SECURE FILE TRANSFER APPLICATION

Functional Specification

Author: Aoife O'Brien

Student ID: C00214279

Project Supervisor: Patrick Tobin

Recipient: Institute of Technology Carlow

Date: Friday 15th November 2019

Contents

Abstract

The purpose of this document is to set out how this application will work. This document will detail the functionality that the application will provide, how users will interact with it and what it will look like. Some consideration will also be given to any design issues which may present themselves to ensure that a realistic system is specified. The primary target users of this application are small and medium enterprises. This application will provide these organisations with an easy and efficient way to securely transfer confidential files between employees, helping to ensure that modern security standards and best practices are adhered to.

Introduction

The ability to move data reliably and securely from one location to another is an important key to success and survival for many companies. In businesses for which their primary product or service is data, this capability is even more critical.
The file transfer application will be a user friendly and intuitive system to use. It will provide users with a clean graphical user interface (GUI), which is easy to navigate. Users will have the ability to send files as frequently as they wish. When sending a file, users will only require minimal interaction with the system, making it particularly useful for people with a non-technical background. Users will not need to worry about handling encryption keys for example, as this takes place on the backend of the system. The application will securely transfer the files between users. It will do this using strong security techniques, i.e. encryption and VPN technology. It will also give users the opportunity to verify the integrity of files through the use of a hashing algorithm.

Project Scope

Goal

To design an easy-to-use and efficient secure file transfer application.

Requirements

Tools

Java: Java will be the programming language used to develop the front end of the application, i.e. the graphical user interface and its functionality. This language was chosen because it meets the security and functionality requirements outlined for this part of the application. E.g. it is cross-platform, distributed, secure and robust, among other characteristics.

Encryption Algorithm(s): A hybrid encryption approach will be used to encrypt the files before transfer and to decrypt once received.

MySQL Database: A MySQL database is required to securely store users' credentials upon registration with the application and to retrieve these credentials as necessary.

Java Socket(s): A Java socket pair will be used to transfer the encrypted files between users.

VPN Tunnel: A VPN tunnel will be implemented into the sending process for increased security.

Server: A server will be used to perform the relevant operations between the sender and the recipient. The server will be assigned a variety of roles to perform the necessary operations, e.g. file server and application server roles.

End Device(s): An end device such as a PC or a laptop will be necessary to run the application.

Libraries

The Swing Library in Java will play a primary role in the development of the user interface.

The Java Bouncy Castle Library, an all-purpose cryptographic library, will be used to aid in the encryption and decryption operations.

The Java Database Connectivity (JDBC) driver library will be used to connect to the MySQL database.

Deliverables

- A simple graphical user interface (GUI) which allows users to register and authenticate with the system and send and receive files securely between other users using secure mechanisms via text boxes, file choosers, drop down menus and icons.
- A server which securely hosts the application and performs all the relevant application operations efficiently.
- A database which securely stores user credentials.
- A virtual private network (VPN) tunnel will be incorporated into the transferring of the files to securely send them across to their destination.

Assumptions

- Time and resources will allow for the design and creation of all components.
- The encryption algorithms used are currently secure and will remain so for the foreseeable future.

Risks

- Time may not allow for all deliverables to be fully completed.
- The system may not work on different platforms.
- One of the selected algorithms could potentially be broken or declared insecure during or after the development stage.

