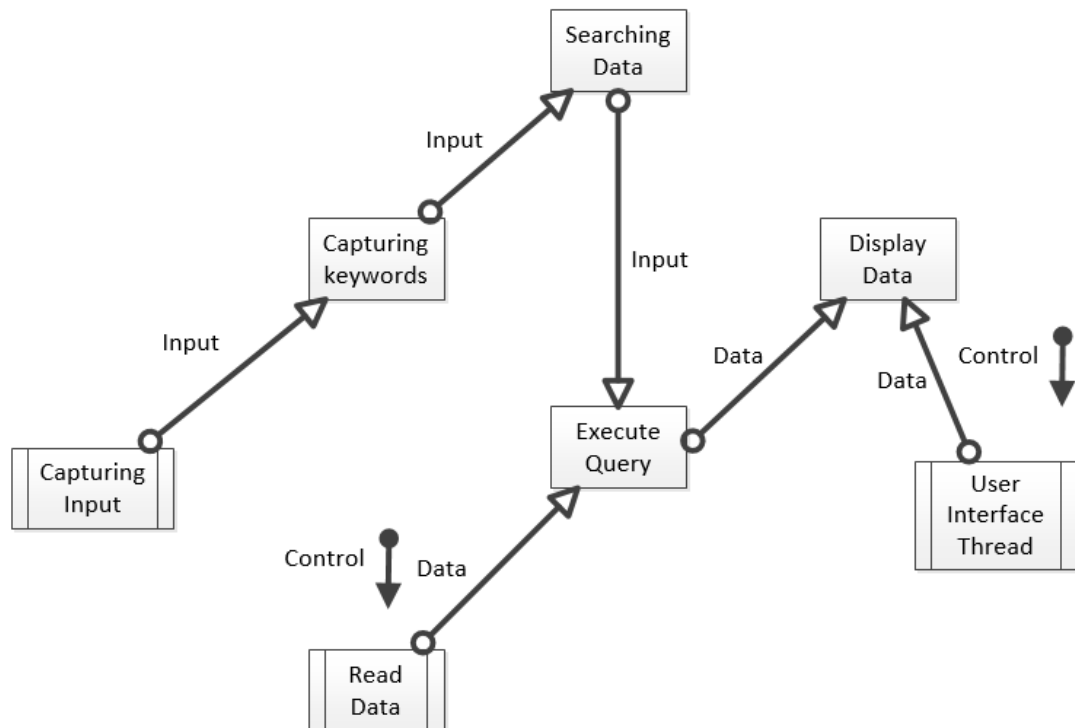


Figure 13 - Structure diagram: searching & representing data

The input of the user is captured as keywords which will be passed onto the main function (Searching Data). The main function sends the input to the module that generates the queries to read the data. The controls (which consist of User Interface Thread, Read Data etc.) are passed onto it. The data is sent to the main function which will then in return send it to the module that will display the data using user interface thread.

3.6.4 Diagnosis

The structure diagram for diagnosis is as follows:

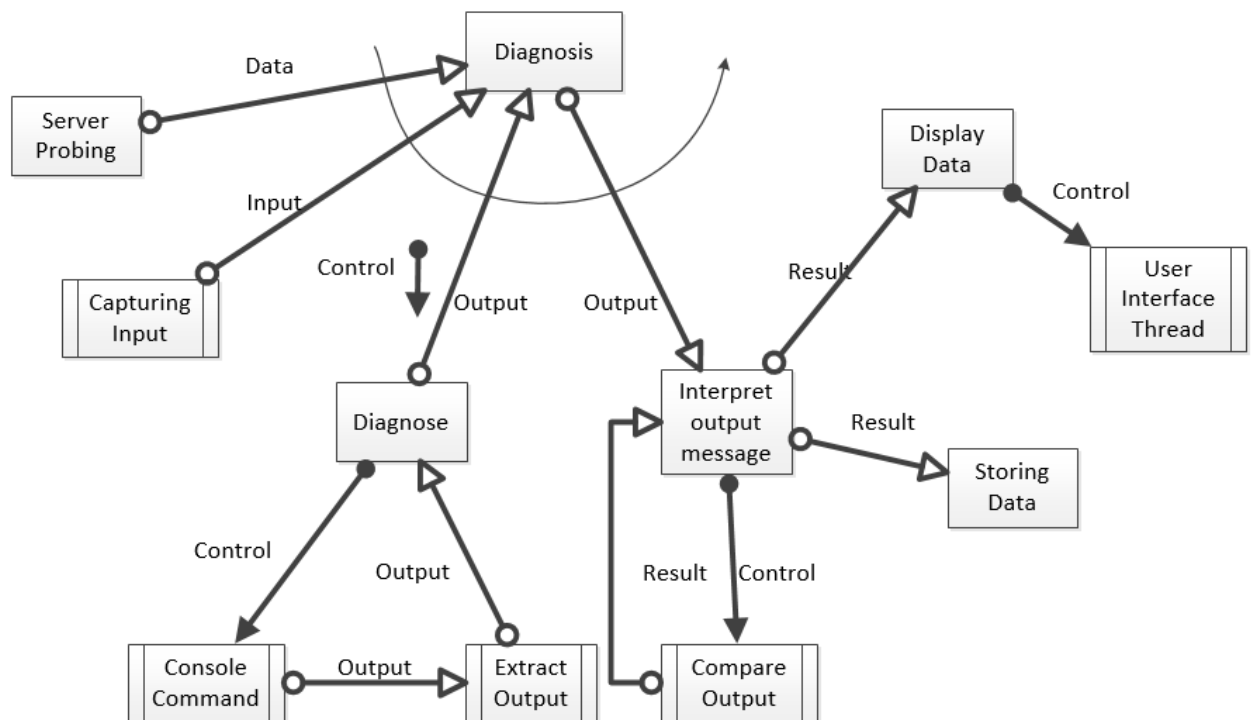


Figure 14 - Structure diagram: diagnosis

The data from probing is sent to the main function (Diagnosis). The keywords from the user input will also be captured. The control (which consists of Diagnose, Console command, Compare output etc.) is then passed down to a process to diagnose the server. This process will call the sub-process that invokes the console commands to diagnose and another sub-process will extract the output to the Diagnose process. The output of the diagnosis is then passed onto the other processes that will interpret the output by comparing it with pre-defined rules. The result of the comparison is stored into database and displayed on screen by invoking the user interface thread. This is an iterative process.

3.6.5 Import & Export Database

The structure diagram for import & export database is like following:

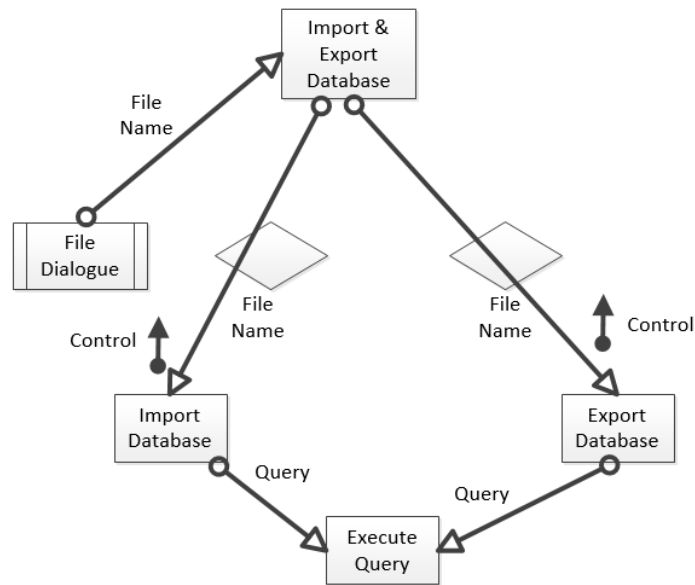


Figure 15 - Structure diagram: import & export database

The file dialogue supplies the file name for importing/exporting database. The file name is passed to the import module or export module depending on the button clicked. The query is generated to import/export the database. After this the control is returned to the main process.

3.6.6 Tables & Charts Export

The structure diagram for tables & charts export is like following:

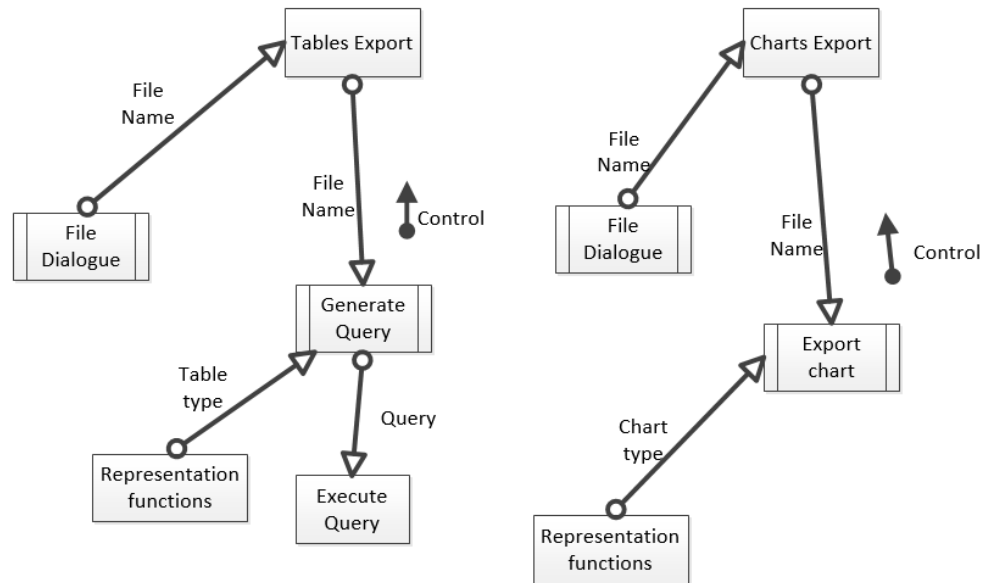


Figure 16 - Structure diagram: tables & charts export

The file dialogue supplies the file name for exporting the table/chart. The file name is passed to sub-processes to either generate the query according to the table type or export the chart according to the chart type. The type of table/chart comes from the representation process. In table export the query is executed to export the table. After this the control is returned to the main process.

4. Software Choices

The name of the software is: IPv6 performance analyser. Current version is: Build 1.0.

The explanation and working of this software is not necessary to be explained here.

This software is built based on C# algorithm, Python and works on all Microsoft Windows platforms.

5. Architecture/Infrastructure

5.1 Infrastructure overview

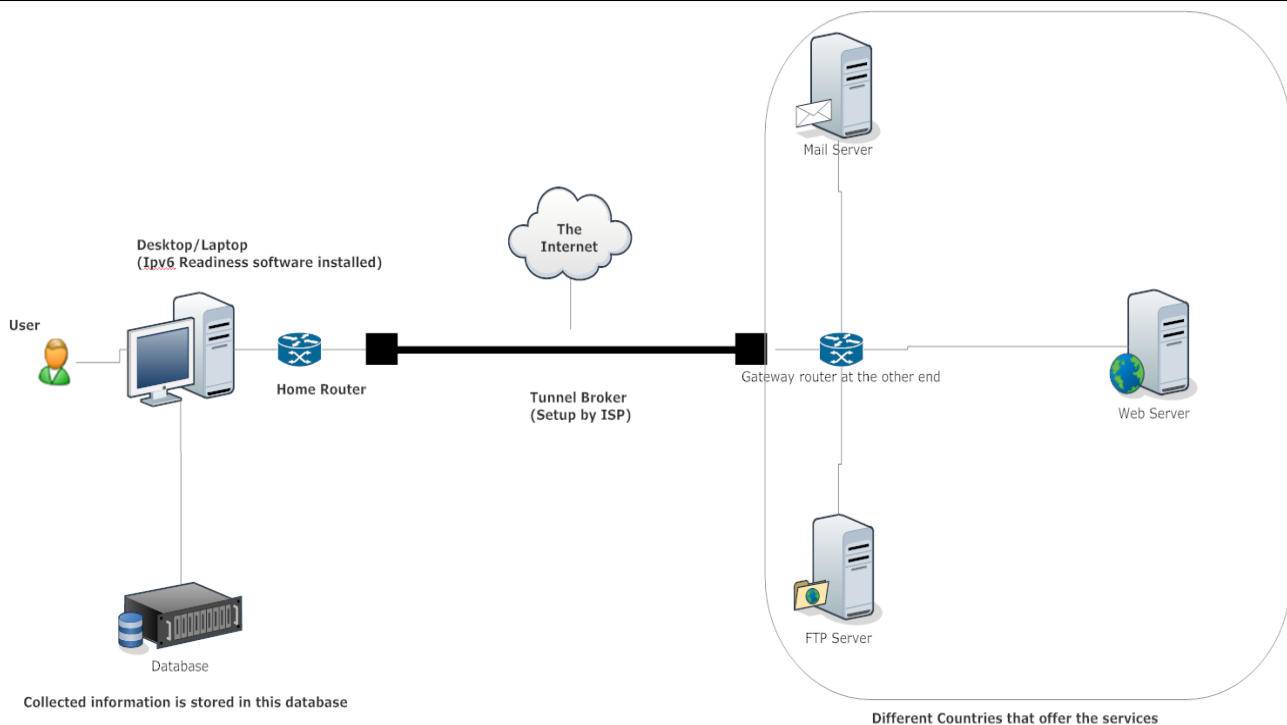


Figure 17 - Infrastructure overview diagram

The user uses the software to analyze the performance of the Internet servers that are of different types including web server, mail server and so on. The tunnel is setup from the computer to an Internet tunnel broker's server to provide IPv6 connectivity to local machine. The measurement results are then stored on to the local database server.

5.2 Architectural Diagram

The architectural diagram is as follows:

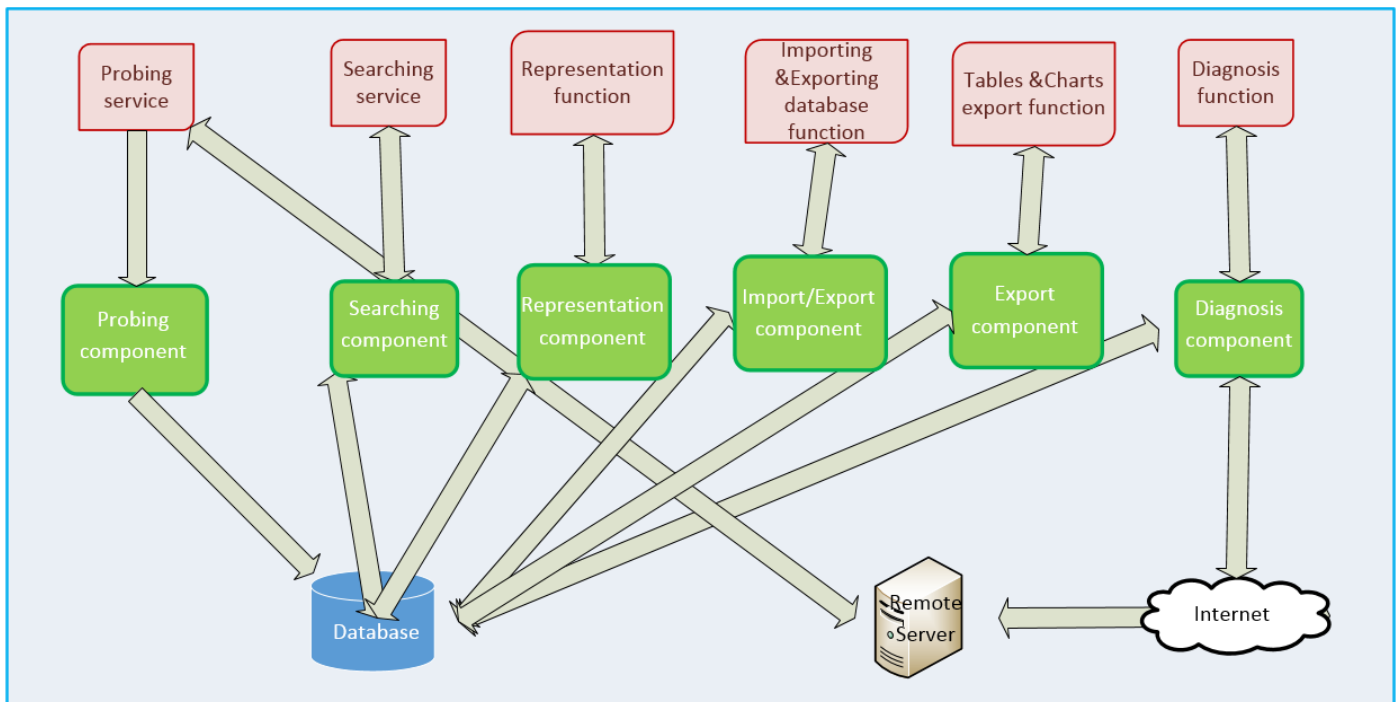


Figure 18 - Architectural Diagram (Nordgren, 2010)

In this program, the components interact with each other and the external environment. The probing function requests the remote server for response and calls the diagnosis as well as the storing function when batch probing is used. The probing function also invokes console commands from the operating system and the .Net framework functions to call the network related functions in order to complete the probing function. The storing function connects to the database by referencing the database library functions. The search function interacts with the database and represents the data within the software. The representation function calculates the data in the database by creating temporary views in the database and represents them as diagrams. The diagnosis function calls the .Net framework again to diagnose the problem and send requests throughout the network adapter. The responses are interpreted and stored into database. The import & export database function uses the import & export components to import/export the database and the tables & charts export function also queries the database to export the table.

