ENTS 640
NETWORKSAND PROTOCOLSI

PROJECT
FALL 2016

Alekhya Dixit, Pulavarthy

UID:115033085

Apeksha Chauhan

UID: 115035027

OVERVIEW/ ABSTRACT

A code for a java application comprising of a client and a server that use Java's UDP sockets to send and receive messages is provided. The code also implementsintegritycheck, timeout andretransmission operations to ensure reliable communicationwhile using unreliable data transfer services. The server application is to respond to the client's request of a particular measurement ID by sending the corresponding temperature value as stored in the text file.

PROBLEM DEFINITION

Client Program Objectives:

1. To select a random measurement ID from the list of data given in the data.txt file.
2. To generate a random 16-bit unsigned digit which is to be used as a request IDfor an assembled request message being sent from client to server.
3. The request message needs to be converted into bytes for transmission through UDP sockets.
4. To integrate a timer function that waits for the response message of the server with respect to the request message sent (append checksum value) and implements appropriate error messages and timeout operations.
5. To carry out integrity check operation that calculates checksum for the response message received from the server and compares it with the actual checksum value integrated in the response message.
    - If no errors, then the program should print the corresponding temperature value received from the server.
    - If errors are detected, the client should respond by printing the appropriate error message. (Response code)
6. The client keeps running i.e., does the complete request/response sequence until the application is terminated.

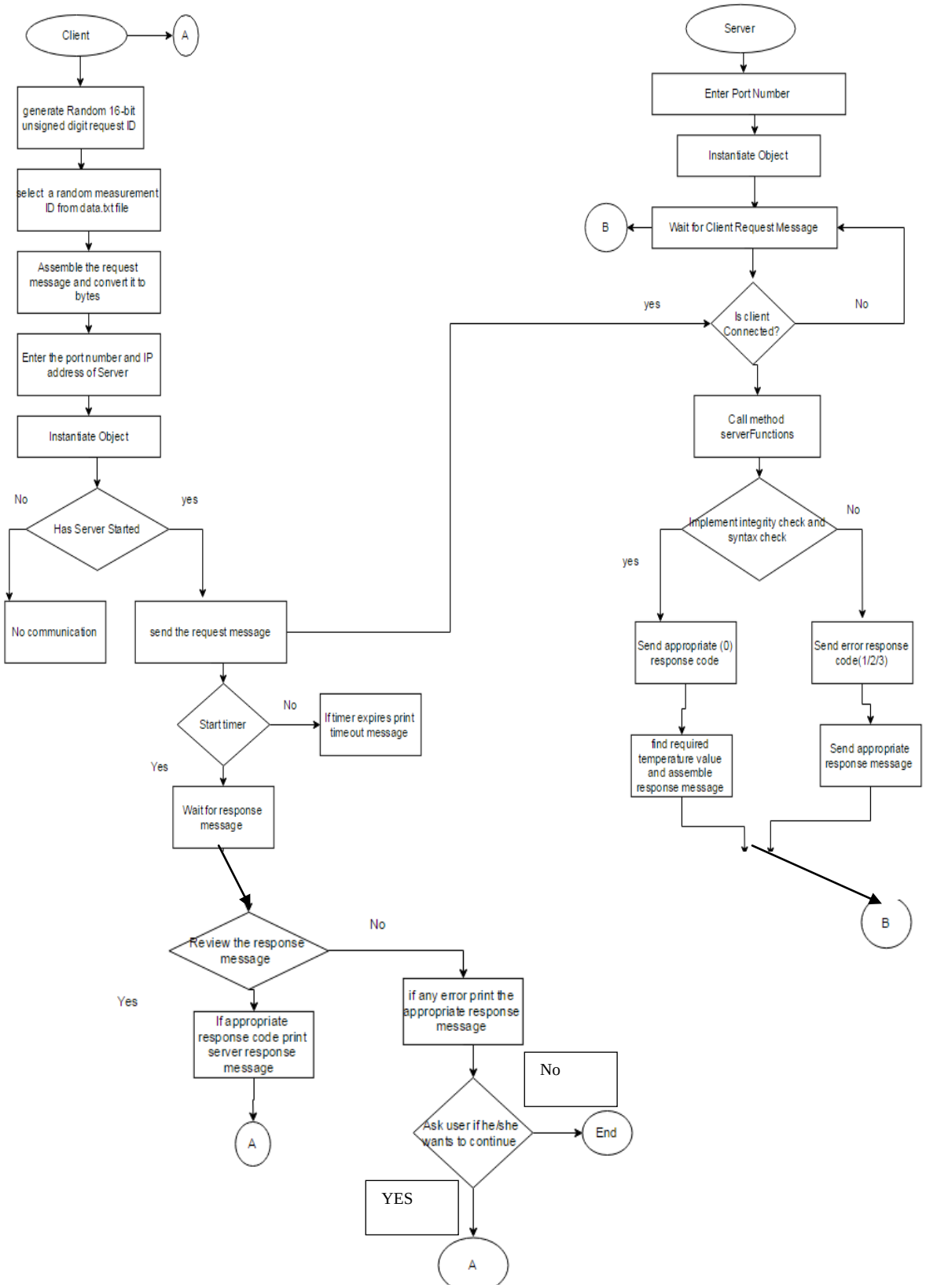
Server Program Objectives:

1. The server application waits for the request message from the client.
2. Consequent to the reception a request message from the client application, the server needs to implement the integrity check function to calculate the checksum value of the received request message and then compare it with checksum value appended to the request message.