Target Market

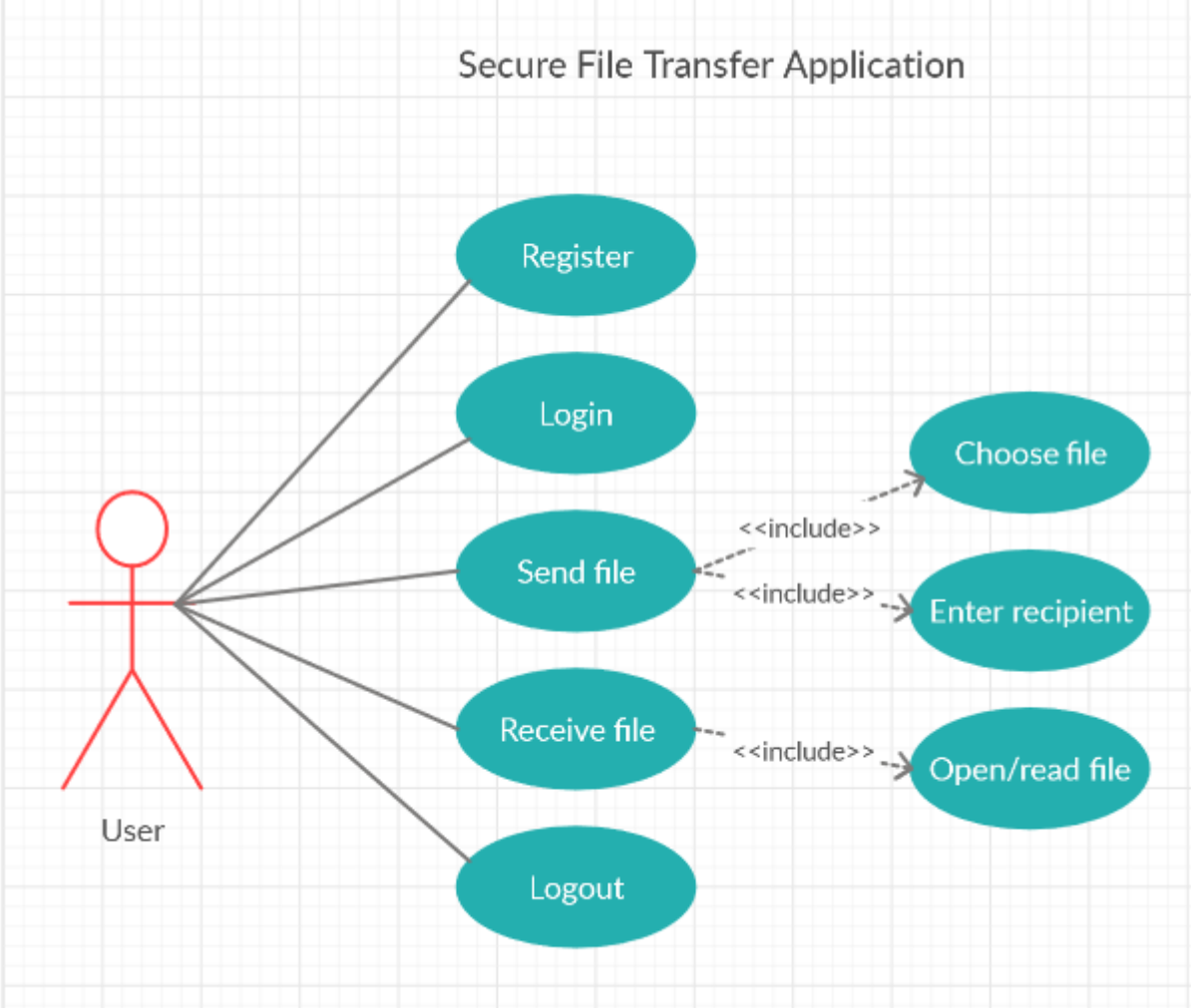
While this project is more of a proof of concept for the moment, the primary user group of this system would be organisations, more specifically small and medium sized businesses. This system aims to provide a means of secure file transfer for companies who may not have

the need or resources for a larger more expensive solution, i.e. SME's who often do not have the same financial capacity of large organisations.

Application Users

Within the application, all users will be standard users with standard privileges. These will be the normal users of the application. They will be able to register, login, send and receive files and logout. There will be a single admin account, which I will use when necessary to manage, edit and maintain the application and its associated infrastructure.