5.3 Infrastructural Requirements

- **Capacity:** The software should be able to handle the probing requests of about 5,000 servers on a list sequentially. The data it sends to every server will be about 1KB so the total traffic incurred should be about 5MB.
- **Performance:** The software should be able to respond in 2 seconds during normal conditions except when using the probing functions which may vary from 10 seconds to 30 seconds depending on the network condition.
- **Integration & compatibility:** The software is only compatible with Microsoft Windows platform and the Dynamic Link Library (DLL) files are necessary to perform network probing and database access.
- **Platform Strategy:** A lot of functions will be realized by invoking console commands in Windows platform. These commands are all available in Linux but the program can only run on Windows.
- **Security:** The software may suffer from Structured Query Language (SQL) injection but should not have an issue in buffer overflow.
- **Backup & recovery:** The database must be backed up manually and the recovery of data is done on the basis of the backup.
- **Scalability:** The software is scalable to be installed on multiple machines. Multi-thread processing is possible in the future.
- **Future proofing:** The software combines different functions that can still be used in the future even if similar program emerges.

5.4 Alternative Design

An alternative design to this software is designing two pieces of software where one is used for probing and storing while the other is used for searching and representing data. In this design the probing and storing functions are not necessarily realized by C# language and can be realized by script languages such as Python. The data can be measured by sending requests using the script language, and stored into the database, and this data can be retrieved and represented using C# as in current design.

6. Interface Design

6.1 Application Interface Design

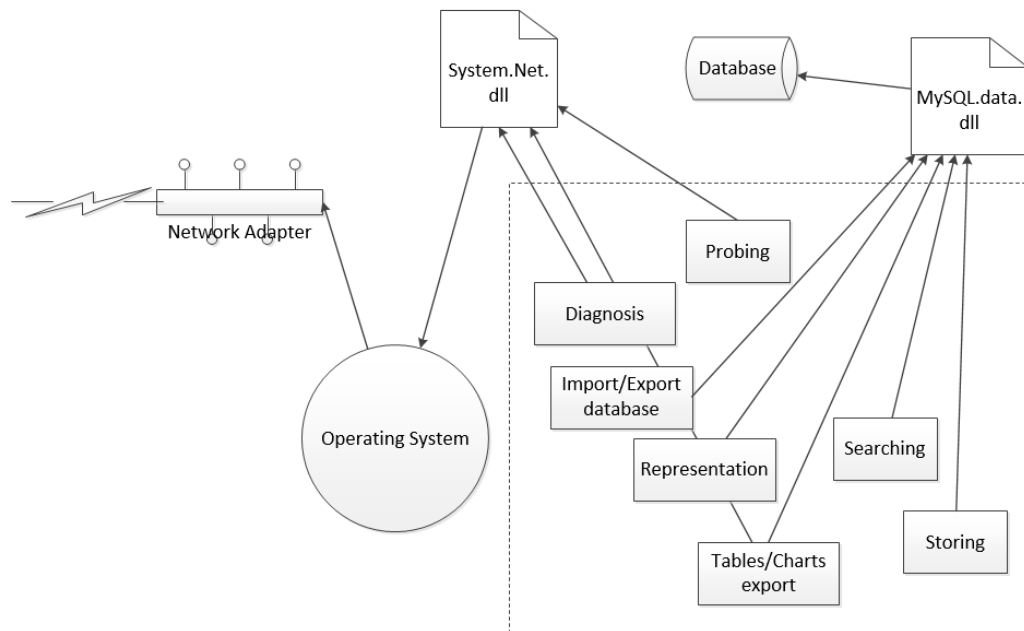


Figure 19 - Application interfaces

All functions inside the box are implemented. Diagnosis, Exporting charts and Probing functions will use the system.net.dll file to reference the .Net framework which will invoke relevant network commands in the operating system to interact with remote servers throughout the network adapter. Representation, Searching, Storing, Importing/Exporting database and exporting table functions will use the MySQL.data.dll file to connect to local MySQL server to access the database that can then perform these functions by storing and retrieving data.