- If the syntax checks fail, it should send back an error response with response code 2.
- If the integrity check operation passes, it should convert the received message into a character sequence (a String object) and check the syntax of the request message thoroughly

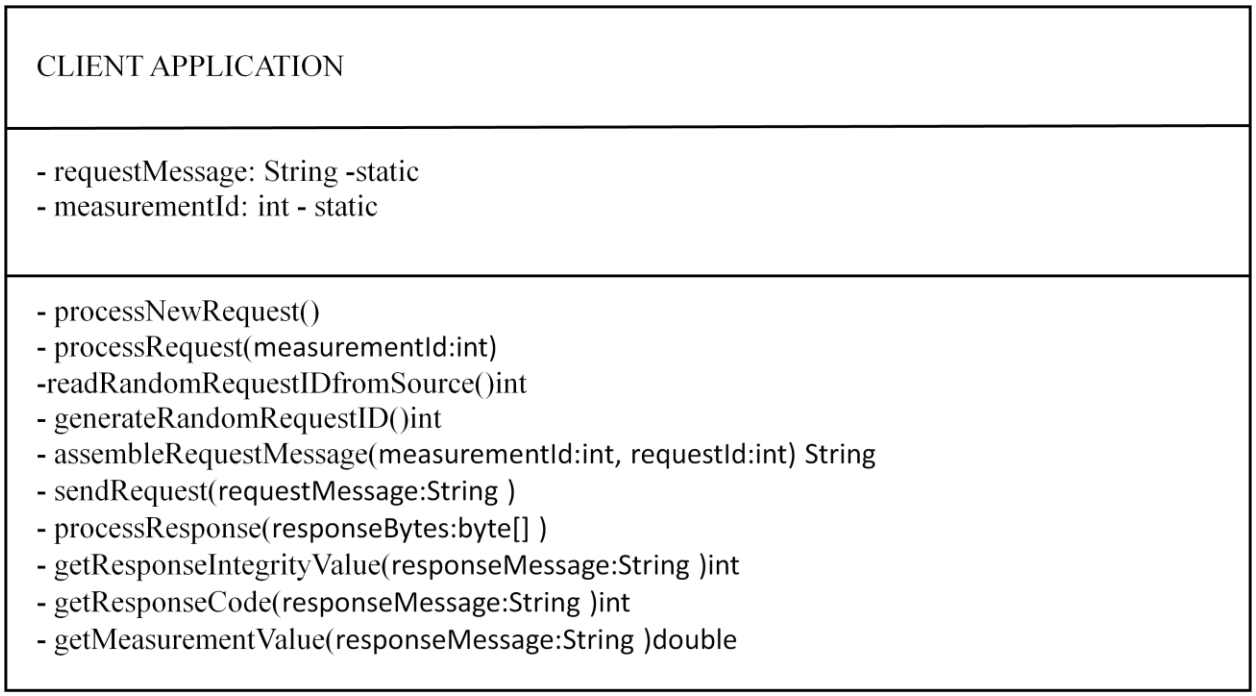
3. The server should locate the measurement value for the requested measurement ID. If there is no measurement value for the detected request, then it should return an error response with response code 3.
4. If there is no error, then the server application is to assemble the response message with response code 0 including measurement value and the Integrity check sum value.
5. The assembled response message is to be converted to a byte array suitable for transmitting through UDP socket to client application.
6. The server application needs to keep running until termination.