Use Case Diagram



(Created using Creately.com, 2020)

Brief Use Cases

Register

Actors	User, Application, MySQL Database
Preconditions	The user has successfully launched the application.
Actions	The user will enter the information asked for on the registration screen, i.e. first name, last name, username, email address and password. The user is required to confirm their password in a second field provided. This information will be securely stored in the database.
Expected Result	The user is successfully registered with the system.
Alternative(s)	1. The user enters invalid information, e.g. an already taken username – A generic error message is returned, e.g. "*Invalid username and password combination. Please try again*". 2. The user confirms their password and the two password fields do not match upon comparison – An error message is returned saying "*Passwords do not match. Please try again*".

Login

Actors	User, Application, MySQL Database
Preconditions	The user has successfully registered with the system.
Actions	This use case starts when a user loads the login page with the button provided on the registration page. The user enters their username and password. The information is validated by the system. If the information is correct, the user is granted access to the application.
Expected Result	The user successfully logs into the application and is presented with the welcome screen.
Alternative(s)	1. The user enters incorrect information – An error message is returned saying "*Wrong username or password. Please try again*". 2. The user forgets their password and chooses the forgotten password link provided on the screen and follows the given steps to reset their password.

Send File

Actors	User, Application, MySQL Database
Preconditions	The user has successfully logged in to the system.
Actions	This use case begins when a logged in user wishes to send a file to another user. The user chooses the file from their device using the file chooser provided. This information is received by the application and the selected file is encrypted in the background. The user then enters the recipient and

	clicks Send.
Expected Result	The user has successfully chosen a file to send.
Alternative(s)	1. The user selects a file which no longer exists – A file error message is returned and the user can start the process again.

Receive File

Actors	User, Application, MySQL Database
Preconditions	The recipient is logged in to the application. A file has successfully been sent to the recipient.
Actions	This use case begins when a logged in user receives a file from another user. The file is decrypted by the application in the background, and saved to the recipients device. The recipient has the option to verify the integrity of the received file using the hash given. The user then opens and reads the file.
Expected Result	The file is successfully decrypted and is then visible and legible for the recipient to read.
Alternative(s)	1. Decryption fails or a file gets corrupt – An error message is returned to the user. The user can contact the sender and let them know that there was an issue with receiving the file, along with the administrator so that the issue can be investigated and resolved.

Logout

Actors	User, Application, MySQL Database
Preconditions	The user wishes to log out of the application.
Actions	This use case begins when a logged in user wishes to log out of the system. The user clicks the '*Logout'* button, presses '*Yes*' when asked '*Are you sure?*', and is then logged out.
Expected Result	The user successfully logs out of the application.
Alternative(s)	1. The user presses the Logout button by accident – A pop up box will appear on the screen asking the user to confirm the logout. The user can cancel the logout at this stage if they do not wish to logout. The user will remain logged in.

Core Functionality

The core functionality includes registration, login, sending files, receiving files and logging out, and is as follows:

- Firstly, users will be required to register with the system and will be asked to authenticate with their username and password each time they open the application.
- These credentials will be stored securely in the backend database and retrieved and checked each time a user attempts to login.
- The primary functionality must allow the sender to enter the recipient's name and select a file to send from the file chooser menu provided.
- The application will encrypt the file using a hybrid encryption approach and send the file across to the recipient using Java sockets.
- When the recipient receives the file, the system will decrypt it using the corresponding decryption processes for the hybrid encryption implementation. Users will also be provided with a hash of the original file in case they wish to verify the file integrity before opening it.
- Lastly, users will be able to logout of the application when finished.

Secondary Functionality

Secondary functionality includes the following:

- A VPN Tunnel incorporated into the sending of the files for an added layer of security.
- Password reset capability for users.
- A notification message for the sender upon the recipient getting the file.

Metrics

The success of this project will be based on the following factors:

- ✓ **Security:** It should securely transfer files between users using secure techniques including strong encryption.
- ✓ **Usability:** All functional requirements should be implemented.
- ✓ **Efficiency:** It should perform efficiently and in a timely manner.

