

Remote Key/Value Database

Operational Concept Document

Version 1.0

Arshitha Suresh
CSE681 – Software Modeling and Analysis – Project 3
September 2015
SU ID: 545391055

Table of Contents

1. Introduction	4
1.1. Executive Summary.....	4
1.2. Remote Key/Value Database Specifications	5
1.3. Architectural Overview of Remote Key/Value Database	6
2. Users and Uses	7
2.1. Users.....	7
1. Business Analyst.....	7
2. Software Developer	7
3. Software Tester.....	7
4. Developer Analyst.....	7
5. Quality Analyst.....	8
6. Database Administrator	8
2.2. Uses.....	8
1. Write Client	8
2. Read Client	9
3. Views	10
Read Client	
1. Connect to Server.....	10
2. Query on Keys.....	11
3. Query on Pattern.....	12
4. Query on Time Interval.....	13
5. Display.....	14
Write Client	
1. Connect to Server.....	15
2. Add Key.....	16
3. Delete Key.....	17
4. Edit Key.....	18
5. Display.....	19
4. Remote Key/Value Database Partitions	20
4.1. Remote Key/Value Database Client Package.....	20

4.2. Remote Key/Value Database Package.....	23
5. Tasks and Activities	27
5.1. Remote Key/Value Database Client Activities.....	27
5.2. Remote Key/Value Database Client Server Activities	29
6. Critical Issues	31
1. Server is down.....	31
2. Connection terminates between client and server	31
3. Performance.....	31
4. Simultaneous Write by clients	32
7. Conclusion	33
8. References	34

1. Introduction

1.1 Executive Summary

Enhancement is part of the Software Development Life Cycle, as an extension of Project 2 -“Key-Value Database”, our Project 4 focuses on establishing a database connection to the Key/Value database created. Database plays a vital role in every computer application. It is responsible for storing related data in an organized manner, perform operations such as edit/delete/insert and also to retrieve data stored in the database. Every user of the application would be contributing to the database either by, storing or retrieving information from the database that is set up on the server. In order for the clients of the application to be able to access the database, a “**Database Connection**” needs to be established between the clients and the server. Such a tool that helps bridge the gap between the clients and the database includes features like remotely accessing the database through client server model. This architectural document proposes such a tool called “Remote Key/Value Database”.

Remote Key/Value Database is a tool which will help to establish a database connection between the clients and the database stored on the server. The server is set up to store the Key/Value noSQL database. The Remote Key/Value Database is built on a client-server based approach which allows users to perform database operations remotely. The client will have graphical user interface which is implemented using Windows Presentation Foundation (WPF). It will give user the option to specify server address and the databases available on the server to which the connection needs to be established. At the server end, we will have Query Engine that breaks down the requests sent by the clients to analyze what operation needs to be performed on the database (read or write). Communication between client and server will be implemented using Windows Communication Foundation (WCF).

Intended users of this Remote Key/Value Database:

- 1) Software Developer
- 2) Software Tester
- 3) Business Analyst
- 4) Developer Analyst
- 5) Quality Analyst
- 6) Database Administrator

While designing the Remote Key/Value Database, we also have to consider critical issues and identify solutions to make it work efficiently. Listed below are the critical issues:

- 1) Database Server is down
- 2) Connection terminates between client and server
- 3) Performance

1.2 Remote Key/Value Database Specifications

The purpose of Remote Key/Value Database is to use the noSQL database we built in Project #2 on a remote location.

1) Client should meet the below requirements:

- The client will have graphical user interface which list all the active servers that are available for establishing the database connection.
- Every server should then send list of databases it contains to the client. This list will be displayed on the GUI so that the user knows what databases are available on each server.
- User will select the server and the database on which it wants to make a connection and perform operations.
- Client sends out the requests, it wishes to perform on the database in the form of messages to the server.
- Client receives the response over a response queue and displays on GUI.
- Clients can either perform a read or write operation on the database. Based on the operation performed they can be divided into Read clients and Write clients.
- Read clients are clients that send query requests to the remote database.
- Write clients are clients that send data to the remote database.

2) Server should meet the below requirements:

- Before receiving the request, all the servers should maintain a table containing the database names they store that will be exposed to the clients.
- After receiving the request from the client and establishing a connection, the respective server will analyze the type of operation to be performed.
- Server should perform the appropriate operation on receiving requests from the clients.
- Server is also responsible to send out results of operations performed on the database to appropriate clients.

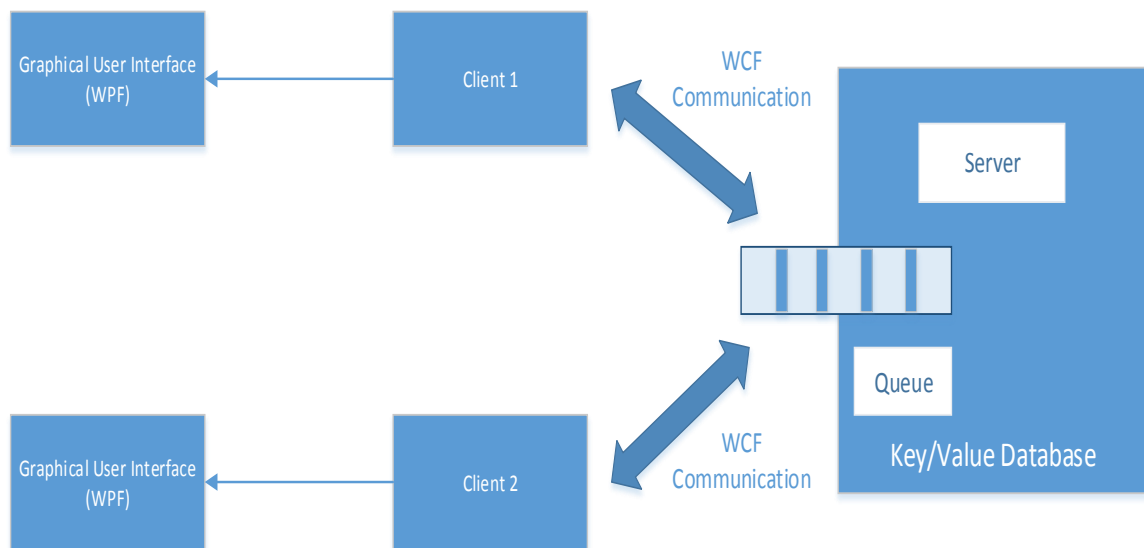
3) We shall use Windows Communication Foundation for all the communication between client and server. Communication should be based on message passing.

4) There are multiple client and servers so there will be multiple requests coming in. We are using queuing technique to store the request from the client.

5) All client display shall be implemented with Windows Presentation Foundation (WPF).

1.3 Architectural Overview of Remote Key/Value Database

The Remote Key/Value Database is a client-server based model, where the database is stored on a server. There will be multiple concurrent users of the whole system. So, here critical part of the architecture is that how the system handles the concurrent requests from multiple clients and how the responses of these requests are generated. The figure below describes the architectural overview of the Remote Key/Value Database.



Graphical user interface lists all the active servers and the databases available on the respective server. The user then selects the server and the database name from the graphical user interface. Clients request the server to establish a database connection. Multiple clients can request the same server, in this case all the requests are placed in the queue. Server uses query engine to analyze the requests received. After receiving the request from client, Server then performs the read or write operations on the database. It then sends the output generated back to the client. Based on the given option, client then displays the output on the GUI.

2. Users and Uses

2.1 Users

Remote Key/Value database presents the ability to adapt itself according to a wide range of user needs. Listed below are those users who interact with Remote Key/Value database in different ways:

1. Business Analyst

Business Analyst acts as a mediator between the client and developer. He is responsible for gathering requirement and updating the team for any changes in the project. He would perform an Internal User Acceptance testing before giving it to the client. He will use the remote key/value database to ensure that all the requirements are met by generating the output report.

2. Software Developer

The main role of the software developer is to understand the requirements thoroughly and implement the code in the most optimized fashion. This tool is primarily designed for developers because they are ones who actually develop the code. They use the database based on the requirements of the client-server application. They help ensure in smooth functioning of further life cycle.