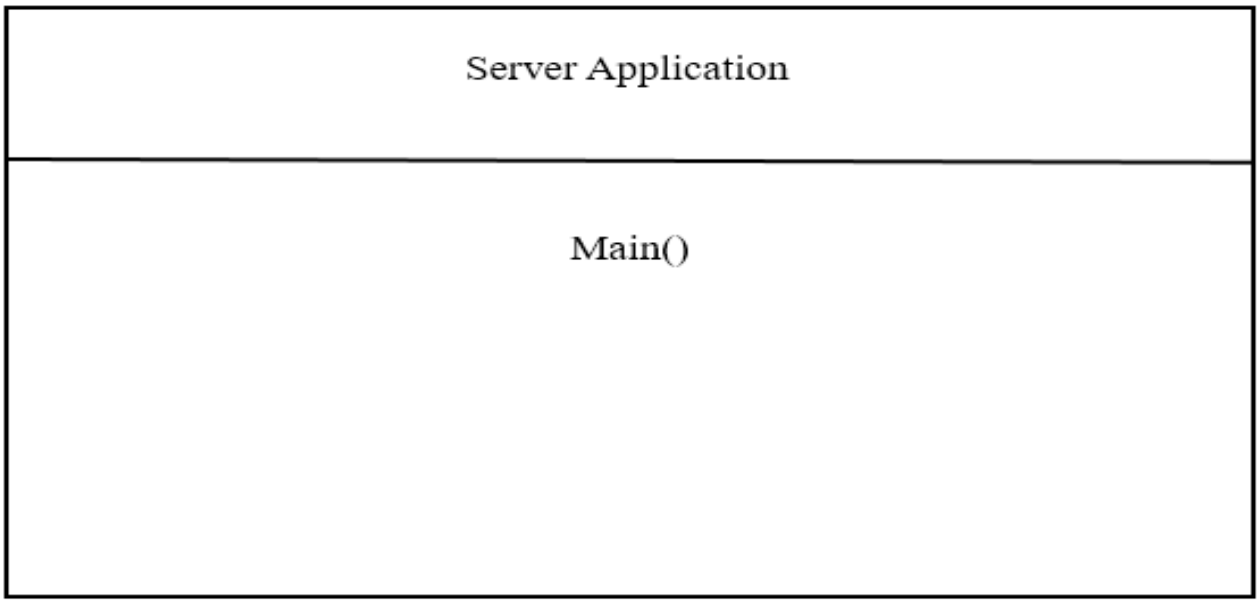
FLOWCHART FOR APPLICATION:

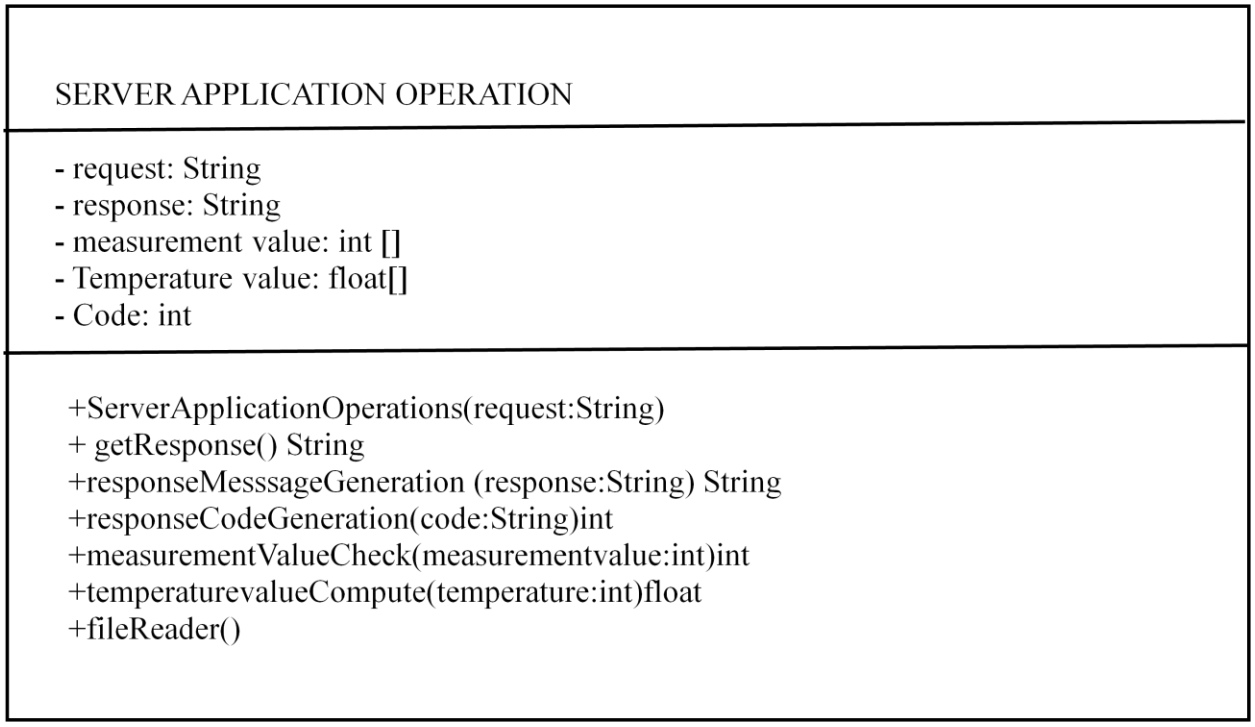


UML DIAGRAMS

The Client and Server application programs various objectives have been realized by defining different methods in their classes. This practice aided in integrating various functions of these applications effectively and efficiently.







