

ABSTRACT

The issue of privacy of data is very paramount with each of us, especially for those who share their computers. Since, there is no room for error when it comes to protection of confidential files; the way in which the security can be provided to our files and applications becomes a matter of primary importance. This project proposes a system that provides two-level security to ones' files and applications. The first level of security is provided by ones' mobile phone which acts as a Bluetooth device. The second level of security is provided by the users' personal security password. In addition to the two-level security, the system also uses AES algorithm for file encryption and decryption, which is proven to provide a higher level of security to the files. Therefore, the proposed system will be a simple, personalized, and effective method to secure the files and applications in the computers that are shared by multiple users.

1. INTRODUCTION

1.1 INTRODUCTION

Computer Security refers to protection of the computing system and the data available in it. Security of data is a major issue that needs to be handled when it comes to computer security. Over the years various locking mechanisms and cryptosystems were designed and implemented to handle the needs of security.

The most basic method used to secure a file is to lock it with a password. The access to the file is revoked unless the system is provided with a valid key to unlock the file. Even though password mechanism is available in all the systems for securing files, it is proved to be inefficient due to the availability of various password cracking tools. This inability to provide security give rise to the “two-level authentication mechanism”. Various unique identities like “Unique Device Address”, “IMEI Number” were combined with ones’ personalized password for providing the user with the two-factor authentication.

In order to strengthen the security of the data, cryptosystems were developed. Instead of locking the files, the cryptosystem converts a plain text to ciphertext for encoding and the reverse for decoding the data. These File Encryption and Decryption mechanism are proven essential for providing the required level of security to the data. Encryption is a technique used to convert data into code, to prevent unauthorized access and Decryption is the process of transforming data that has been rendered unreadable through encryption back to its unencrypted form. Over the years many Encryption and Decryption techniques have been proposed for the protection of data.

For the implementation of this particular project, the domain of Bluetooth technology and password mechanism is chosen for providing the user with

two-level authentication and AES encryption mechanism is used for preserving the privacy of datas in the files and the applications.

1.2 BLUETOOTH

A Bluetooth technology is a high speed low powered wireless technology link that is designed to connect phones or other portable equipment together. It is a specification (IEEE 802.15.1) for the use of low power radio communications to link phones, computers and other network devices over short distance without wires. Wireless signals transmitted with Bluetooth cover short distances, typically up to 30 feet (10 meters).

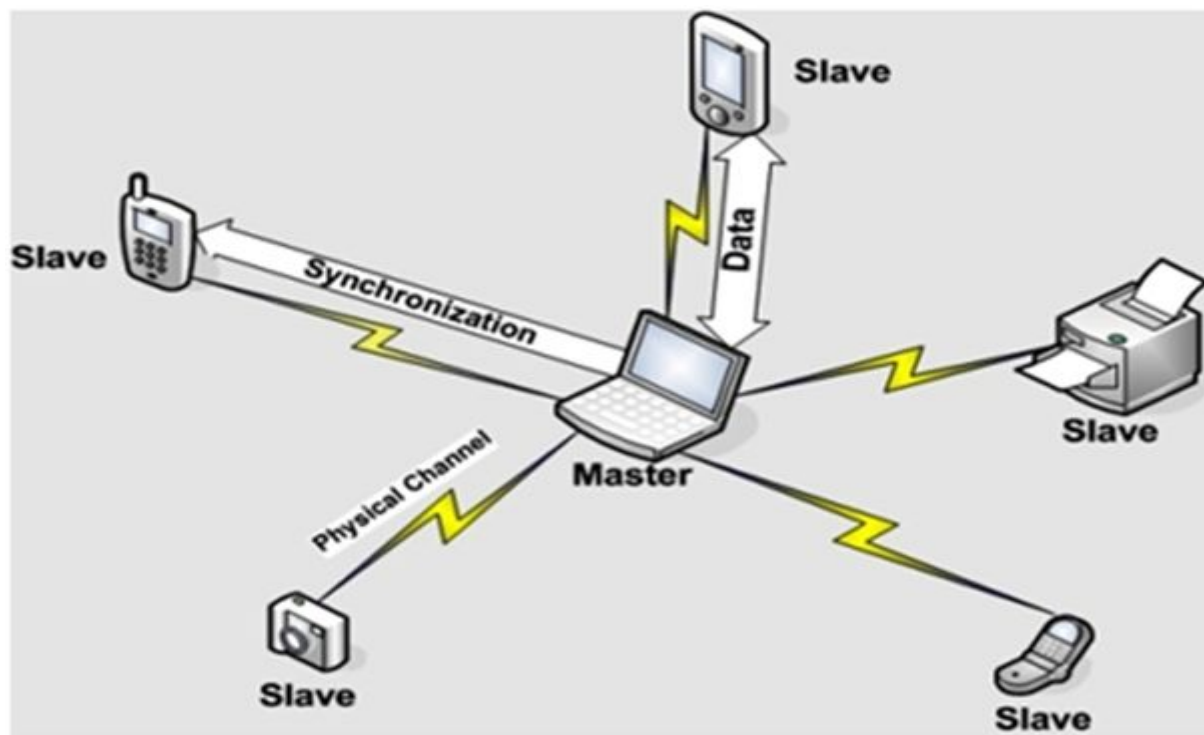
It is achieved by embedded low cost transceivers into the devices. It supports on the frequency band of 2.45GHz and can support up to 721 Kbps along with three voice channels.

Bluetooth can connect up to “eight devices” simultaneously and each device offers a unique 48 bit address from the IEEE 802 standard with the connections being made point to point or multipoint. The 48 bit address is known as Bluetooth Device Address, commonly abbreviated BD_ADDR. This address is represented as a 12-digit hexadecimal value in which the most-significant half (24 bits) of the address identifies the manufacturer and the lower 24-bits are the unique part of the address. This address is used for frequency synchronization of the devices.

1.2.1 How Bluetooth Works

Bluetooth Network consists of a Personal Area Network or a piconet which contains a minimum of 2 to maximum of 8 Bluetooth peer devices- Usually a single master and up to 7 slaves. A master is the device which initiates communication with other devices. The master device governs the

communications link and traffic between itself and the slave devices associated with it. A slave device is the device that responds to the master device. Slave devices are required to synchronize their transmit/receive timing with that of the masters. In addition, transmissions by slave devices are governed by the master device (i.e., the master device dictates when a slave device may transmit). Specifically, a slave may only begin its transmissions in a time slot immediately following the time slot in which it was addressed by the master, or in a time slot explicitly reserved for use by the slave device. Figure 1.1 projects the overall



Bluetooth architecture discussed above.

Figure 1.1 –Bluetooth Architecture

The frequency hopping sequence is defined by the Bluetooth device address (BD_ADDR) of the master device. The master device first sends a radio signal asking for response from the particular slave devices within the range of addresses. The slaves respond and synchronize their hop frequency as well as clock with that of the master device.

1.3 BENEFITS OF USING BLUETOOTH FOR PROVIDING SECURITY

As already discussed, the uniqueness Bluetooth Device Address provides a best platform for achieving two-level authentication and a personalized way for securing their privacy.

The fact that all the hand-held devices are embedded with Bluetooth and the increasing usage of hand-held devices helps the proposed system to provide personalized security to ones' files, folders and applications.

1.4 AES Algorithm

The Advanced Encryption Standard (AES), also cited as Rijndael (its original name), is an encryption of electronic data specification which was established by the U.S. National Institute of Standards and Technology in the year 2001. AES is grounded on the Rijndael cipher developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, who presented a subject matter to NIST during the AES selection process. The AES algorithm is a symmetric block cipher that can process data blocks of 128 bits using cryptographic keys of 128, 192 and 256 bits. AES algorithm was preferred for this implementation since it is widely considered very difficult to solve. Also performance of AES algorithm is complex which makes it difficult to crack. Figure 1.2 shows the overall working of the AES Algorithm.