3. Software Tester

Tester is responsible for doing thorough testing of the entire project like analyzing client requirements, preparing test strategy, understanding the application under test. He would run the remote key/value database in order to understand that all the modules operate as per requirements. Based on the analysis, he will make effective test plan that minimizes retest when we find defects requiring to the code.

4. Developer Analyst

Developer Analyst is responsible for assigning task to the individual members and checking progress on a daily basis. He also ensures that the developers are up to date about the requirements. He would use remote key/value database to check the requirements are met and are up to date because he is the one doing code review for the team members. During the review process, he would review all the modified packages and their specifications.

5. Quality Analyst

Quality Analyst is responsible for ensuring any performance related issues of the key/value database. He ensures that the database works efficiently and in case of issues, he is responsible for taking care of performance tuning of the database.

6. Database Administrator

Database Administrator (DBA) is responsible for the overall control of the database at a technical level. In database environment, the DBA authorizes access to the database making sure that the database is accessed by authorized users thus protecting the database. DBA is also responsible for coordinating and monitoring its use and acquiring software , hardware resources as needed. Not everyone must be able to access the database, in which case the data can be easily corrupted.

2.2 Uses

Below are the uses of the Remote Key/Value Database:

Write Clients :

Application	Write clients send data to the remote database.
User	User who uses the Key/Value database
Applies to	Remote Key/Value Database
Steps	<ol style="list-style-type: none"> 1. The writer client inserts data on to the database once the connection has been established. 2. User enters the key and value data that needs to be inserted along with the child relationship details. 3. Client sends the data to be inserted to the server
Events	<ol style="list-style-type: none"> 1. Data to be inserted is collected from the user. 2. Client sends the data that needs to be inserted into the database. 3. On successful insertion a success message would be sent from the server to the client as acknowledgement. 4. Error message would be displayed to show that an error occurred during insertion of the data in case of failure. For example : duplicate key insertion, an error message saying : "Key already exists!!!".

Read Clients :

Application	Read clients are clients that send query requests to the remote database.
User	User who uses the Key/Value database
Applies to	Remote Key/Value Database
Steps	<p>1. Client can query the database on the server about the following –</p> <ul style="list-style-type: none"> ○ Value of a specified key ○ Children of a specified key ○ Set of all keys matching a specified pattern which defaults to all keys. ○ Keys that contain a specified string in their metadata ○ Keys that contain values written within a specified time date interval <p>2. Based on the query submitted by the client the database would be queried for results matching the query</p> <p>3. On completion of execution of the query, the results would be sent back to the client.</p>
Events	<ol style="list-style-type: none"> 1. Client sends query requests to the server 2. Server executes the query and returns results back to the clients 3. On successful completion of execution of the query the output containing the matching result would be displayed by the client. 4. Else an error message stating the reason for the exception would be displayed by the client.

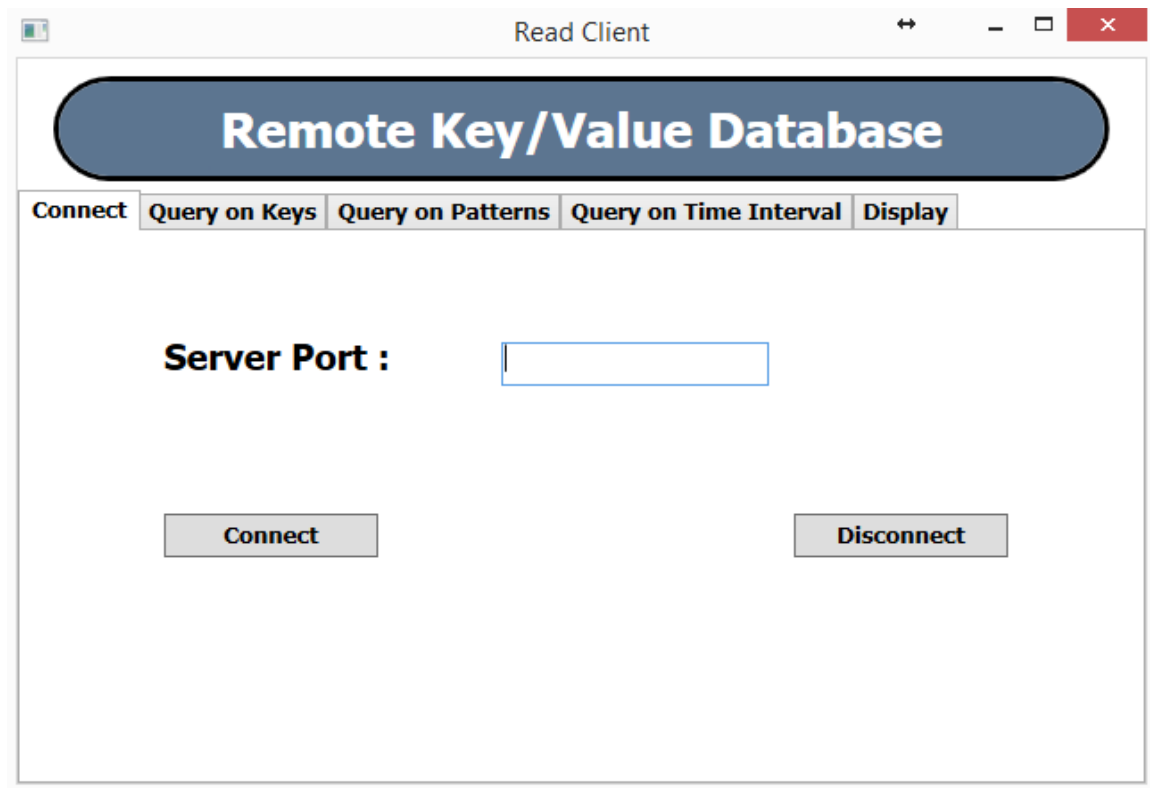
3. Views

Remote Key/Value Database client provides a graphical user interface to the user for interacting with the system. Remote Key/Value database can have two types of clients – Read client or Write client. Write client is not obligated to have a GUI. Read client graphical user interface is demonstrated and will be implemented using Windows Presentation Foundation WPF).

3.1 Graphical user interface for Read Client contains the following panes:

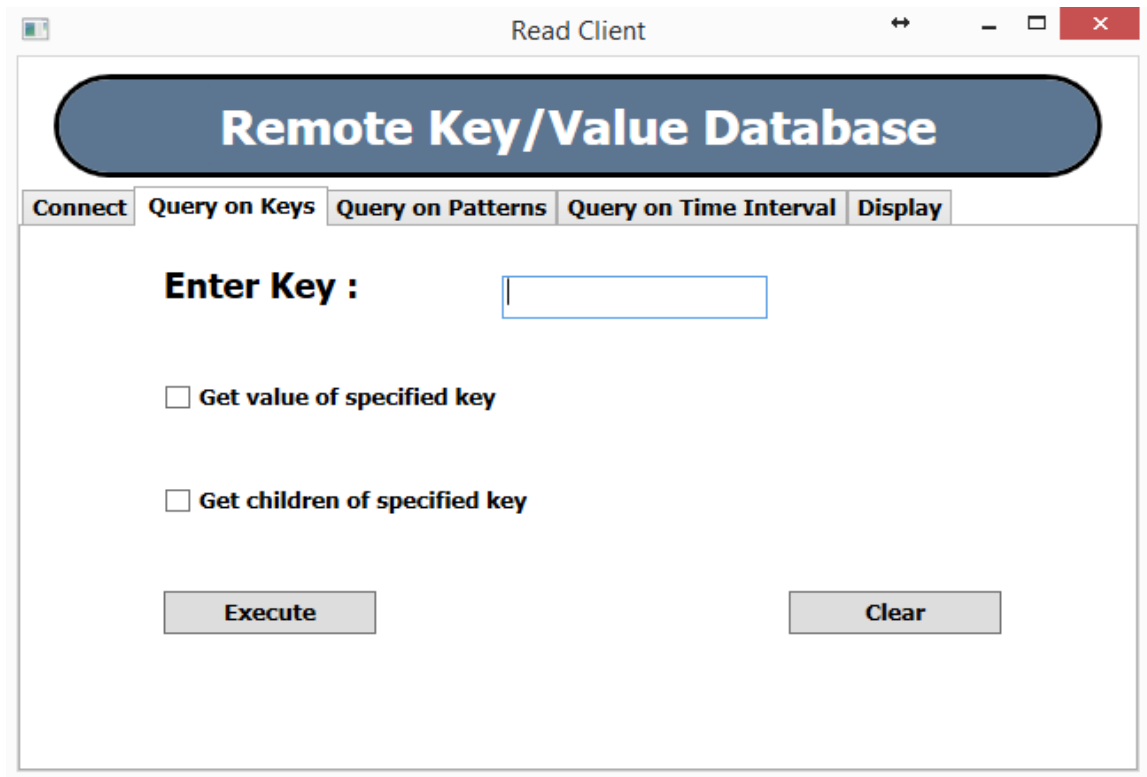
- Connect to Server
- Query on Keys
- Query on Patterns
- Query on Time Interval
- Display

1) Connect to Server



User enters the port number of the server that he wishes to connect with and clicks on the connect button. Connection gets established. When the user wishes to disconnect with the user , he can do so by clicking the disconnect button.

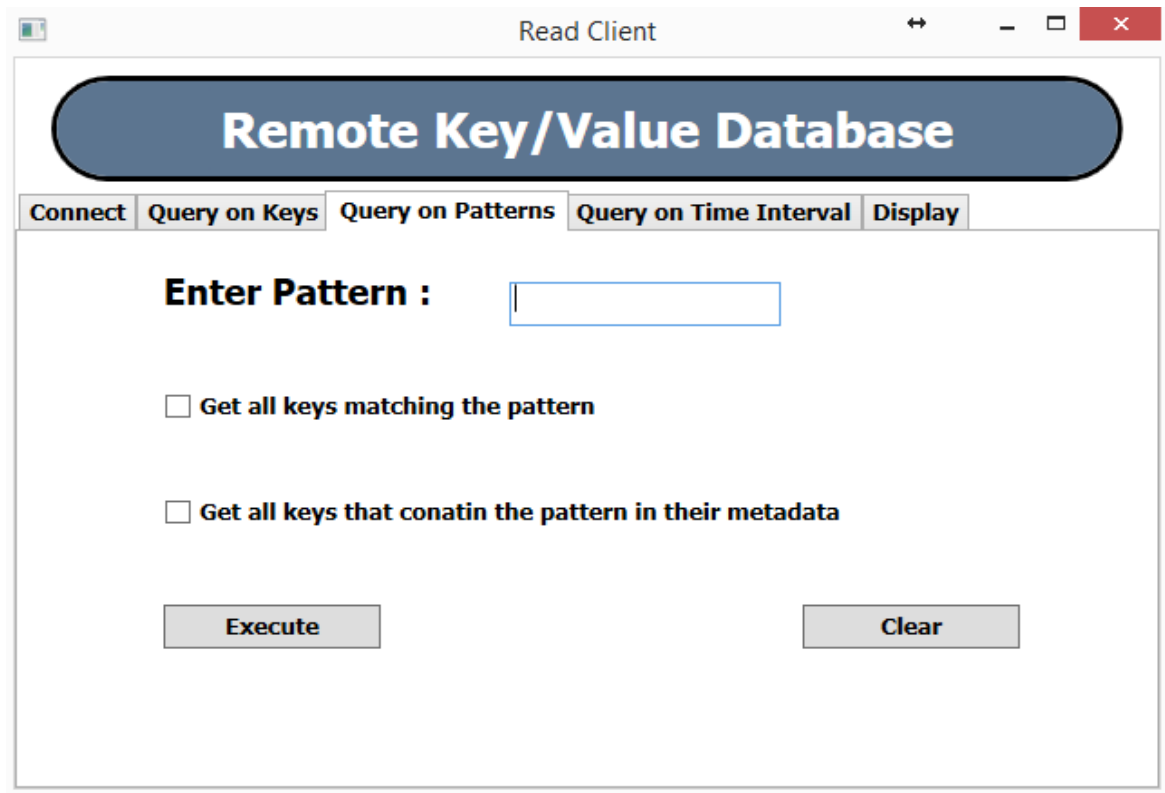
2) Query on keys



The screenshot shows a window titled "Read Client" with a dark blue header bar containing the text "Remote Key/Value Database". Below the header is a tabbed interface with five tabs: "Connect", "Query on Keys", "Query on Patterns", "Query on Time Interval", and "Display". The "Query on Keys" tab is currently selected. Inside this tab, there is a label "Enter Key :" followed by a text input field. Below the input field are two checkboxes: "Get value of specified key" and "Get children of specified key". At the bottom of the tab are two buttons: "Execute" and "Clear".

User enters the key and selects if he wishes to get the value of the specified key or the children of the specified key. And clicks on execute button to execute the command. The result of this query would be displayed on the Display tab. By clicking the clear button the entered value gets erased from the textbox.

3) Query on Patterns



The screenshot shows a web application window titled "Read Client". At the top, there is a blue rounded rectangle containing the text "Remote Key/Value Database". Below this, there are five tabs: "Connect", "Query on Keys", "Query on Patterns", "Query on Time Interval", and "Display". The "Query on Patterns" tab is currently selected. The main content area of the tab contains the following elements:

- A label "Enter Pattern :" followed by a text input field.
- Two checkboxes:
 - Get all keys matching the pattern
 - Get all keys that contain the pattern in their metadata
- Two buttons: "Execute" and "Clear".

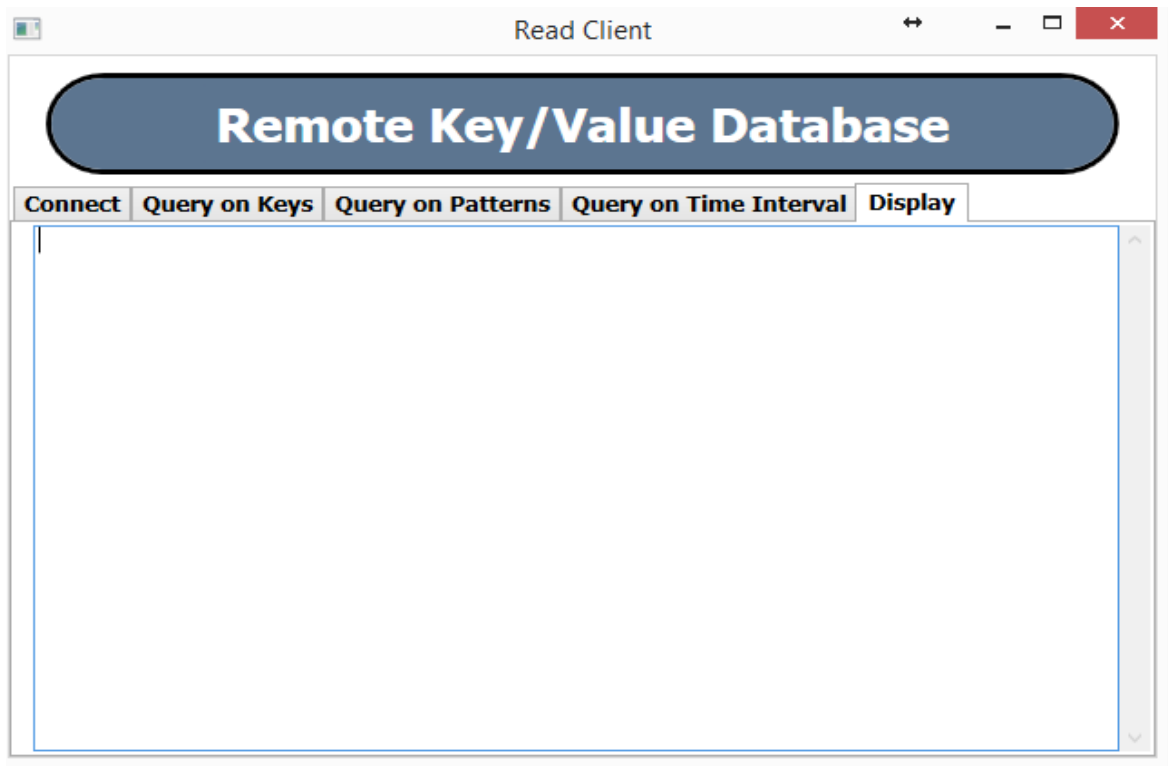
User enters the pattern and selects if he wishes to get all the keys matching the pattern or if he wishes to get all keys that contain the pattern in their metadata. Once the user enters the required data he can click on execute for the query to be executed on the database. The result of the query would be displayed on the display tab. By clicking the clear button the entered value gets erased from the textbox.