CLIENT AND SERVER APPLICATION DESIGN AND ANALYSIS

Client Objectives Analysis:

1. Main Function: call processNewRequest :
    - calls the function "readRandomRequestIDfromSource"- i.e, to read a request_ID
    - calls processrequest with measurementID as parameter.
        - (a) Function: processRequest
            - Generate random requestID
            - Assemble the request message(without including integrity check)
            - Using StringBuilder to add checksum value to the assembled request message.
            - Ignoring the spaces by replacing with "null" = requestMessage
        - Send requestMessage to the server

2. Function: readRandomRequestIDfromSource
    - Using Buffer Reader and Arraylist properties: read and split the line to extract measurementIDs (store measurement IDs in an array List)
    - Use random function to choose one of the measurementIDs.
    - Select a random measurement ID from the array list(limited by its size) using random function and store as String

- Convert measurementID from String to Integer and ID return it.

3. Function: generate RandomRequest ID(16-bit unsigned number range-(0-65535))
    - Use Random Function
    - Store randomly generated RequestID as “char” – typecast .
    - Return the randomly generated requestID number

4. Function: assembleRequestMessage
    - Assembling the request messsage using String Builder and using the variables requestID and mesaurement ID .
    - Return request message(will not have checksum yet)

5. Function: sendRequest
    - Convert request message appended with checkSum to bytes
    - Initialise InetAddress and assign a port number to send the request message to the server
    - Start Timer and wait for response message from server
        - Set initial timeout to 1000ms(1s)
        - And initialise a retry counter to 1 (retry 3 times if the first time fails. And print according timeout messages)
        - Double the timeout interval after every failed attempt
        - After 4th attempt transmitter i.e the source should declare communication failure and print an error message when there is no response rom server.
    - Initialise required variables to receive the response of the server. Store in “responseBytes”.
    - Close Socket.
6. Function: processResponseBytes – This function is to evaluate the integrity check for the response message , and compare it with the value supplied in the response message. If not equal it should start again from creating a random requestID.
    - Store the responseMessage in a String.
    - Calculate the integrity check value for the response message by passing the string through the integrityCheck function.
    - (a) Create a function : getResponseIntegrityValue
        - Extract the checksum value from integrity response

- If expected and actual checkSum values are not equal, call “processRequest” for retransmission
- (b) Create a function: getResponseCode
    - Extract the response code
- (c) Create function: getMeasurementValue : Extract the measurement Value
- Check the value of the response code and print appropriate message
- If response code is not equal to zero, then print a message indicating the particular error and then ask the user if he/she would like to resend the request if needed.
    - Initialise userChoice variable. Read user Input with Scanner function and respond accordingly.

Server Objectives Analysis:

Server Application Program:

1. The Server Application program has one main () function that initializes a variable to receive the request message and keeps waiting for the request message from client.
2. The received data is converted to a string format and passes the string to the Server Functions program that performs required operations.

Sever Application Operations Program:

3. A constructor is defined to receive the request message and is passed to the responseMessageGeneration method.
4. Method: responsecodeGeneration(String request)
    - This method breaks the request message received from the string into <request>, <Id>, </id>, <measurement>, </measurement>, measurement value and checksum value.
    - The function computes the checksum value for the request message (without checksum value) and compare this value to the one provided in the request string. If the integrity check passes a thorough syntax check of the received message and passed to the corresponding responseCodeGeneration method.
    - The method checks whether the measurement value is present in the data by calling the method measurementValueCheck(measurementValue).
        - Method: measurementValueCheck(int value): This method checks if the measurement value is present in the data.txt file or not. If the data is present it returns the corresponding measurement value. If the data is not found, then an appropriate error message is printed.

