**Requirements**

**Introduction**

The purpose of this document is to define the requirements for the development of a simple client/server network.

This client will include a dictionary with pickling functionality (binary, JSON, and XML formats), and the server will be configurable to handle encrypted content and allow a user to print the contents too.

**System Overview**

**Architecture**

* https://www.figma.com/file/YG1RrE2DLz1iYZhGaoxFsh/GROUP-PROJECT?type=whiteboard&node-id=0%3A1&t=FnkDWWGEn9d7JTEE-1

**System Components**

The system will consist of the following components:

* Network Server
* Dictionary Module
* Encryption Module
* Server Component
* Client Component

**Functional Requirements**

**Client**

**Network:**

* Establish a connection to the server. It can be local or over the network.

**Dictionary Operations (Use Case 1):**

* Create a dictionary and populate it (not defined how).
* Serialize the dictionary with the following options:
  + Binary
  + JSON
  + XML
* Send the serialized dictionary to the Server.

**Text File Operations (Use Case 2):**

* Create a text file.
* Allow the user to encrypt the contents of the text file.
* Option for the user to either send the encrypted payload to the server or plain text.

**Server**

**Network:**

* Listen and accept connections from clients. It can be local or over the network.

**Data Handling (Use Case 1)**:

* Receive and deserialize the dictionary or text file from the client.
* If encrypted, decrypt the dictionary before deserializing it.
* Print the received data to the screen or to a file.

**Data Storage (Use Case 2)**:

* Configuration option on the Server to allow Server output to be stored to files, printed to the screen (UI) or both.

**Testing**

**Exception Handling**

* Proper exception handling must be implemented to manage potential errors.

**Unit Tests**

* We will conduct three, unit tests to check the functionality and correctness of the code. These will include:
  + Testing the client module is working as expected.
  + Testing the server is working as expected.
  + Testing the transfer of data between the client and server is behaving as expected.

**Code Reviews**

* Code should be reviewed and approved by independent testers.

**Performance**

* The system should transmit data between the client and server in < 60 seconds.

**Non-Functional Requirements**

**Dictionary User Interface**

* The user interface of the application shall be intuitive, and user-friendly so that a user is able to easily set the pickling formats and encryption settings.

**Coding Standards**

* All code must adhere to the Python Enhancement Proposal (PEP) standard for code style, naming conventions, and formatting.

**Source Control**

* The project shall be uploaded and maintained on GitHub for version control and collaboration purposes.
* Commits made must include well-documented commit messages, describing the changes.

**User Guides**

* User guides should be created to provide a walkthrough on how to:
  + Use the Dictionary function
  + Encrypt a file
  + Decrypt a dictionary
  + Send a non-encrypted file
  + Send an encrypted file
  + Disable the connection between the client and the server

**Constraints**

* The system shall be developed using the Python 3 and above.