FURPS

Functionality

This application must allow users to send and receive files. The application must serve the primary purpose of securing the files throughout transfer.

Usability:

Users must be able to register with the application in less than five minutes.

Once logged in, users should remain logged in to the application until they choose to log out.

Users should be able to instinctively navigate the application without completing any usage tutorials.

The response time of the system should be quick, satisfactory and efficient.

Any responses presented to the users should be formatted in a clear, legible and understandable format. These features include but are not limited to, the notification message given to the sender, the variety of generic messages returned upon successful registration and both successful and invalid logon attempts.

Reliability:

The application will be developed to be as robust as possible. As the most likely weakest link in the system, the front end in particular will be designed with this in mind. Within the language used for design, all relevant potential exceptions will be handled in the correct manner. The centralised design of the application will allow for quicker a recovery period should an availability issue arise.

Performance:

The application user interface should load for the user in under thirty seconds.

The application should perform at quality speed and efficiency. The most time consuming and resource intensive elements will be the processes of encryption and decryption, which

will vary depending on the size of the file being used. However, this will be considered and tested during the development stage.

File compression will be considered prior to encryption in order to increase speed and efficiency.

Supportability:

The system will be developed as a Windows based application, but with the intention of working cross-platform. If time allows once the application is up and running, the code will be edited to function on a Linux based system also.

The application should be capable of performing on all modern Windows based personal computers.

Project Inspiration

The ability to move data reliably and securely from one location to another is an important key to success and survival for many companies. In businesses for which their primary product or service is data, this capability is even more critical.

As previously stated in the accompanying Research Manual for this project, I believe that the concept of secure file transfer is not given the respect it deserves.

I discussed that what led me to do this project is the fact that while the action of sending sensitive data is performed millions of times worldwide every day, many people do it without giving it a second thought. The fast pace of the working life today makes time very valuable. People often try to save time in any way they can. Unfortunately, one key area this time saving seems to harm is secure file transferring.

I believe that protecting sensitive data is of paramount importance today, and I don’t believe that best practice is being used in a frightening number of organisations. This may be due to a variety of reasons, such as lack of awareness and education, lack of resources and much more. However, the handling of sensitive files and information should be a top priority for businesses, and the potential damage from the exposure of such files can often be far worse than the time, effort and money involved in putting a secure system in place.

Similar Products

Existing solutions currently available on the market include;

Go Anywhere Managed File Transfer Solution

Go Anywhere is a service designed by the company HelpSystems that provides a paid managed file transfer service to organisations. This solution "automates and secures file transfers using a centralised enterprise-level approach and provides a safe and audited method for automatically transferring files within and outside of an enterprise" (Go Anywhere, 2019).

LiquidFiles

"Liquid Files is a Virtual Appliance that you install in your VMware, Microsoft Hyper-Visor, Xen environment, in your own private Amazon EC2 Cloud, or if you prefer on a dedicated server. LiquidFiles aims to fulfil all needs most organizations need to Send, Receive and Share Files of any file size in and out of your environment" (Liquid Files, 2019).

How they differ

Both above solutions differ from each other and the system in question regarding various factors including product format, environment(s), size, cost, and features and capabilities.

For example, LiquidFiles comes in the form of a virtual appliance, where this system will be in the form of a centralised application. The choice of environments on which to install each solution varies also, with choices including on-premise, cloud solutions, XEN Environments and more. The above solutions are larger than the system in question and are more suited towards large organisations, as they offer more advanced features that would benefit a large organisation. Whereas, this system would be a more cost-effective solution.

References

GoAnywhere.com, 2019. *GoAnywhere Managed File Transfer (MFT)* [Online] Available at: <https://www.goanywhere.com/solutions/managed-file-transfer> [Accessed 24th October 2019].

LiquidFiles, 2019. *LiquidFiles* [Online]. Available at: <https://www.liquidfiles.com/> [Accessed 24th October 2019].