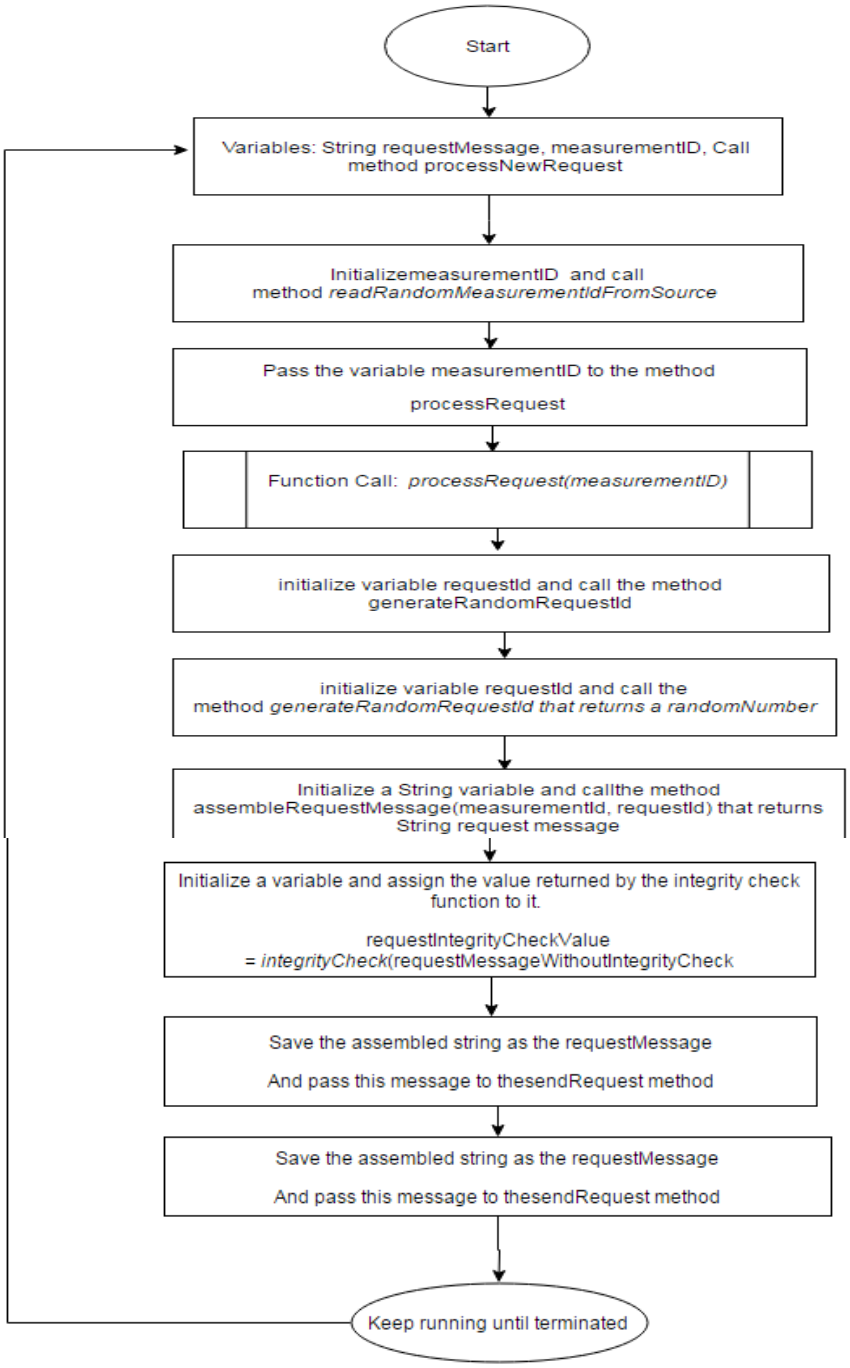
5. Method: responseMessageGeneration performs the following operations:
    - It divides the received request message to extract the “requestID”, “measurementID”, “measurement value/ temperature value” and the “checksum value” from the request string.
    - It then passes the request message to the “responseCodeGeneration” method that returns the appropriate response code.
    - It compares the response code to the appropriate values (0/1/2/3) and generates the response message according to the table given below:

Response code	Meaning
0	OK.The response has been created according to the request.
1	Error:integritycheckfailure. The request has one or more bit errors.
2	Error:malformed request.The syntax of the request message is not correct.
3	Error:non-existent measurement. The measurement with the requested measurementID does not exist.

        - For a responseCode of zero, the method passes the measurement value/ temperature value retrieved from the method temperatureValueCompute.
            - Method: temperaturevalueCompute : For responseCode = 0 the method passes the measurement value retrieved from the request totemperatureValueCompute(measurementValueInt)function which returns the value corresponding to the measurement id.
    - For any other code value generates a response with the responseCode value only.

