Deliverables:

* Challenge/Scenario to solve
* Description of the System
  + Components:
    - Registration Server
      * Purpose:
      * Function
    - Client
      * Purpose
      * Function
      * Interaction with Other Components
    - Test Cases
      * Purpose
      * Function
      * Interaction with Other Components
* Network System Architecture Diagram
  + Layers (UI, Network layer, DB Layer) accompanied with description.
  + Network Nodes, Edges, Databases, etc.
* Implemented Network Concepts with justification
  + Examples:  
    TCP/IP programming, implementation of network protocol, network error handling, etc.
* System testing (2 Unit and 2 Stress Test Cases)
  + Including development of tests
* Program
  + Each person is in charge of 1 layer.

Network System Engineering

[Name of the System]  
*Revision 1.0*

**Table of Contents**

[1. Introduction 1](#_Toc160115263)

[1.1 System Purpose 1](#_Toc160115264)

[1.2 Challenges 1](#_Toc160115265)

[1.3 Solution Overview 1](#_Toc160115266)

[2. Network System Design 1](#_Toc160115267)

[2.1 Functional Specifications 1](#_Toc160115268)

[2.2 Network Components 1](#_Toc160115269)

[2.2.1 Registration Server 1](#_Toc160115270)

[2.3 System Architecture 2](#_Toc160115271)

[3. System Testing 2](#_Toc160115272)

[3.1 Unit Test Cases 2](#_Toc160115273)

[3.2 Stress Test Cases 2](#_Toc160115274)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Name | Reason For Changes | Date |
| 1.0 |  | Network System Engineering | dd/mm/yy |
| 2.0 |  |  | dd/mm/yy |
| xx |  |  | dd/mm/yy |

Approved By

Approvals should be obtained for project manager, and all developers working on the project.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Signature | Department | Date |
|  |  |  |  |
|  |  |  |  |

# Introduction

## System Purpose

## Challenges

## Solution Overview

### Protocol Design

* Application Layer:
  + A domain-specific application-layer protocol will be created for structuring our messages.
  + Binary Encoding and Decoding over the transport and network layer will be used for simplicity.
    - Sample Messages:   
      “Register:{cliend\_id}:client\_info”  
      “Request:{client\_id}”
  + Both server and client require the capability of encoding and decoding the binary messages.
    - Server will take a different action depending on the received message type.
* Transport Layer:
  + TCP will be utilized for transmitting and delivering data over a network (Same Subnet).
* Network Layer:
  + IP will be utilized for sending and receiving over a separate network.

# Network System Design

## Functional Specifications

### Address Registration

* Client establishes a TCP/IP connection to the server and sends a registration request with the relevant metadata.
* Server responds with an acknowledgement message and the unique ID of the registration.

### Address De-Registration

* Client establishes a TCP/IP connection to the server and sends a deregistration request with the relevant metadata.
* Server responds with an acknowledgement message.

### Address Retrieval

* Client establishes a TCP/IP connection and sends a retrieval request.
* Server responds with the requested information over TCP. Client’s application layer must interpret the received information.

### Client-Client Testing

* Client performs various test cases over the address retrieved from the server.
  + The goal of these test cases is to retrieve information about the infrastructure of the network.

#### Test Cases

## Network Components

The server must be hosted on a static IP so that clients can know where to find it. The server’s address will be hard coded into the clients.

### Registration Host / Server

#### Core Functionalities

* **Purpose:**
  + Acts as a central hub for storing and providing host client registrations to requesting host clients. Its intermediary role aims to allow for “consensual” device discovery.
* **Functionalities:**
  + A client provides the required information for performing a registration in the host. The host then stores the provided data for future use.
  + Client provides the registered devices information on request.
  + A client can deregister from the host.
  + Must be able to encode and decode UTF-8 Strings into/from binary.

#### System Components

* **TCP Listener:**The server will run a TCP listener on a selected port, continuously accepting incoming connection requests from clients.
* **File-based Storage:**The server will store client registrations in a local file.

### Client

#### Core Functionalities

#### System Components

* **TCP Client:**

Each client will implement a TCP client that is capable of establishing connection with the server.

### Test Cases

* Test Case:
  + Purpose
  + Functionality
  + Implementation

## System Architecture

The system should provide the capability of accessing desired target clients over a subnet. This will be achieved via the use of public Ips when dealing with inter-network communication, in which case the router device will act as a Layer 3 device (a router), and intra-network communication, in which case the router device will act as a Layer 2 device (a switch).

A computer screen shot of a computer server

Description automatically generated

Figure 1 - Application's Network Diagram

The procedure for sending messages between client and server is well standardized. Any message type will follow the same basic procedure, simplifying any future development of program expansion.

A black background with white text

Description automatically generated

Figure 2 - UML Sequence Diagram - General Procedure

Given the objective of the application, the procedure for obtaining a target client is also well defined. It is up to the test case to then specify exactly what the description of the test will be.

A black screen with white text

Description automatically generated

Figure 3 - UML Diagram - General Testing Procedure

# System Testing

## Unit Test Cases

## Stress Test Cases

Appendix A: Screen Shots and Source Code Snippets

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Template for Unit Test Case | | | | | |  |  |
|  | Project Name: |  | Test Designed by: |  |  |  |  |
| Module/Component Name: |  | Test Designed date: |  |  |  |  |
| Release Version: |  | Test Executed by: |  |  |  |  |
|  |  | Test Execution date: |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Pre-condition |  | | | | | | |
| Postconditions: |  | | | | | | |
|  |  |  |  |  |  |  |  |
| Test Case# | Test Title | Test Summary | Test Steps | Test Data | Expected Result | Status  (Pass/Fail) | Notes (if any) |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Template for Stress Test Case | | | | | |  |  |
|  | Project Name: |  | Test Designed by: |  |  |  |  |
| Module/Component Name: |  | Test Designed date: |  |  |  |  |
| Release Version: |  | Test Executed by: |  |  |  |  |
|  |  | Test Execution date: |  |  |  |  |
| Test Case# | Test Title | Test Summary | Test Data | Breakout Point | Safe  Point | Status  (Pass/Fail) | Notes (if any) |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |