<<Encrypted Chat Software>>

Software Design Specification

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 7/8/15 | 1.0 (Final) | SDS … | Ruben |

Table of Contents

[1. 1. INTRODUCTION 4](#_Toc426897935)

[1.1. Purpose 4](#_Toc426897936)

[1.2. Scope 4](#_Toc426897937)

[1.3. Document’s Intended Audience 4](#_Toc426897938)

[1.4. Overall Overview 4](#_Toc426897939)

[2. SYSTEM OVERVIEW 4](#_Toc426897940)

[2.1. MAJOR DESIGN ARTIFACTS 4](#_Toc426897941)

[2.2. FUNCTIONAL REQUIREMENTS 4](#_Toc426897942)

[2.3. SUPPORTING MATERIALS 4](#_Toc426897943)

[2.4. DEFINITION AND ACRONYMS 5](#_Toc426897944)

[3. DESIGN CONSIDERATIONS 5](#_Toc426897945)

[3.1. ASSUMPTION 5](#_Toc426897946)

[3.2. CONSTRAINTS 5](#_Toc426897947)

[3.3. SYSTEM ENVIRONMENT 5](#_Toc426897948)

[3.4. DESIGN METHODOLOGY 5](#_Toc426897949)

[3.5. ANTICIPATION DURING DEVELOPMENT 5](#_Toc426897950)

[3.6. RISK AND FUTURE DESIGN CHANGES 6](#_Toc426897951)

[4. ARCHITECTURAL DESIGN 6](#_Toc426897952)

[5. HIGH LEVEL DESIGN 8](#_Toc426897953)

[6. LOW LEVEL DESIGN 9](#_Toc426897954)

[7. USER INTERFACE 9](#_Toc426897955)

[8. SCENARIOS 10](#_Toc426897956)

[9. LESSON LEARNED 16](#_Toc426897957)

1. INTRODUCTION

## Purpose

The purposes of this document are to describe all the functions and system requirements of this software, to explain the design of a system in terms of several aspects, for example the architecture system, high level and low level.The design of the software will also be explained in this below.

## Scope

* UML diagram, describe the flow of the program.
* This software use Java Runtime Environment to run.
* All messages are encrypted by using AES.

## Document’s Intended Audience

This document is intended to all users who use this software.

## Overall Overview

This document will describe the system design we have made in terms of some aspects and considerations.be divided into several parts, for example the aarchitecture, high-level and low-level design is more of a basic design of design system. These include the collaboration between the subsystems, the course of each feature with sufficient detail and base architecture of the system used. All contents are written in the table content.

# SYSTEM OVERVIEW

## MAJOR DESIGN ARTIFACTS

1. Encryption
2. Server and Client

## FUNCTIONAL REQUIREMENTS

* + Connect to server or client
  + Send and received message
  + Message Encryption

## SUPPORTING MATERIALS

Reference:

http://pirate.shu.edu/~wachsmut/Teaching/CSAS2214/Virtual/Lectures/chat-client-server.html

## DEFINITION AND ACRONYMS

SDS : Software Design Specification.

SRS : Software Requirement Documentation.

AES : Advanced Encryption Standard

NIC : Network Interface Card

# DESIGN CONSIDERATIONS

## ASSUMPTION

* 1. All computers are installed with NIC can be connected to a network
  2. All computers have been installed with Java Runtime Environment

## CONSTRAINTS

This software only can communicate in an intranet network. We have not developed connection via internet network.

## SYSTEM ENVIRONMENT

This software develops using Java where is completed with a network with 2 or more computers.

## DESIGN METHODOLOGY

UML Design Approach: Using UML to describe all the requirements. UML diagrams are used, among others: Use case diagrams, collaboration, activity and deployment diagrams.

## ANTICIPATION DURING DEVELOPMENT

Anticipation that we have made during the development of our project as below:

* We will list down all details to the design documentation in the form of logs. This documentation is useful as a reference, if changes are made by each member of the group.
* Iterative process that we apply to anticipate changes in the design, especially regarding changes in requirements. If there are any requirement changes, we can conduct additional design requirements and concerns about
* There are several designs which we feel is too hard to do, especially with regard to programming is quite complicated. Therefore, we do various alternatives by using algorithms that are easier that the design can still be made and also functioning properly.