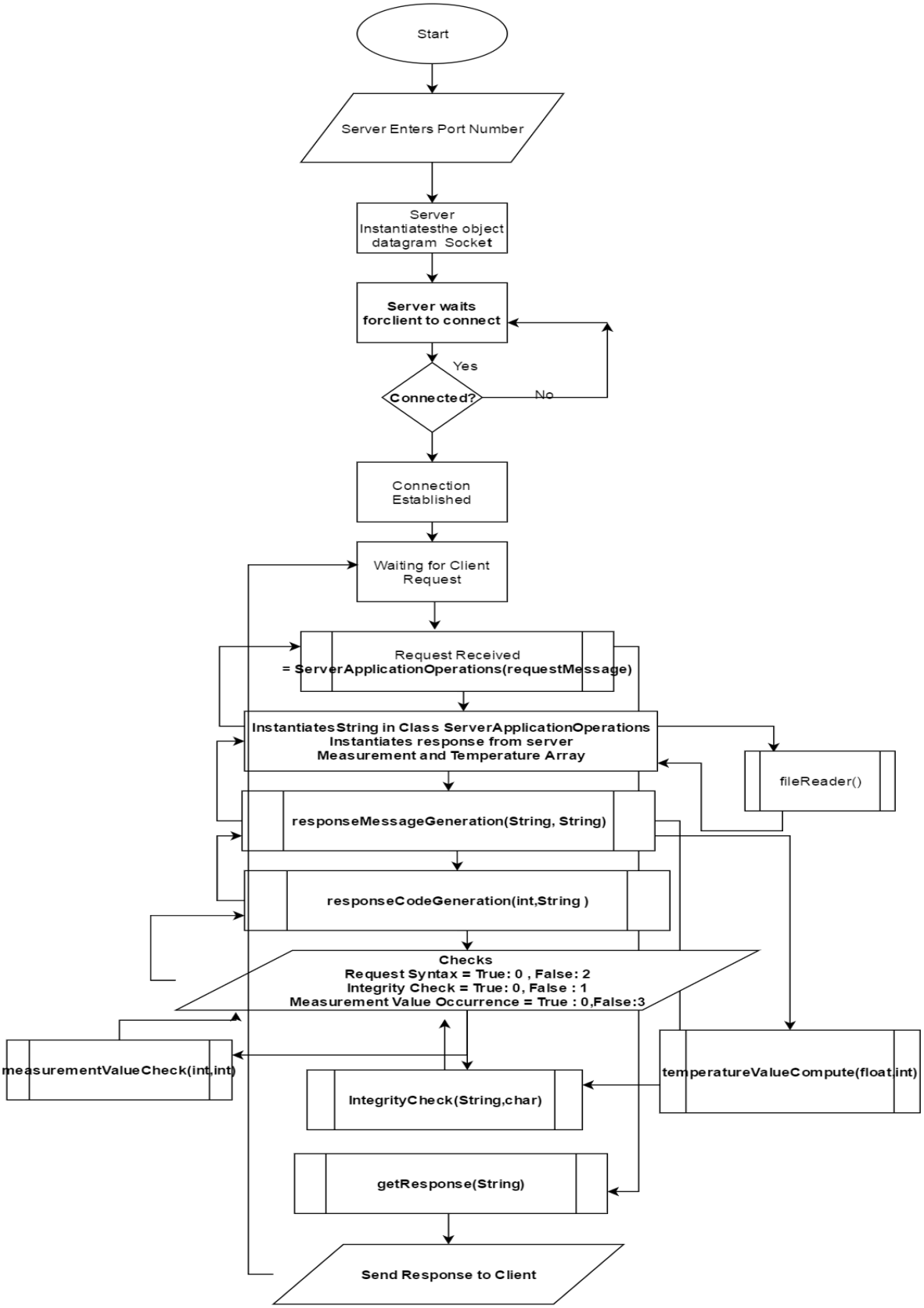
6. Other Methods Employed:
    - measurementValueCheck: This method returns an integer array containing all the measurement values.
    - temperatureValueCompute: This method computes the temperature value corresponding to the measurementID.

CLIENT APPLICATION FLOWCHART:

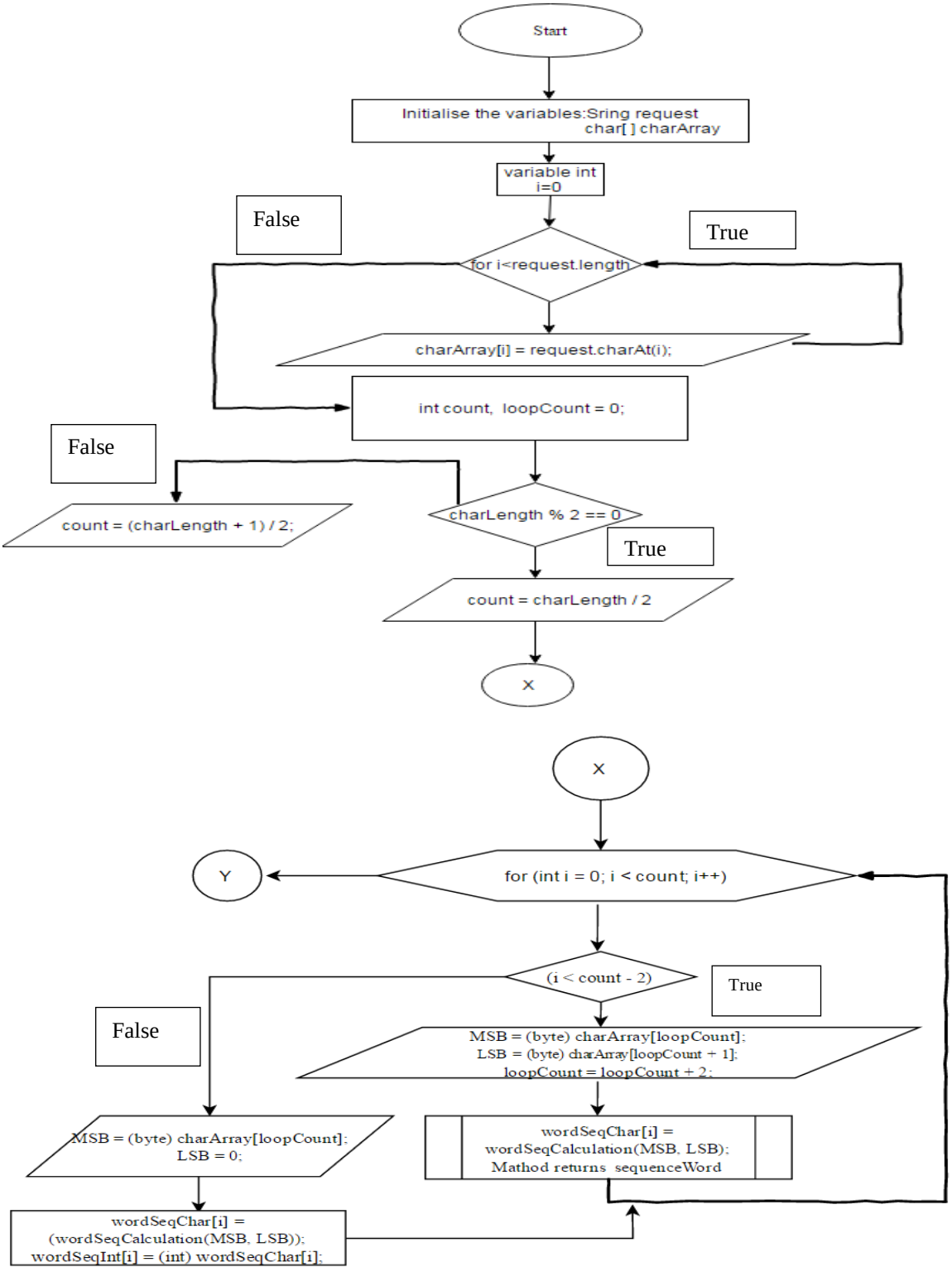


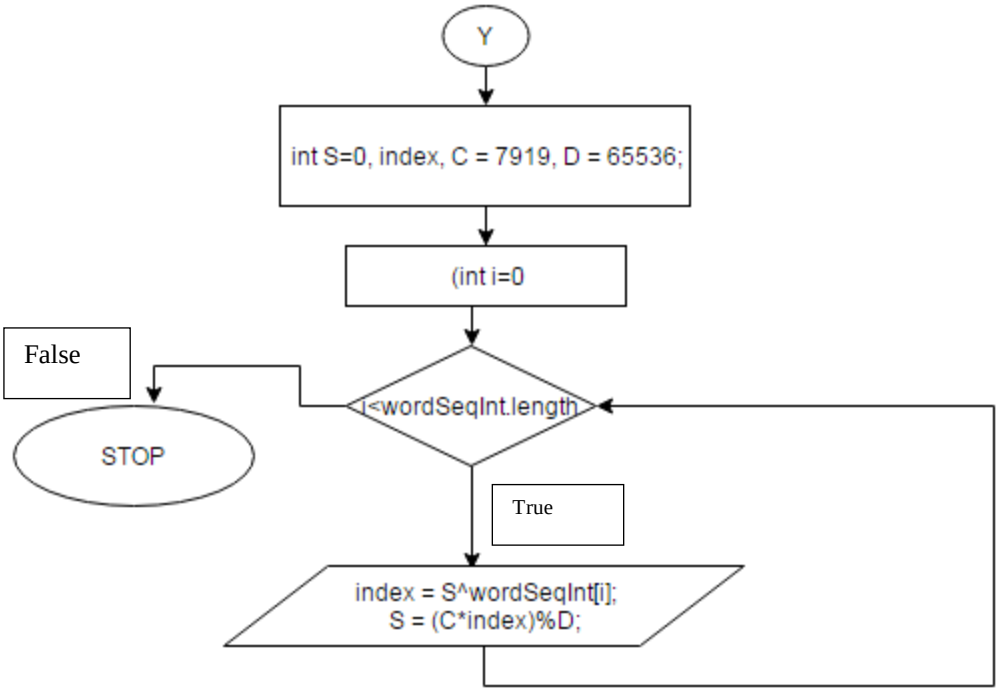
*Refer Client analysis for detailed method implementations

SERVER APPLICATION FLOWCHART



INTEGRITY CHECK FUNCTION FLOWCHART:



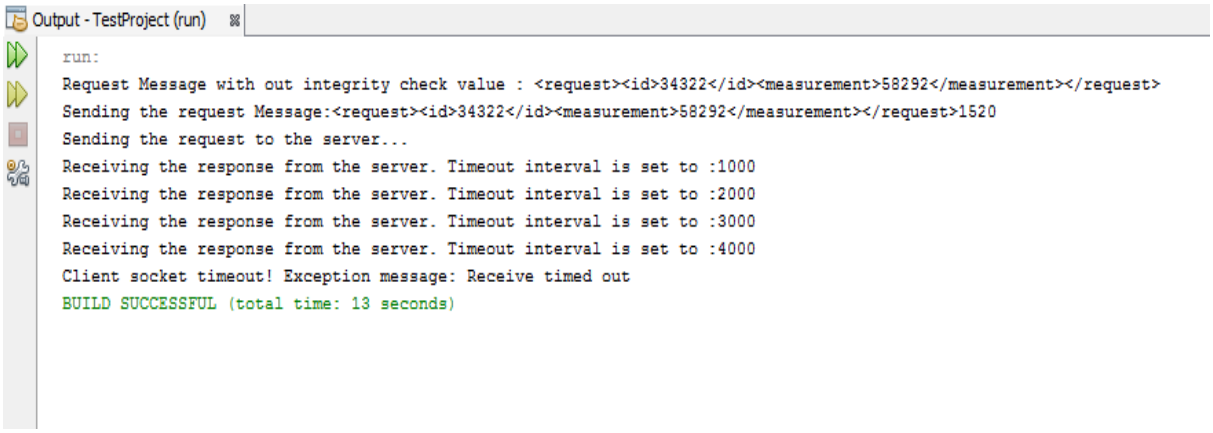


*Refer Server analysis for detailed method implementations

OUTPUT:

