Software Requirements Specification

for

<Packet Sending Tool>

Version 1.2 approved

Author:

Harshit Vashisth(1000014073)

Instructor:

Mr. Kapil Dev Sharma

(Assistant Professor)

Table of Contents

1. Introduction

1.1 Purpose

This document describes the software requirements and specification for Packet Sender team namely Packet Sending Tool so as to serve as a guide on one hand and a software validation document for the prospective client on the other.

1.2 **Document Conventions:** Font: Times Now Roman 12

1.3 Intended Audience and Reading Suggestions

The document is intended for all the stakeholders customer and the developer (designers, testers, maintainers). The user is assumed to have basic knowledge of English, computers and computer operations.

1.4 Definitions and Abbreviations

1.4.1 Definitions

- **Packet**

A packet is a small segment of a larger message. Data sent over computer networks is divided into packets. These packets are then recombined by the computer or device that receives them.

- **Admin**

Administrator. He is the sole authority to modify the code of the software adding new features etc .

- **User**

The holder of one end of the software who will send the data to another user . A user account can be maintained/accessed by anyone they shares their password with, the correspondence is not relevant to this problem.

- **Data**

The data of the different sender is stored there the history of past packet senders, along with their IP Adress and port number and the micro's and macros

.

- **User ID**

A unique ID assigned to both; The Sender and the Reciver once they open the software on the system . Each User's ID is nothing but the IP address of the users . Different formats helps the

system and the developers to show up things and features differently. Each IP address along with he port number is unique.

- **Packet Count**

Total count of Packets received from the Sender . Which is used to see the amount of lost data

- **Consultation**

A GUI interface which is easy to operate among both the user . All the data would be encrypted and both user ,doctor and the system can get the conversation done between both only on request.

1.4.2 Abbreviations

Throughout this document the following abbreviations are used:

IP: Internet Protocol
TCP: Transmission Control Protocol
UDP: User Datagram Protocol
SSL: Secure Sockets Layer

1.5 Project Scope

Packet Sender is a special programming tool used by Administrators to verify the network / net protocol. Packet Sender is a special programming tool used to send packets in a network, analyze them, and display them in a particular order in a Graphical User Interface and having features to recognize special keyword replies to some particular patterns. In its simple form Packet Sender simply sends different data packets or datagrams to another given network interface.

2. Overall Description

2.1 Product Perspective

The website network does not work independently. It works in parallel of both, user side and client side server and both the servers are connected to the database which is being updated at every instance whenever some changes are being observed.

Communication interface: The Sender and Reciver can communicate using the Chat Bot Interface based on UDP.

Software interface: The commands and updates sent via the communication network are specific to the target database. At present, single database would be there participating in this.

Hardware interface: The software will run on any basic computer which haves an internet connection .

.

User Interfaces

Customer: The user interface should be intuitive, such that 99.9% of all new users are able to complete their tasks without any assistance.

Admin: Admin is responsible for removing/adding users on requests. The Sender is approved to send when the reciver address giving by the sender is the valid IP address.

Maintainer: The maintainer is responsible for adding new features to the software and maintaining existing software. A maintainer should be possible to add a new feature to the software within minimum time and without taking down the software for longer duration.

2.2 Product Functions

The Software should work 24 hrs. The software identifies a IP Address and port number . It collects IP Address and port from the sender then there are two option to send the data like UDP and TCP . UDP service directly to the reciver . In the TCP its bit difficult first the user define the path from which the data is to sent , after the path is define the data is sent to the particular path only achieving 100% accuracy to send the data .

2.3 User Classes and Characteristics

The database required enough security measures to keep the server safe and secure. User credentials must be stored in encryption so that they can't be altered through some other agencies in any case. The software must handle concurrent accesses to the same account correctly. The database should be interrupted by any other software .

2.4 Operating Environment

The hardware, software and technology used should have following specifications:

- Ability to validate the IP Address
- Active Internet Connection
- Continuous power supply

- Ability to connect to user with the system
- Ability to take input from user
- Ability to validate reciver IP Address .

2.5 Design and Implementation Constraints

Validate Receivers IP Address :

- Validate for the IP Address
- Validate that the User is entering correct details
- If details are wrong, prompt error message "ENTER A VALID ID ADDRESS AND PORT NUMBER "

Validate for the receiver:

- Validate if receiver IP Address is valid or not .
- If IP Address is valid then build the connection to send .

GUI of software:

- User interface should be simple and easy to use.
- Every function of in the software should be represented independently.

Intense Traffic Generator :

- It should be able to achieve the speed of mili seconds .
- If the there is a problem while sending the data the error message pop up.
- There should be a button the reset and a button the stop and start the process.
- The values reflected to the user should be accurate .

Chat Bot:

- The Chat bot message should be able to send the message along with the intense traffic generator.
- The counting should count all the message received from the receiver.
- For every consecutive message received from the receiver , increment logic counter by 1.
- Reset login counter to 0 after the reset button is initiated.

2.6 Assumptions and Dependencies

- System has an up time of 100% and never goes down.
- Active and stable internet connection is there.
- User Shares his correct IP Address and the port number.

3. External Interface Requirements

3.1 User Interfaces

Functional Requirements

The functional requirements are broadly written into a single section , i.e. requirements of the user and are validated accordingly.

3.1.1 Requirements of the Users

The requirements for the User are organized in the following way General requirements, requirements for authorization, requirements for checking connectivity .