6.2 User Interface Design

The software will have a user interface as follows:

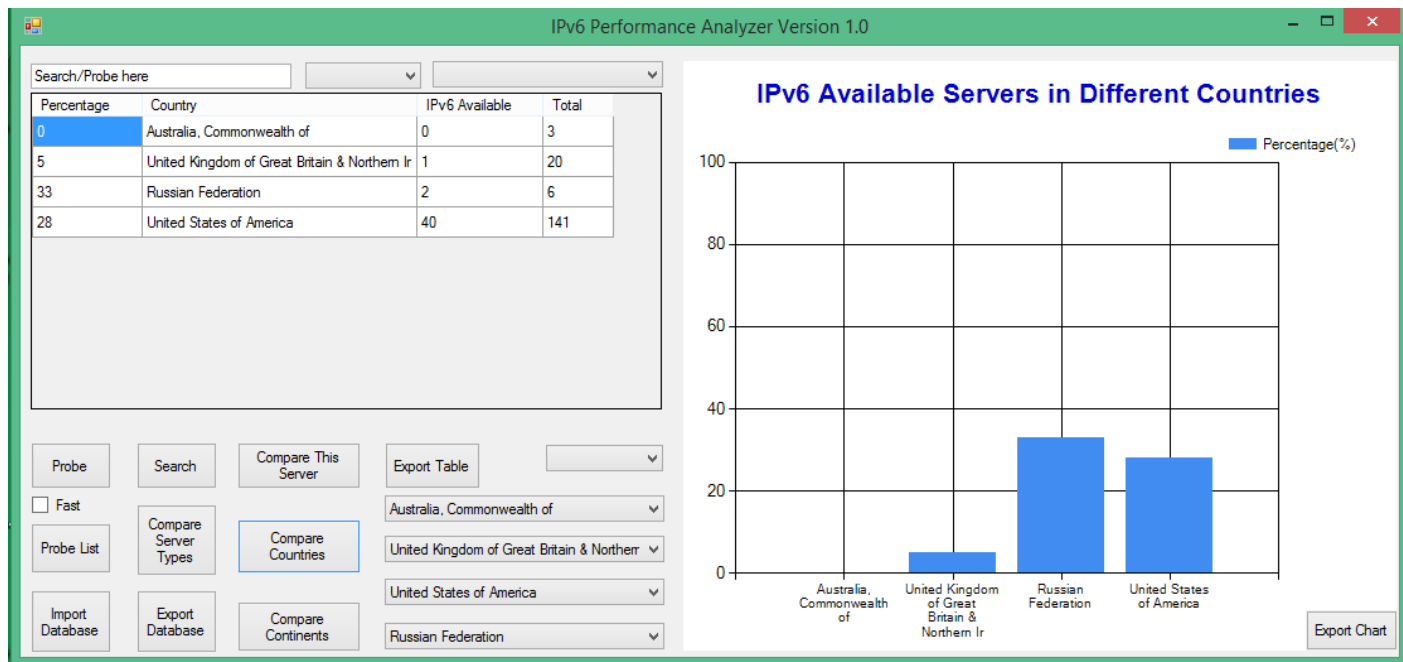


Figure 20 - User interface

This user interface is mainly a form that runs on Windows platform. The resolution is set to about half the window's size. All functions can be called by interacting via this user interface.

6.3 Interface Specifications

6.3.1 Input

- **Search/Probe Here:** server's domain name can be input here to probe or diagnose and search keywords can be input here to search the database
- **Continent:** A dropdown list including all the continents can be selected here
- **Country:** A dropdown list to select the country to narrow the range of search as in "Continent"
- **Service:** A dropdown list which specifies the type of service of the servers when using the representation functions
- **Search:** The button that can generate the query to search the database
- **Progress bar:** This bar will be used when probing a list of servers
- **Probe:** This button will probe a server whose name is entered on the "Host/Search" textbox

- **Probe List:** This button will probe a list of servers that can be chosen from the file dialog
- **Compare Server Types:** This button will display the percentage of numbers of IPv6 available servers using server type as the abscissa
- **Compare Continents:** This button will display the percentage of numbers of IPv6 available servers in different continents
- **Dropdown Lists:** Countries can be selected here to compare the percentages of IPv6 available servers in these countries by clicking on "Compare Countries"
- **Compare This Server:** The performance of IPv4 and IPv6 will be compared for only this server whose name is typed in the Host/Search textbox
- **Import Database:** The database will be imported from a database file
- **Export Database:** The database will be exported to a file
- **Fast:** Checking this will enable the fast probing when probing servers
- **Export Table:** The table being displayed will be exported to a file
- **Export:** The chart being displayed will be exported to a file

