Software Requirements Specification (SRS) for Computational Security Project

*Baseline version 0.1*

*Issued on : July 12, 2015*

Issued by :Clientele, Inc.

Issued for : Client

**Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Changes** |
| 0.1 | July 12, 2015 | Devi | initial version |
|  |  |  |  |

**Document Approval**

The following Software Requirements Specification has been accepted and approved by the following :

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Title** | **Date** | **Signature** |
| Devi | Project Manager |  |  |
| Client |  |  |  |
|  |  |  |  |

**Table of Contents**

List of figures 4

List of tables 5

Chapter I Introduction 6

* 1. Purpose
  2. Scope
  3. Definitions, Acronyms, and Abbreviations
  4. Overview

Chapter II General Description 7

* 1. Product Perspective
  2. Product Functions
  3. User Characteristics
  4. General Constraints
  5. Assumptions and Dependencies

Chapter III Specific Requirements 8

* 1. External Interface Requirements
  2. Functional Requirements
  3. Use Cases
  4. Classes/Objects
  5. Non-functional Requirements
  6. Design Constraint

Chapter IV Analysis Model 12

* 1. Sequence Diagrams
  2. State-Transition Diagram

Chapter V Change Management Process 15

**List of Figures**

Class Diagram 11

Sequence Diagrams 12

State Diagram

**List of Tables**

Abbreviations Table 6

Use Case Scenarios 9

**Chapter 1**

**INRODUCTION**

* 1. **Purpose**

This document is written to identify the requirements and design of the Computational Security project. Said project is to create a software that is a chat messenger software that provides encryption.

* 1. **Scope**

The software will be able to create a chatroom with encryption implemented. The software will be divided into two parts : client and server. The message sent in the chatroom will be encrypted, which will provide security and confidentiality.

This project will focus on the software itself. Hardware and other requirements and dependencies are assumed to be fulfilled, and is out of the scope of this project.

* 1. **Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| Term | Definition |
| Client | A class in the software that functions as the participant of an existing chatroom. |
| Server | A class in the software that manages the chatroom. |
| User | Refers to the one using the software, be it the client or the server |

* 1. **Overview**

This document will have four parts:

1. General Description  
   This part describes the project’s informal requirements.
2. Specific Requirements  
   This part describes the project’s requirements in technical terms. Mainly written for the developers.
3. Analysis Model
4. Change Management Process

**Chapter 2**

**GENERAL DESCRIPTION**

1. **Product Perspective**

This project is an independent project.

1. **Product Functions**

The software will be divided in two parts :

1. Server, which, when executed, will create a chatroom
2. Client, which, when executed, will be able to join an existing chatroom

Both server and client will have encryption and decryption ability, to process the message received and to send a message respectively.

1. **User Characteristics**

The software is designed for a client. They won’t really look at the system, only to look at its output.

1. **General Constraints**

These are general constraints of the software :

* The software is written in Java.
* The software will run under Windows 7.

1. **Assumptions and Dependencies**

These are the assumptions and dependencies of the software :

* While the system should run with Java Runtime Environment installed, it will only be supported in Windows platform

**Chapter 3**

**SPECIFIC REQUIREMENTS**

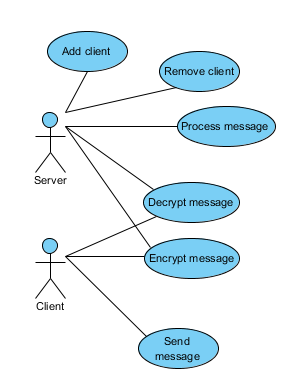
1. **External Interface Requirements**

The software is made with the operating system Windows in mind – if ran in other OS, the software might not work. As for the interface, the software is a command-prompt based chatroom, so no further additions is necessary.