General

Functional requirement 1*:*

- **Description:** *Initialize a user with a specific IP address and port number*
- **Input:** *User is assigned IP address and ports of choice.*
- **Processing:** *Storing the IP Addresses in the database.*
- **Output:** *Reflect the security and connectivity of the network.*

Functional requirement 2:

- **Description:** *If its not possible to send the data the software should reflect the error message.*

Functional requirement 3:

- **Description:** *If the IP Address aren't available at a particular moment An error message is displayed.*
- **Input:** *A Ip address , port number, speed to be achieve.*
- **Processing:** *The message is converted into small packets of data and is send to the receiver side.*
- **Output:** *Display an error message.*
- **Authorization:** *The authorization starts after a user has entered IP Address and Port number of the receiver .*

Functional requirement 4:

- **Description***: The Chat Bot should be able to recogonise some particular pattern and is able to reply back from the user side.*
- **Input:** *Sender send the packet along with hidden pattern which should be recogonise*
- **Processing***: Check if the pattern matches we the predefine pattern which should be replied back .*
    - *The information received should be recogonised .*

- **Output:** *Display error message and reply back to the sender* .

3.2 Hardware Interfaces

 Description: If the credentials is valid, the system should read the IP Address initial to check the user type.
Input: Valid Receiver Details
Processing: Read the initials of the receiver side.
Output: Initiate authorization dialog accordingly.

Functional requirement 6:

- **Description Authorization dialog:** The user is requested to enter IP Address and port number . The system verifies the user IP Address and port number with the one in database.
- **Input:** Credentials from the sender .
- **Processing:** Verify for correctness and return true if it matches the database.
- **Output:** Accept or reject authorization of the credentials with an error message if its wrong.

Functional requirement 7:

Description: The Packet count is able to count the packet received .
Input: Response from authorization dialog: "Wrong IP Address or Port Number " if the details entered by the user is not valid .
Processing: If the User End gets any of these messages from the server, the software stops when the sender press the stop button .
Output: The count of packets receiving from the user .

Functional requirement 8:

 Description: If the Ip address and port number is verified then the data is send .
 Input: The Ip Address and port get verified from the receiver end.
 Processing: Finishing authorization.
 Output: Data is send to the receiver.

Functional requirement 9:
 Description: The Sender panel shows the data which is send including ascii value , ip addresses , messages and error.
 Input: It is automated user have to input only once.
 Processing: Store and reflect the information of every packet .
 Output: Display all the information in the GUI of sender.
Functions: The requirement that any Packet Sending Tool should have.

Functional requirement 10:

 Description: Store the information of different IP addresses .
 Input: Should be done from the detail panel where user has asked to enter the different values.
 Processing: It is done with the save button
 Output: Separate screen with those details which would be auto implemented if clicked on any row of details.

Functional requirement 11

 Description: Inputing dialog box.
 Input: User has to input his name message , ascii, address,port , resend delay.
 Processing: Error if the entered details are not valid .
 Output: Would be reflected in the receiver or sender panel .

Functional requirement 12:

 Description: Intense Traffic Generator.
 Input: Ip address and other details should be automatically stored user only has to enter the speed or resend delay .
 Processing: Send request to the function to send the data to the receiver end.
 Output: It would be reflected to the receiver side just the count of packets sending alogn with sending speed and time would be reflected.

Functional requirement 13:

 Description: Chat bot is able to send the particular pattern.
 Input: Pattern or string of message which will be recogonised by the user.
 Processing: The message or the pattern would be send along the intense traffic generator.
 Output: Software reflects the received data from the reciver chat bot in an another window.

4. System Features

4.1 System Feature 1

The customer user interface should be intuitive, such that 99.9% of all new users are able to surf and navigate on the software without any assistance and problems.

4.2 System Feature 2 (and so on)

The hardware should have following specifications:
 Ability to read the user inputs
 Touch screen for mobile connections
 Stable and an Active internet connection.
 Ability to connect to data base
 Ability to give outputs to users through a display.
 Ability to validate a user

4.3 Software Interfaces

The software interfaces are specific to the target website and user consultation systems**.**

5. Other Nonfunctional Requirements

5.1 Performance Requirements

 It must be able to perform in unstable connections also.
 Uninterrupted interrupted connections .
 High data transfer rate
 Able to handle many users at a time.
 Quick Response every time.
 Reply to particular patterns.
 Able to identify the quality of network .
 Should be able to tell if the security is compromised.

5.2 Safety Requirements

 Must be encrypted in the back-end
 Must be DDoS protected.
 Must have a detection technique for distinguishing between bots and Humans.
 Must be able to detect illegal/impermissible commands
 Must have a trace to the lost packets .

Security Requirements

 Users accessibility is censured in all the ways
 Users are advised to change port every two months
 Users are advised not to share their IP Address and port with anyone
 The maximum number of attempts to enter the IP address and port should be five.

5.3 Software Quality Attributes

Security
Performance.

5.4.1 Availability: The software has to be available 24 hours a day with at least 104 kb/sec connection speed

5.4.2 Security: The network should provide maximal data security .In order to make that much more transparent there are the following requirements:
1. It must be impossible to plug into the network.
2. It must be impossible to interrupt / peek through one’s connection

5.4.3 Maintainability: Only maintainers are allowed to maintain the software and the server .

6. Other Requirements

6.1 Data Base

The system must be able to use several data formats according to the data formats that are provided by the data bases of different users.A saved user data must have all the required attributes, i.e IP Address , port , message, name , resend delay etc.

6.2 Constraints

- Internet connection is a major factor.
- Ip Address and port number should be valid.
- Able to increase the rate of data transfer .