6.3.2 Output

- **Table:** This area displays the search result that will display the related information of the servers including name, country, continent etc. The column width can be adjusted by dragging the border so that contents can be displayed if hidden. The scrollbar will be implemented too. Clicking on the column headings can sort the list with ascending order or descending order by text. Typing keywords again in the Host/Search field will sort the displayed data to only contain the records containing these keywords. This table area is hidden until the "search" button is clicked.
- **Feedback Window:** This window is visible by default and is of the same size and location with the table above. It displays the message reflecting the information obtained by probing and the diagnosis result. The results can be inserted into the database by clicking on the "Insert" button.
- **Chart Area:** This area represents the result of calculation from the database which displays the resulting tables statistically to indicate either the percentages of the IPv6 servers or the latencies of the servers. Different kinds of charts will be used here and each of them will only be visible whenever the respective functions are called.

6.3.3 Formatting

The input to the Host/Search field must be string characters in order to be resolved as IPv4 or IPv6 addresses. If a host that has neither IPv4 nor IPv6 address, an error message will indicate this problem and the diagnosis function will be called to diagnose this problem. If an invalid IPv4 or IPv6 address is entered, the same will happen too. If a valid IPv4 or IPv6 address is inputted, the software will be able to resolve the address and perform functions correctly.

The right-click menu can be displayed on the Host/Search textbox for functions including copying, pasting etc. The table items can be multiple-selected by clicking and dragging the mouse. "Ctrl + C" can be used to copy table items to clipboard.

7. Conclusion

This document includes the data design, process design, infrastructure design and the interface design of the program mentioned in the document previously. In the data design, the relational E-R diagram is provided; the data structure and operation are explained as well as a given CRUD matrix is present. In the process design, the level-1 and level-2 DFD are provided to describe the process flows of the software. Processes are described using Structured English. Expanded use case narratives explained the process of the functions. The processing control and algorithm including the structure charts are provided to describe the hierarchy of the processes. In the architectural design different diagrams are drawn to describe the architecture of the software with the external and internal actors. Alternative design is discussed too. In interface design, both human interface and application interface are designed as diagrams. Input/ Output formatting is discussed too.

8. Appendix

8.1 Appendix A: Deliverable Task Breakdown Statement

Appendix A

Deliverable Task Breakdown Statement

Deliverable Name: *Design Document*

Project Team Name	Team Number
<i>IT08 (IPv6 Readiness)</i>	<i>IT08</i>

Team Member Names	Deliverable Percentage Completed	Date	Signature
Abdul Sami	20%	10/09/2015	<i>[Signature]</i>
Bilawal Mushtaq	17.5%	10/09/2015	<i>[Signature]</i>
Leslie Vundu	20%	10/09/2015	<i>[Signature]</i>
Man Fu Lei	25%	10/09/2015	MAN FU L2
Robert Smart	17.5%	10/09/2015	Robert

Instructions:

The purpose of this statement is to provide documentary evidence of each team member's contribution to the submitted deliverable.

Please complete the information as specified above. It should list each team member's percentage contribution to the particular deliverable. This statement needs to be signed by all group members to indicate their agreement to percentage breakdown.

The completed statement should be scanned, and copied into each group deliverable as Appendix A.

This statement should at all times be as accurate as possible, being a true reflection of work performed. That is, specify percentages according to actual contribution; do not just divide percentage evenly among team members.

8.2 Appendix B: Data Dictionary

Table	Field Name	Primary Key	Description	Data Type	Length	Not Null	Unique	Constraint	Referencing Table.Attribute	Update Action
continent	continentName	Yes	The name of the continent such as Asia	Character	45	Yes	Yes			
country	countryCode	Yes	The code of the country like "CN"	Character	45	Yes	Yes			
	countryName		The name of the country like "China"	Character	45	Yes	Yes			
	continentName		The name of the continent such as Asia	Character	45	Yes		Foreign key	Continent.continentName	Cascade
status	StatusID	Yes	The identifier of the status in this database	Integer	11	Yes	Yes			
	Description		The description of the status such as "IPv6 network unreachable"	Character	45	Yes	Yes			
server	ServerID	Yes	Identifier of the server in this database	Integer	11	Yes	Yes			
	FQDN		The Fully Qualified Domain Name of the server	Character	45	Yes	Yes			
	countryCode		The name of the country like "China"	Character	45	Yes		Foreign key	Country.countryCode	Cascade
	StatusID		The identifier of the status in this database	Integer	11	Yes		Foreign key	Status.StatusID	Cascade
addresstype	AddressFamily	Yes	Address family of the IP address that has only two types: IPv4	Character	45	Yes	Yes			

			and IPv6							
ipaddress	IPAddressID	Yes	Unique identifier of this IP address in this database regardless of its type or format	Integer	11	Yes	Yes			
	Address		The string of address such as "180.216.48.249"	Character	45	Yes	Yes			
	AddressFamily		Address family of the IP address that has only two types: IPv4 and IPv6	Character	45	Yes		Foreign key	Addressstype. AddressFamily	Cascade
weekday	WeekdayName	Yes	The name of the weekday such as "Friday"	Character	45	Yes	Yes			
service	ServiceID	Yes	The identifier of the network service in this database	Integer	11	Yes	Yes			
	ServiceName		Name of ther network service such as "HTTP"	Character	45	Yes	Yes			
test	TestID	Yes	The identifier of this test in this database	Integer	11	Yes	Yes			
	WeekdayName		The name of the weekday such as "Friday"	Character	45	Yes		Foreign key	Weekday. WeekdayName	Cascade
	ServiceID		The identifier of the network service in this database	Integer	11	Yes		Foreign key	Service. ServiceID	Cascade
	IPAddressID		Unique identifier of this IP address in this database regardless of its type or format	Integer	11	Yes		Foreign key	Ipaddress. IPAddressID	Cascade

	ServerID		Identifier of the server in this database	Integer	11	Yes		Foreign key	Server. ServerID	Cascade
	Latency		The time it takes to finish this test such as "15 millisecond"	Decimal	6 digits without decimal					
	TestTime		The time when this test was taken such as "11:05"	Time		Yes				

Table 7 - Data dictionary

9. List of References

- Nordgren, B. (2010). Geospatial ECM Architecture Using Alfresco. Retrieved September 2, 2015 from [https://collab.firelab.org/software/projects/feis-spatial/wiki/Geospatial ECM Architecture Using Alfresco](https://collab.firelab.org/software/projects/feis-spatial/wiki/Geospatial_ECM_Architecture_Using_Alfresco)
- Stumpf &Teague. (2005) *Object-Oriented Systems Analysis and Design with UML: Chapter 4 Essential Use Cases and System Sequence Diagrams*. Retrieved September 4, 2015 from <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=12&ved=0CFgQFjALahUKEwj7vO2DoN3HAhWD56YKHayQBI4&url=http%3A%2F%2Fmoodle.najah.edu%2Fmod%2Fresource%2Fview.php%3Fid%3D45188&usg=AFQjCNHuntJiUEoLluLUgilXTK70XiozYA&sig2=p2MyGp4sUuqI75XIvBTrpw>
- Schwalbe, K. (2014). *Information technology project management*. Boston, MA: Course Technology, Cengage Learning.

10. Glossary

.Net framework: is a software framework developed by Microsoft that runs primarily on Microsoft Windows

Application Interface: is a set of routines, protocols, and tools for building software applications

C#: is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines

Create Read Update Delete (CRUD) Matrix: are the four basic functions of persistent storage

Data design: is the process of producing a detailed data model of a database

Data dictionary: is a centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format

Data flow diagram: is a graphical representation of the "flow" of data through an information system

Database normalization: is the process of organizing the columns (attributes) and tables (relations) of a relational database to minimize data redundancy

Dynamic Link Library: is Microsoft's implementation of the shared library concept in the Microsoft Windows and OS/2 operating systems

Expanded use case narratives: captures the sequence of messages from an actor to the system as well as the system's response to each message

Relational Entity Relationship diagram: is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database

Structure chart: is a chart which shows the breakdown of a system to its lowest manageable levels

Structured English: is the use of the English language with the syntax of structured programming to communicate the design of a computer program to non-technical users by breaking it down into logical steps using straightforward English words

Structured Query Language: is a standard interactive and programming language for getting information from and updating a database

Sub-process: a process that is part of a larger process

Thread: is the smallest sequence of programmed instructions that can be managed independently by a scheduler, which is typically a part of the operating system

User Interface design: is the design of user interfaces for machines and software