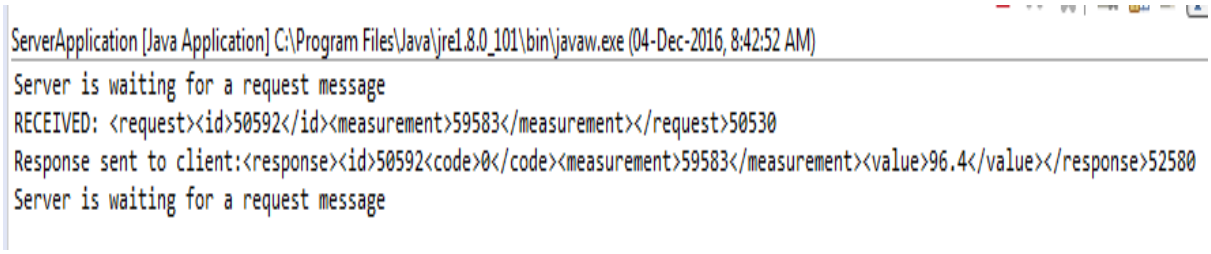
The outputs for various cases are as given below:

i. Receiver Timed out:

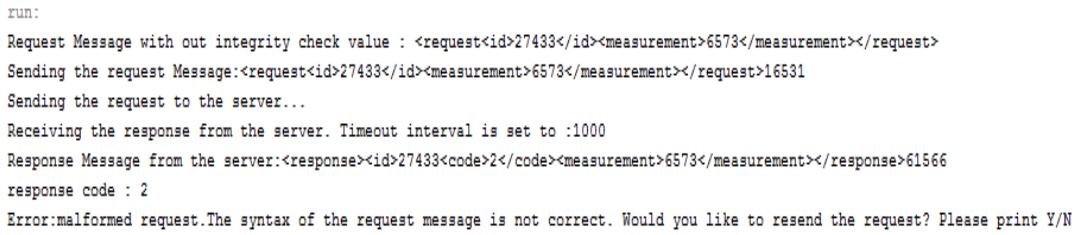


ii. Client Response Received:

iii. Server Request Received and response sent:



iv. Syntax incorrect:



v. Measurement Value does not exist

```
run:
Request Message with out integrity check value : <request><id>38652</id><measurement>1234</measurement></request>
Sending the request Message:<request><id>38652</id><measurement>1234</measurement></request>12978
Sending the request to the server...
Receiving the response from the server. Timeout interval is set to :1000
Response Message from the server:<response><id>38652<code>3</code><measurement>1234</measurement></response>27906
response code : 3
Error:non-existent measurement.The measurement with the requested measurement ID does not exist. Would you like to resend the request? Please print Y/
```

vi. Integrity Check Failure:

```
run:
Request Message with out integrity check value : <request><id>52173</id><measurement>3805</measurement></request>
Sending the request Message:<request><id>52173</id><measurement>3805</measurement></request>1234
Sending the request to the server...
Receiving the response from the server. Timeout interval is set to :1000
Response Message from the server:<response><id>52173<code>1</code><measurement>3805</measurement></response>13319
response code : 1
Request message's integrity check failed with an response code 1. Would you like to resend the request? Please print Y/N
```

vii. Server Side:

```
ServerApplication [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (04-Dec-2016, 8:56:39 AM)
Server is waiting for a request message
RECEIVED: <request<id>27433</id><measurement>6573</measurement></request>16531
Response sent to client:<response><id>27433<code>2</code><measurement>6573</measurement></response>61566
Server is waiting for a request message
RECEIVED: <request><id>38652</id><measurement>1234</measurement></request>12978
Response sent to client:<response><id>38652<code>3</code><measurement>1234</measurement></response>27906
Server is waiting for a request message
RECEIVED: <request><id>52173</id><measurement>3805</measurement></request>12348
Response sent to client:<response><id>52173<code>1</code><measurement>3805</measurement></response>13319
Server is waiting for a request message
```

viii. File not found exception

```
Server is waiting for a request message
RECEIVED: <request><id>8843</id><measurement>47351</measurement></request>62801
Unable to open file 'data1.txt'
Response sent to client:<response><id>8843<code>3</code><measurement>47351</measurement></response>47538
Server is waiting for a request message
```