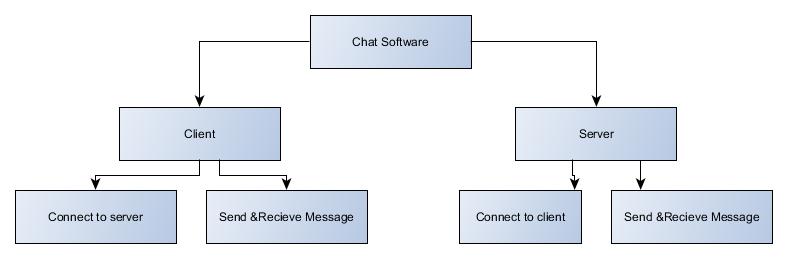
## RISK AND FUTURE DESIGN CHANGES

There are limitations due to the lack of time. So we anticipate risks that will happen:

* Data Corrupted.

There are errors in computer data that occur during writing, reading, storage, transmission, or processing, which introduce unintended changes to the original data. Computer, transmission and storage systems use a number of measures to provide end-to-end data integrity, or lack of errors.

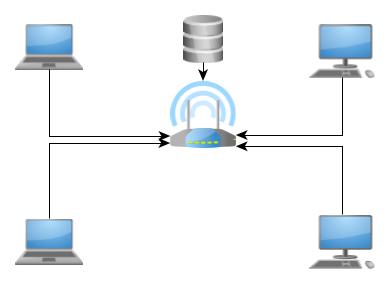
# ARCHITECTURAL DESIGN

**Architecture for Client and Server**

**Data-Flow**

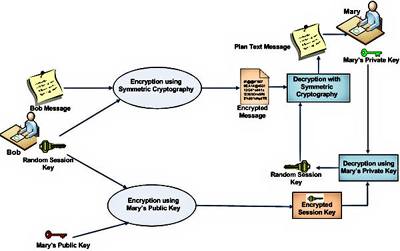
****

**Encrypted Chat Software (General View)**



# HIGH LEVEL DESIGN

**Network Encrypted Chat System**



***Encryption*** *is the process of encoding messages or information which only can read by the authorized parties. Encryption does not of itself prevent interception, but denies the message content to the interceptor.*

This program uses the concept of client/server based, encrypting all messages before send to recipients in order to prevent message being hijacked and knew the content by intruders. It has already certified that it is secure enough. The purposes to build this secure all the messages being sent, to prevent content being interject by hackers.

This project uses tools as below :

* *Programming language used: Java.*  We used Java Programming when we developed this project.
* Database only stored clients and will automatically remove when it exits.

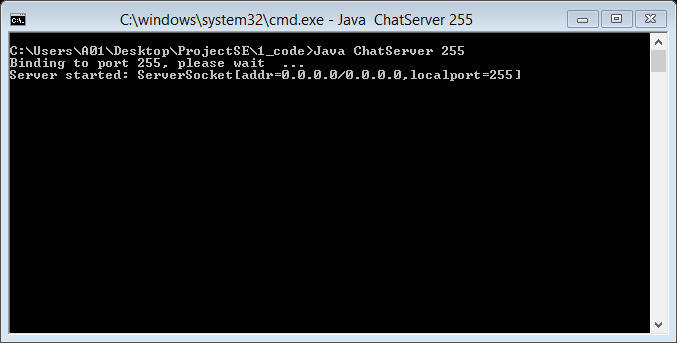
# LOW LEVEL DESIGN

There are some important source code that we used in the manufacture of our project.

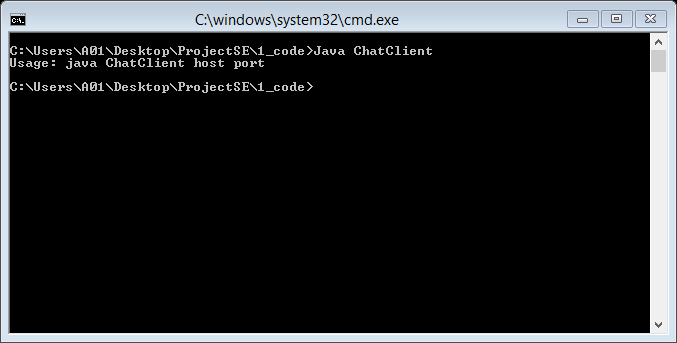
# USER INTERFACE

**General: Command Prompt**

**Server:**

****

**Client:**

****

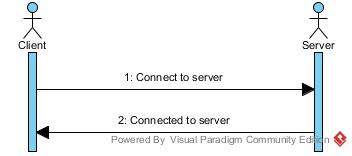
# SCENARIOS

**Actor:**

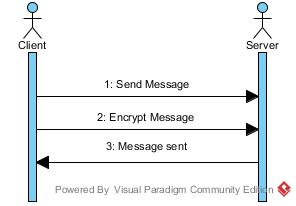
Client/Server

**Role:**

User

****

Connect to Client / Server

****

Send Message

# LESSON LEARNED

As the project is being developed, we have learn some techinques and theory as below:

* + AES Encryption Technique.
  + How does chat server works.