4) Query on Time Interval

The screenshot shows a web application window titled "Read Client". At the top, there is a dark blue rounded rectangle containing the text "Remote Key/Value Database". Below this, there is a horizontal menu with five tabs: "Connect", "Query on Keys", "Query on Patterns", "Query on Time Interval", and "Display". The "Query on Time Interval" tab is currently selected. Underneath the tabs, there are two text input fields. The first is labeled "Enter Start Time :" and the second is labeled "Enter End Time :". At the bottom of the form area, there are two buttons: "Execute" on the left and "Clear" on the right.

User enters start time and end time intervals for which the database is queried. Once the user provides the inputs and clicks the execute button. The request sent to the server where the query will be executed against the database and the result fetched will be displayed on the Display tab. The clear button erases the contents on the textbox.

5) Display

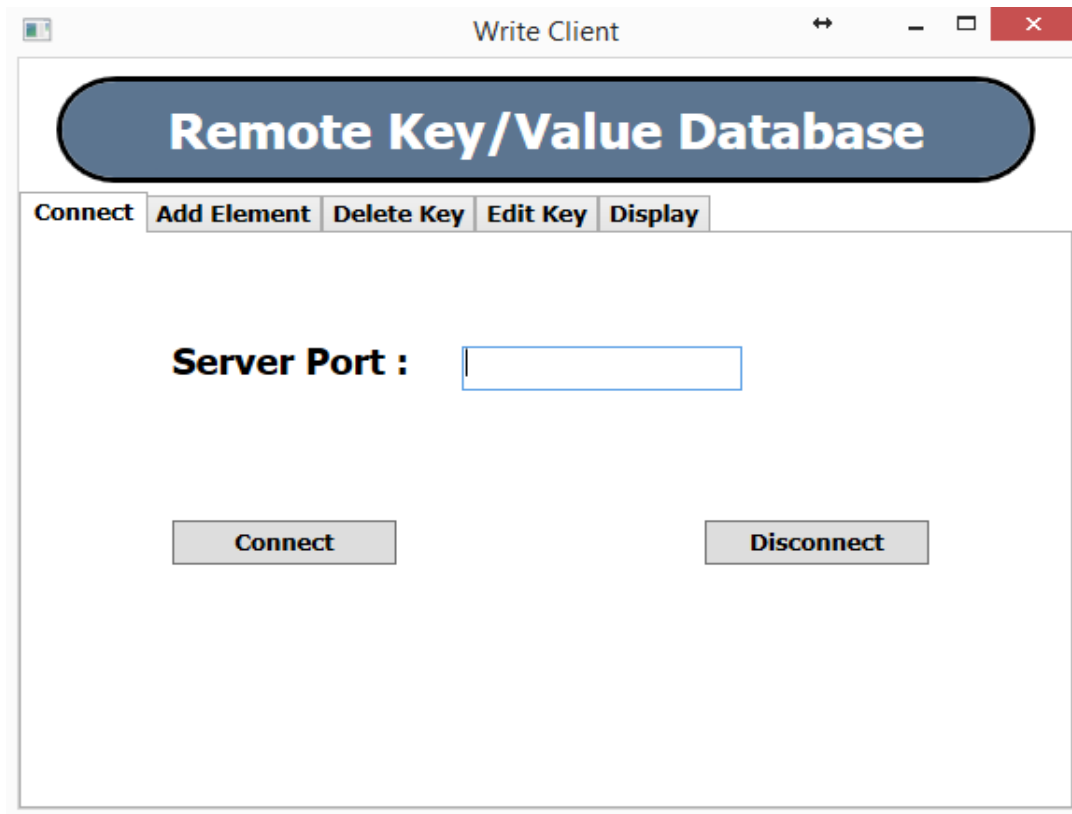


The display tab is used to display the results of every operation performed by the client. The client receives the output and displays it on the textbox of the Display tab.

3.2 Graphical user interface for Write Client contains the following panes:

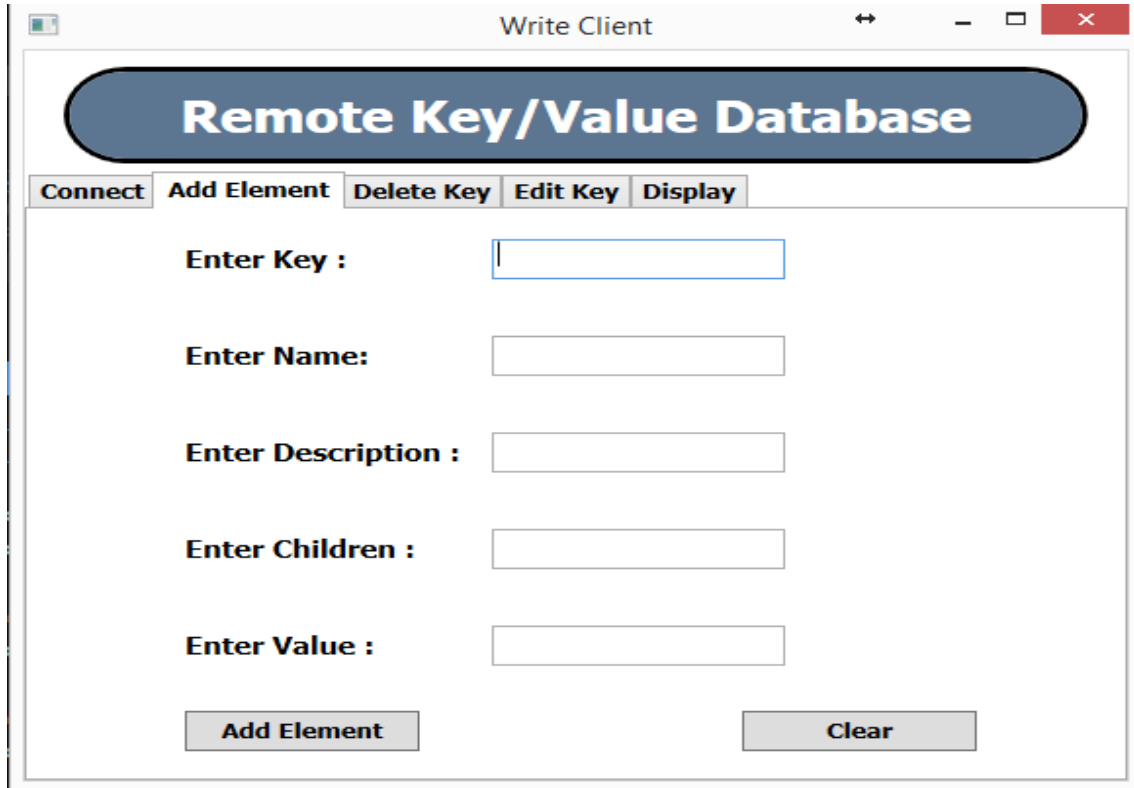
- Connect to Server
- Add Key
- Delete Key
- Edit Key
- Display

1) Connect to Server



User enters the port number of the server that he wishes to connect with and clicks on the connect button. Connection gets established. When the user wishes to disconnect with the user , he can do so by clicking the disconnect button.

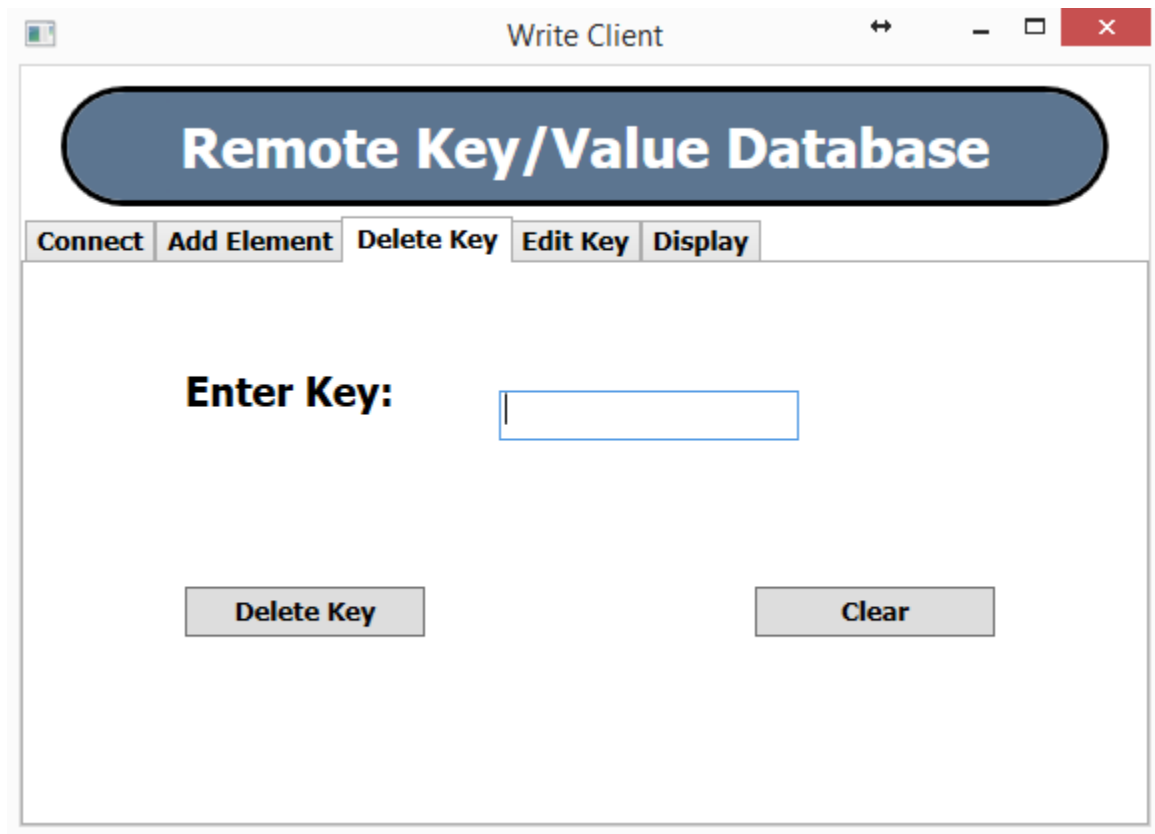
2) Add Key



The screenshot shows a web application window titled "Write Client" with a standard Windows-style title bar. The main content area features a dark blue rounded header with the text "Remote Key/Value Database". Below this header is a horizontal menu with five buttons: "Connect", "Add Element", "Delete Key", "Edit Key", and "Display". The "Add Element" button is currently selected. Underneath the menu, there are five text input fields, each preceded by a label: "Enter Key :", "Enter Name:", "Enter Description :", "Enter Children :", and "Enter Value :". At the bottom of the form area, there are two buttons: "Add Element" on the left and "Clear" on the right.

User enters Key and other details such as Name , Description, Children and Value(Payload) of the element to be added on to the database. Once the required information has been provided user clicks on Add Element, add request would be sent to the database. Clear button can be used to clear the fields.

3) Delete Key



The screenshot shows a web application window titled "Write Client". At the top, there is a dark blue rounded rectangle containing the text "Remote Key/Value Database". Below this, there is a horizontal menu with five buttons: "Connect", "Add Element", "Delete Key", "Edit Key", and "Display". The "Delete Key" button is currently selected. The main content area features the text "Enter Key:" followed by an empty text input field. Below the input field, there are two buttons: "Delete Key" and "Clear".

User enters the key, to be deleted from the database. Once the user enters the key to be deleted and clicks on Delete Key button the delete request would be sent to the server. Clear button clears the text box.

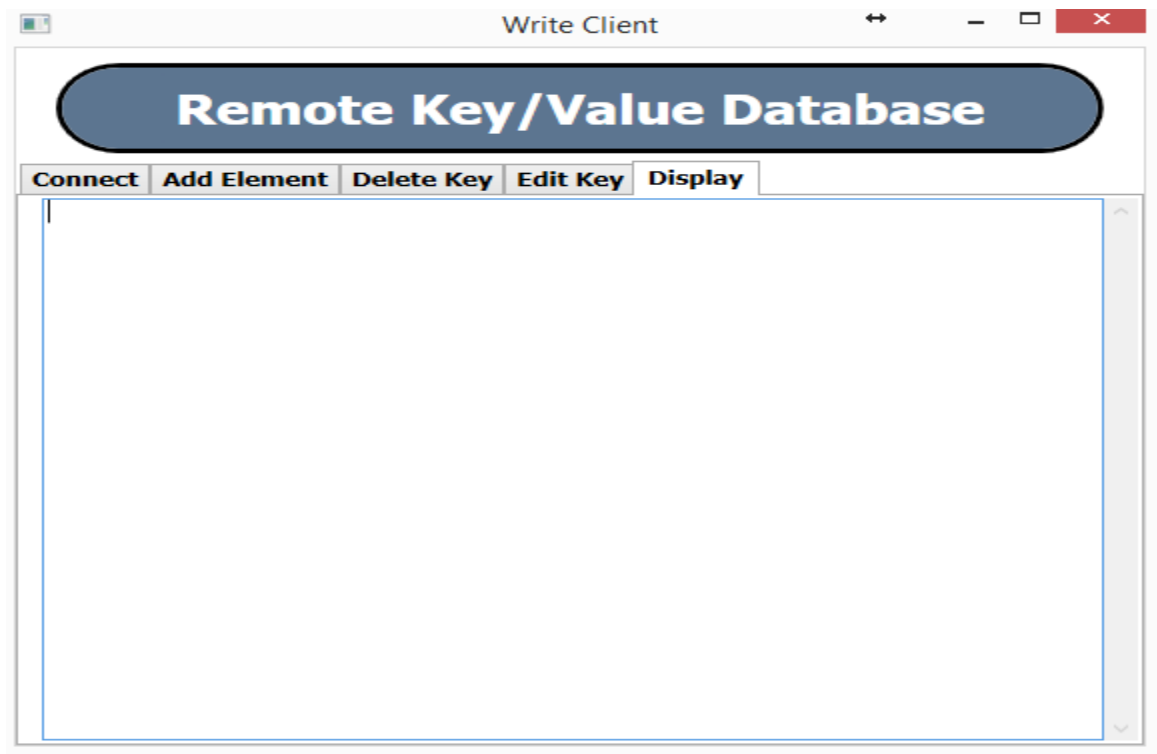
4) Edit Key

The screenshot shows a web application window titled "Write Client". At the top, there is a dark blue rounded rectangle with the text "Remote Key/Value Database" in white. Below this, there is a navigation bar with five tabs: "Connect", "Add Element", "Delete Key", "Edit Key", and "Display". The "Edit Key" tab is currently selected. The main content area contains a form with the following elements:

- A label "Enter Key :" followed by a text input field.
- A checkbox labeled "Enter Metadata" followed by a text input field.
- A checkbox labeled "Enter Value" followed by a text input field.
- A checkbox labeled "Add Relationship" followed by a text input field.
- A checkbox labeled "Delete Relationship" followed by a text input field.
- At the bottom, there are two buttons: "Edit Element" and "Clear".

User enters the key that has be edited and checks the checkbox along with the data that has to be edited. Note that only one of the checkboxes can be checked at any point of time. Once the relevant information has been provided and the edit element button has been clicked the edit request would be sent to the server. Clear button clears the fields.