1. **Functional Requirements**

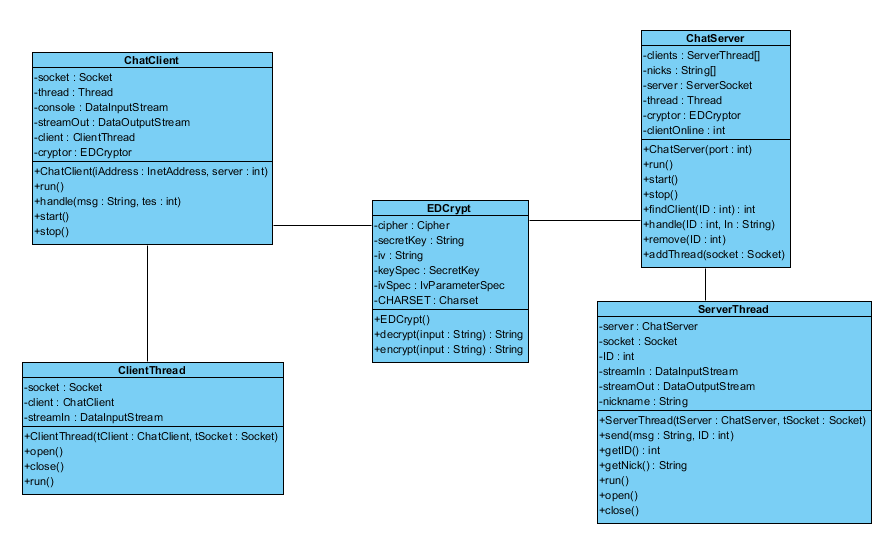
* Server will be able to create a chatroom
* Client will be able to join an existing chatroom
* Client will be able to encrypt messages
* Client will be able to send messages
* Client will be able to receive messages
* Client will be able to decrypt received messages
* Server will be able to encrypt messages
* Server will be able to receive messages
* Server will be able to forward messages to the other clients
* Server will be able to decrypt received messages

1. **Use Cases**



|  |  |
| --- | --- |
| UC-1 : Add Client | |
| Primary Actor | Server |
| Stakeholders and Interests | Server : wants to add a client to the chatroom |
|  | Client : wants to enter the chatroom |
|  |  |
| Preconditions | The chatroom has been made, the client has made a request to join |
| Success Guarantee | A new client enters the chatroom |
|  |  |
| Main Success Scenario | 1. The server receives a request to enter |
|  | 2. The server checks if the chatroom is full or not |
|  | 3. The server adds the client to the chatroom |
|  | 3.a If the chatroom is full, the client is rejected |
|  |  |
| UC-2 : Remove Client | |
| Primary Actor | Server |
| Stakeholders and Interests | Server : wants to remove a client from the chatroom |
|  | Client : wants to exit the chatroom |
|  |  |
| Preconditions | The client sent a request to quit |
| Success Guarantee | The client is removed from the chatroom |
|  |  |
| Main Success Scenario | 1. The server receives a request to exit |
|  | 2. The server removes the client from the chatroom |
|  | 3. The server notifies the remaining clients |
|  |  |
| UC-3 : Send Message | |
| Primary Actor | Client |
| Stakeholders and Interests | Client : wants to send a message |
|  | Receiver(client) : wants to receive a message |
|  |  |
| Preconditions | The sender is connected to the chatroom |
| Success Guarantee | All receivers receives the message |
|  |  |
| Main Success Scenario | 1. The sender inputs the message |
|  | 2. The system encrypts the message |
|  | 3. The system sends the message to the server |
|  |  |
| UC-4 : Process Message | |
| Primary Actor | Server |
| Stakeholders and Interests | Server : wants to process the message |
|  | Client : wants the message sent to the server to be processed |
|  |  |
| Preconditions | The sender is connected to the chatroom, the server has received a message |
| Success Guarantee | All clients receives the processed message |
|  |  |
| Main Success Scenario | 1. The server retrieves the message |
|  | 2. The server decrypts the message |
|  | 3.a If the message is a request, then the requested action will be processed |
|  | (example : exiting chatroom |
|  | 3.b If not, the encrypted message is sent to all clients |
|  |  |
| UC-5 : Encrypt Message | |
| Primary Actor | Sender (Client/Server) |
| Stakeholders and Interests | Sender : wants to encrypt a message to be sent |
|  | Receiver : wants to make sure the messages sent are confidential |
|  | (either Client and Server are the receivers) |
| Preconditions | The sender is connected to the chatroom |
| Success Guarantee | The input message is encrypted |
|  |  |
| Main Success Scenario | 1. The sender requests an encryption process |
|  | 2. The system retrieves an input from the sender |
|  | 3. The system computes the encrpytion |
|  | 4. The system returns the encrypted message |
|  |  |
| UC-6 : Decrypt Message | |
| Primary Actor | Receiver (Client/Server) |
| Stakeholders and Interests | Receiver : wants to decrypt a message |
|  | Sender : wants to make sure the messages sent are confidential |
|  | (either Client or Server are the receivers) |
| Preconditions | The receiver is connected to the chatroom. The receiver has received an encrypted message. |
| Success Guarantee | The input message is decrypted |
|  |  |
| Main Success Scenario | 1. The receiver requests an encryption process |
|  | 2. The system retrieves the encrypted message |
|  | 3. The system computes the encrpytion |
|  | 4. The system returns the encrypted message |