5) Display



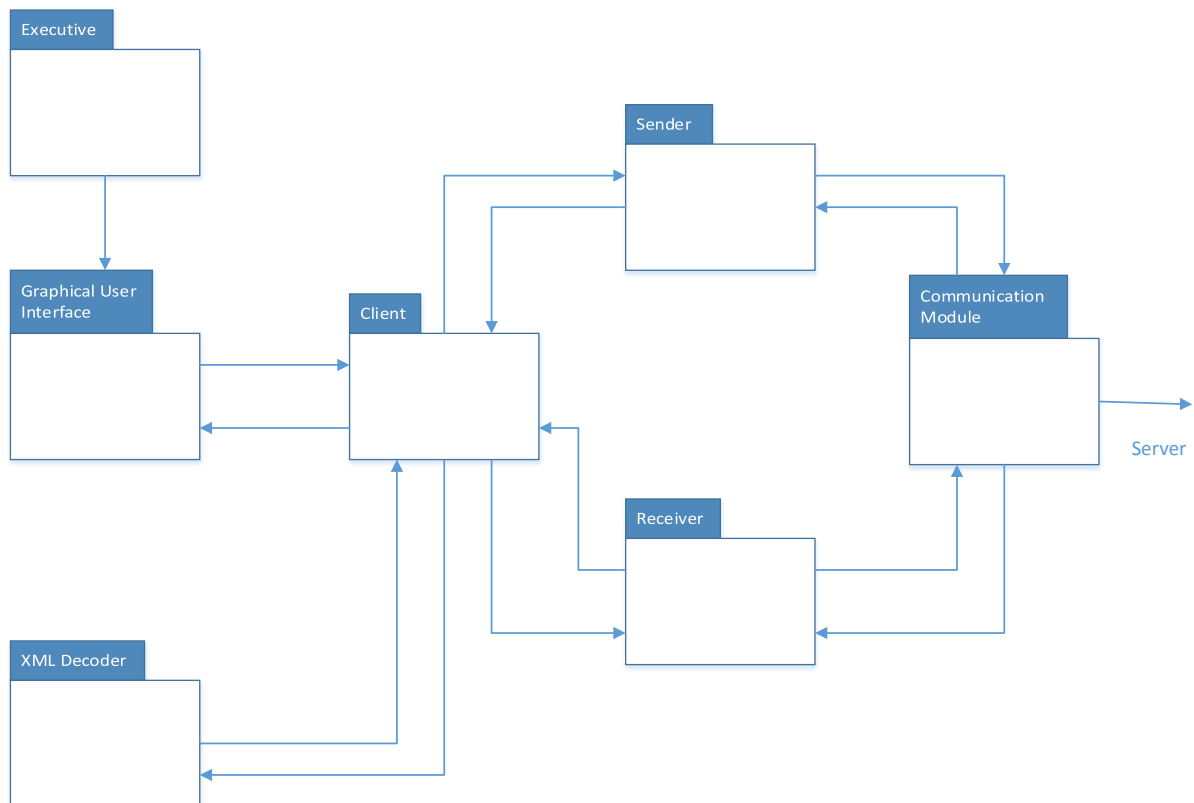
The display tab is used to display the results of every operation performed by the client. The client receives the output and displays it on the textbox of the Display tab.

4. Remote Key/Value Database Partition

Remote Key/Value Database is based on client server model. It can be divided into 2 packages:

1. Client Package
2. Server Package

4.1 Remote Key/Value Database Client Package



The above diagram will help us understand how the packages are dependent on each other to make up a client model.

Tasks performed by the client package:

- 1) Provides Graphical User Interface.
- 2) Sends request provided by user to the server.
- 3) Accepts the response from the server.
- 4) Display the output on the GUI.

Remote Key/Value Database Client Package consists of the following packages:

1) Executive

Executive package is the main module of the Remote Key/Value Database client. This package integrates the client model. The main execution point of the client starts from executive. The graphical user interface would be loaded by the executive, through which user interacts with the system.

2) Graphical User Interface

Graphical user interface package is the point where user communicates with the system. It takes input from the user and displays the response. As discussed in the above section, the Read client GUI provides features to query the database on the server. This package then sends the user entered information to the client package which in turn is packed in to a message that is sent to the server.

It will contain many tabbed panes on the main window which helps the user to navigate through various available options. Each pane is discussed in the view section of the document.

3) Client

Client package plays a vital role in the model. It handles all the communication between graphical user interface, xml decoder and communication module. Whenever user selects queries through the GUI for Remote Key/Value Database, the request is received by the client package. Client then sends this request to communication module which encodes the request into a message and sends it to the server.

When the response is received from the server, the decoded response is sent to the client through communication module. After receiving the response, the result is displayed on the graphical user interface.

4) XML Decoder

The inputs provided to the clients are XML files. XML decoder will take file and decode it and interprets the operation to be performed and sends the operation request to the client.

5) Sender

Sender package is used whenever client wants to send request to the server. We can have multiple requests to send, so we need to maintain a queue which will store all the requests. Client pushes all the requests on to the sender queue of the client

.Communication package will take requests, one by one from the queue and sends the requests to the server it has established a connection with.

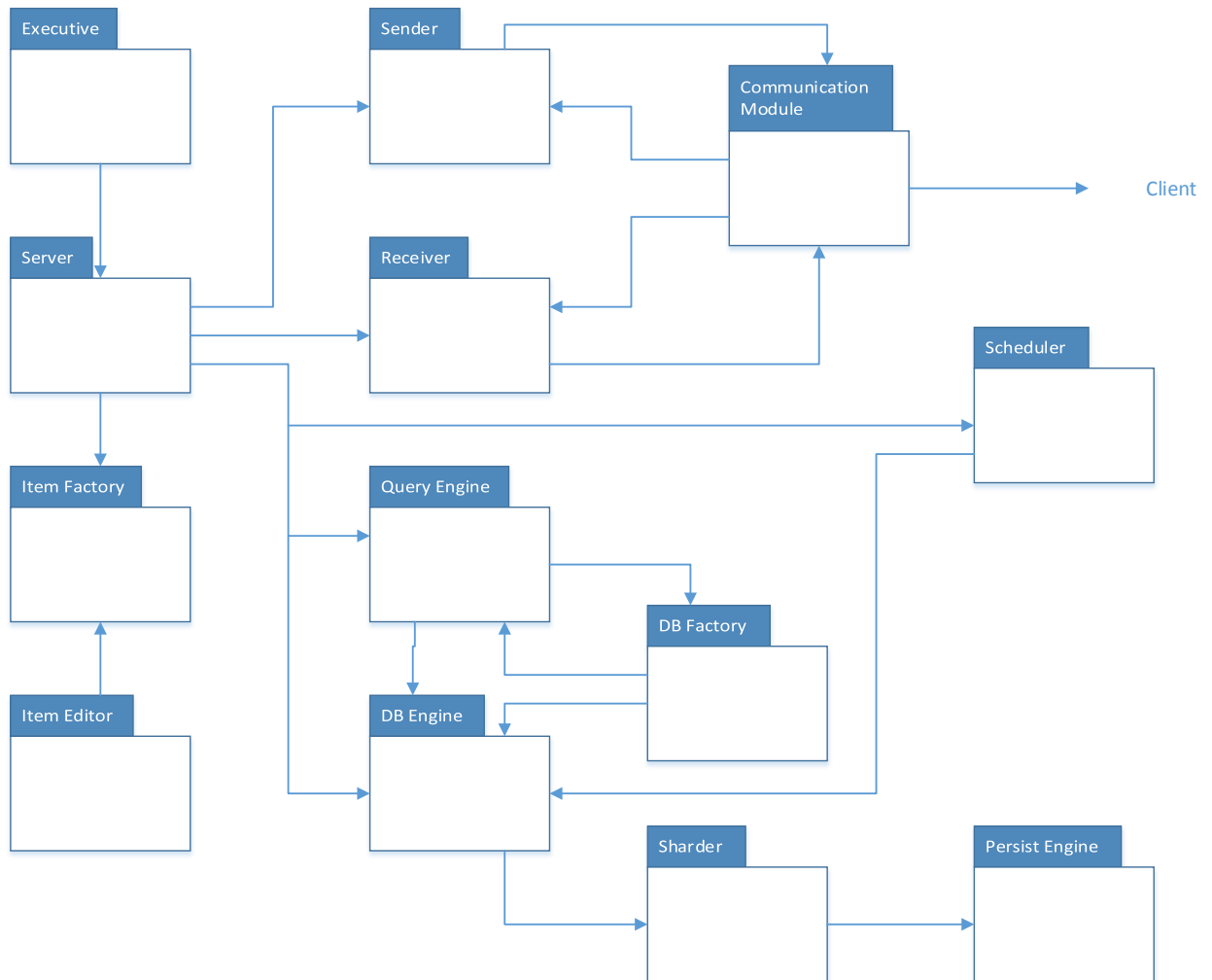
6) Receiver

Receiver package is used when a response is received for a given request. We can have multiple responses, so a queue is maintained to receive multiple responses from the server. Client then takes those responses from the queue and displays the result on to the GUI.

7) Communication Module

This package is used to handle communication between client and server. It helps to create channel between client and server through which client can interact with the server. When a request is received from the client, it encodes the message and sends to the server. Similarly, when the response is received for a given request, it will decode the message and gives it to the client.

4.2 Remote Key/Value Database Server Package



The above diagram will help us understand how the packages are dependent on each other to make up a server model.

Tasks performed by the server module:

- 1) Accepts requests from the client.
- 2) Responsible for all database operations stored on the server.
- 3) Based on the operation requested by the client, server will perform the read or write operations on the database and sends back the response to the respective client.

Remote Key/Value Database Server Module consist of the following packages:

1) Executive

It starts the execution of the server. It contains the main method for the execution of the server.

2) Server

It is the most important package in the server module. The communication between the client and server is handled using the communication module. When it receives a request from the client, it determines the operation to be performed and hands over the request to the database package. It is also responsible for sending response back to the client for a given request.

Server persist the database contents on to an XML file based on the input provided by the client.

3) Sender

Sender package is responsible for sending response to the client for a given request. Since we will have multiple requests coming in, there will be multiple responses. Hence all the responses are stored on the sender queue. Whenever a request is processed it will store the response in the queue. Sender then dequeues the queue and encodes the response to a message format and sends the response to corresponding client using the communication channel.

4) Receiver

Receiver package is responsible for receiving the request. We can receive multiple requests at the same time, we will queue all the requests. Multiple child threads can be created to handle the request overhead on the queue and to ensure performance efficiency. The receiver accepts request from the queue and send it to the database package for processing the request.

5) Item Factory

Item Factory package is responsible for creating instances of Key/Value pairs. Using these instances one can generate unique key and construct a value with supplied parameters. Item factory interacts with executive package in order to pass the key/value pairs created to the DB Engine that inserts them.

6) Item Editor

Item Editor package is responsible for editing key/values. Executive provides a copy of the item to be edited to the Item Editor. After performing the operation, the item is passed back to the DB Engine for insertion.

7) Query Engine

Query Engine package is responsible for the query operations performed on the database. This package plays the role of submitting queries to the database, executing query and also provide the query output.

8) DB Engine

DB Engine package is responsible for populating the key/value instances into the database instance created by the DB Factory package.

9) DB Factory

DB Factory package is responsible for generating key/value database instance that is capable of storing the key/value pairs. It interacts with the Query Engine so that immutable databases can be created based on the queries executed.

10) Scheduler

Scheduler package is responsible for accepting positive time interval or number of writes using which the database would be scheduled to save the data contents at regular intervals.

11) Persist Engine

Persist package is responsible for writing the database on to an XML file. This process should be reversible such that the data in the XML file can be restored on to a database.

12) Sharder

Sharder package is used to partition the database into smaller chunks. Thus enabling performance to be improved on huge data as it would be easier to perform operations on relevant data than the entire database.

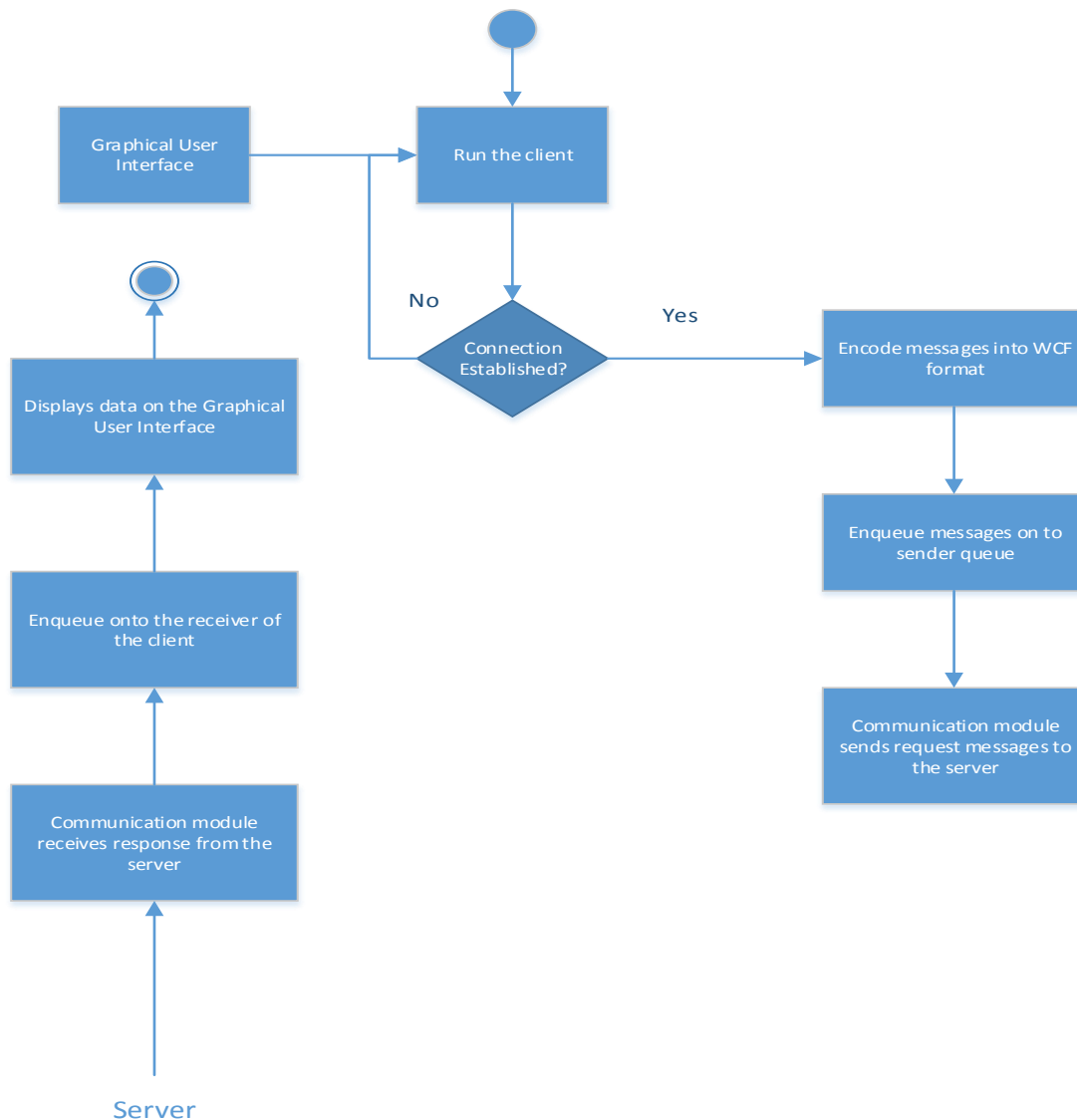
13) Communication Module

This package is used to handle communication between client and server. It helps to create channel between client and server through which client can interact with the server. When a request is received from the client, it decodes the message and sends it to the server. Similarly, when the response is received for a given request, it will encode the response and gives it to the client.

5. Activity Diagram

This section discusses the main activities associated with Remote Key/Value Database client and server. Activity diagram are divided into 2 parts for the better understanding of the system.

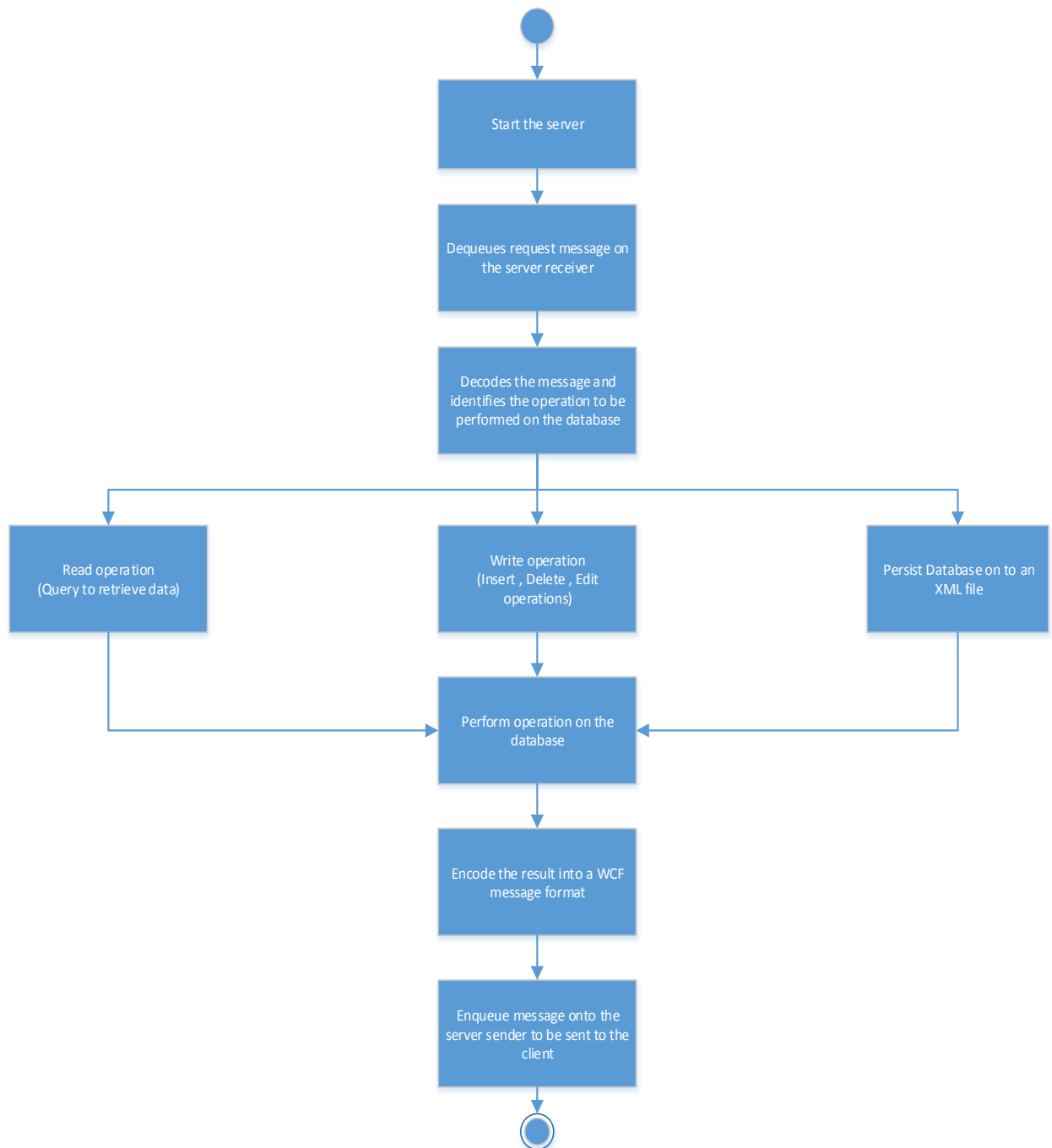
5.1 Remote Key/Value Database Client Activities



- 1) Client receives the server port number from the user and establishes the connection. Once the connection has been successfully established , client can perform other operations over the database.

- 2) Client receives the operation request from the user through the GUI and encodes it into a WCF message. This request message is later enqueued on to the Client sender queue.
- 3) Based on the operation requested by the user, Client can be divided into read and write client. Though the general activities performed by both these clients are the same, operation performed on the database are different.
- 4) Communication module dequeues the messages from the sender queue and sends it to the server.
- 5) Once the server sends back the response , the communication module enqueues the responses on to the client receiver.
- 6) The client then dequeues these messages and displays the result on the display tab of the GUI.
- 7) In case of Write clients, that do not make use of GUIs an acknowledgement stating the status of the operation, such as “Element successfully inserted !!!” will be sent back that can be displayed on the console.

5.2 Remote Key/Value Database Server Activities



- 1) Once the server is started, and the connection is established with the client successfully. The communication module receives the request and enqueues on to the server receiver.

- 2) The server then dequeues these requests and processes them to interpret the operation to be performed on the database.
- 3) Once the operation to be performed is identified , the server then performs either of the 3 operations on the database-
 - Read – querying the database based on the input provided by the user.
 - Write – insert/delete/edit operations on the elements of the database.
 - Persist – persists the contents of the database on to an XML file.
- 4) The result/status of the operation is then encoded in to a message that is sent to the client via the communication module.

6. Critical Issues

1) Server is down

Suppose the server on which the database is stored, is down due to network problems etc. then the clients cannot access the database. The entire operations are shut down as the database is centralized over the server.

Solution: One way to tackle this issue is to take a backup of the database on a backup server. This will ensure smooth processing of the request received from the client and the operations are not stalled .

2) Connection terminates between client and server

Remote Key/Value Database is a network based client server application. There are chances that the communication between client and server is terminated while sending or receiving request/response. Suppose while sending the response to the client for a request, if the connection is lost during the time. This will result in incomplete response.

Solution: We can send acknowledgement message saying that the request or the response is received. Another possible solution is if the connection terminates between client and server, we can have separate threads to establish connection after certain delay. This delay can be configured at the time of implementation.

3) Performance

Suppose there is large number of requests in the queue waiting to be processed. If a single query has large number of data to be processed on the database then eventually it will take long time to process the next request. We process the request sequentially it leads to bad performance because client will have to wait for long in order to receive the response.

Solution: We can have child threads to process the multiple requests in the queue. If we have many requests in the queue, child threads takes request simultaneously and starts processing. In this case, no request will have to wait in the queue for its turn to come. Parallel execution of the request will give good performance compared to sequential execution.

4) More than one user tries to write on the same element simultaneously

Suppose more than one client are accessing the same element on the database and trying to perform an edit operation on the element. This might end up in data inconsistency issues. As the information stored at the end of both the writes might be erroneous.

Solution: When a write client accesses the database we can incorporate some sort of lock on the database. So that no other write operations takes place. Once the user has completed with the operation, the lock would be disabled . Allowing other users to access the data. But this can result in a performance issue.

7. Conclusion

Remote Key/Value Database as the name suggests helps access the noSQL database that is set up remotely over the server via clients. This document describes the architecture, various packages and activities involved while implementing the system. We have also taken care of the critical issues by providing solutions so that the implementation does not contain any flaws.

It provides the below features:

- Graphical user interface for the user to interact with the system
- Option to select server by providing the port number
- It performs user requested operations on the database.
- Displays the generated result on the GUI.

This tool is very helpful in large software organization where the user does not have to be present personally at database location in order to access it. This tool enables the database to be set up in a centralized environment thus enabling the clients to establish connections and perform database operations remotely.

8. References

- 1) <http://ecs.syr.edu/faculty/fawcett/handouts/webpages/BlogNoSql.htm>
- 2) <http://ecs.syr.edu/faculty/fawcett/handouts/CSE681/lectures/Project3-F2015.htm>
- 3) <http://ecs.syr.edu/faculty/fawcett/handouts/CSE681/lectures/Project4-F2015.htm>