1. **Classes/Objects**

**

1. **Non-Functional Requirements**

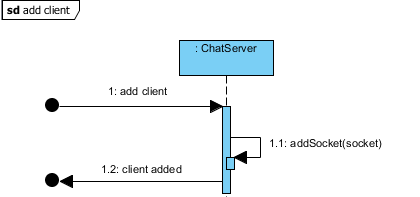
* Creating a chatroom should take less than 1s
* Joining a chatroom should take less than 1s
* The format of the message is UTF-8
* The interface should be straightforward, in such that users need no further training to use it

**Chapter 4**

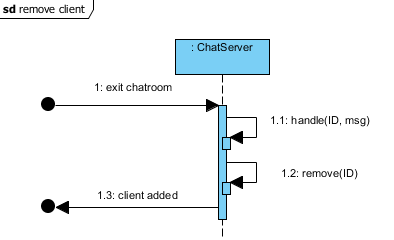
**ANALYSIS MODELS**

1. **Sequence Diagrams**

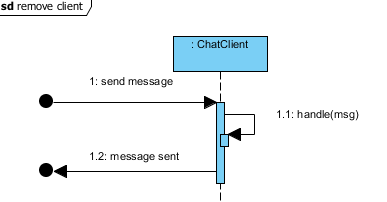
**Add Client**

****

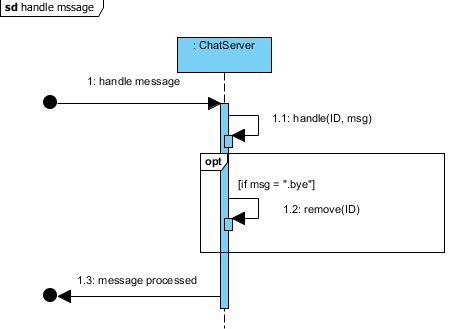
**Remove Client**

****

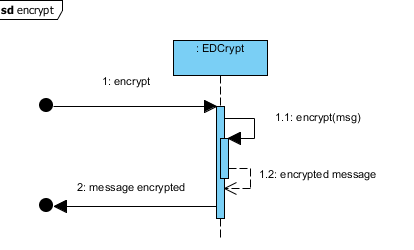
**Send Message**

****

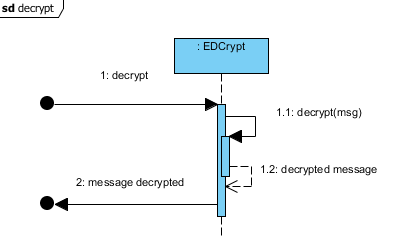
**Process Message**

****

**Encrypt Message**

****

**Decrypt Message**

****

1. **State-Transition Diagrams (STD)**

**Chapter 5**

**CHANGE MANAGEMENT PROCESS**

If the project’s scope and requirements change, our members will discuss on what will the change(s) will effect, and who will handle the changes. These changes will then be reviewed together before it is