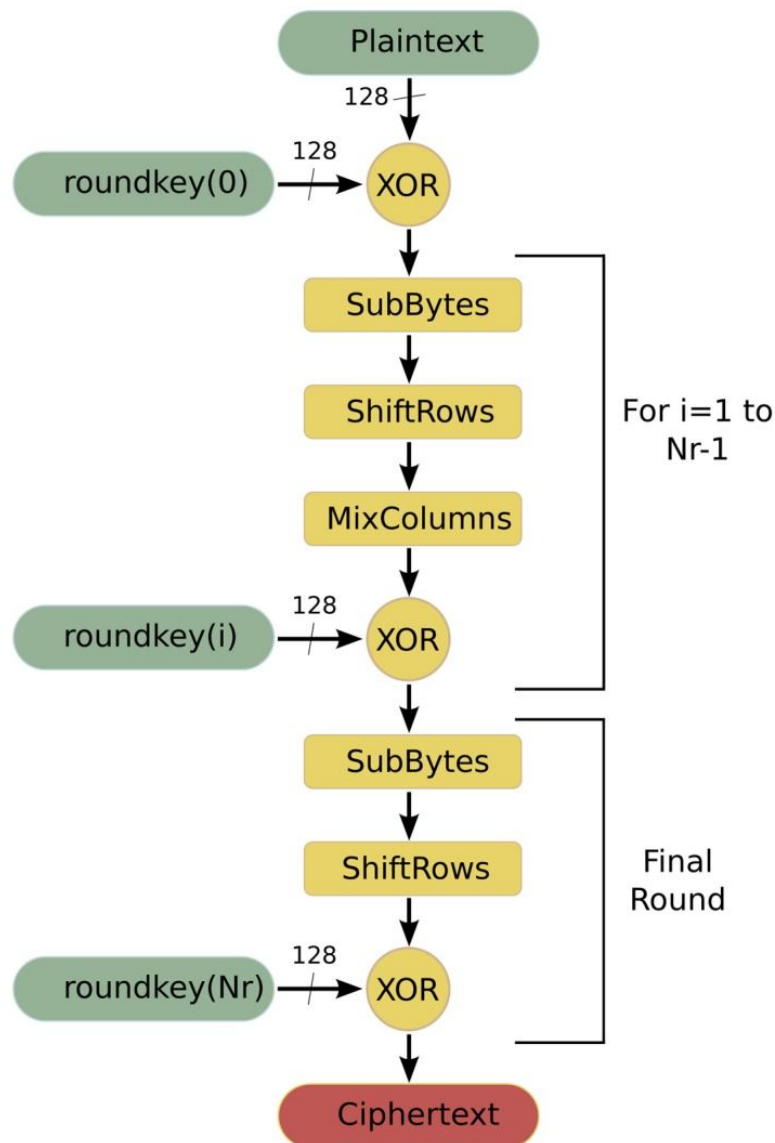


Figure 1.2 –Workflow of AES Algorithm

As shown in Figure 1.2 AES is one of modern symmetrical cryptography algorithm, which has 4 processes in each round:

- I. SubBytes() Transformation,
- II. Shift()Rows Transformation,
- III. MixColumns() Transformation,
- IV. AddRoundKey() Transformation

1.4.1 SubBytes() Transformation

The SubBytes() transformation is a nonlinear byte substitution that operates independently on each byte of the state using a S-box as shown in Figure 1.3.

For example, if $s_{2,1} = \{8f\}$, then the substitution value is determined by the intersection of the row with index 8 and the column with index f. The resulting $s'_{2,1}$ would be a value of $\{73\}$. Where individual byte of the state is referred to as either $S_{r,c}$ and the result after passing through S-box is represented as $S'_{r,c}$.

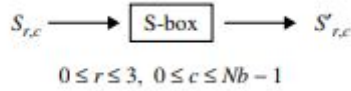


Figure 1.3 –SubBytes() transformation by the S-box

1.4.2 Shift()Rows Transformation

In the ShiftRows(), the first row (row 0) is not shifted and the remaining rows proceed as follows:

$$s'_{r,c} = s'_{r,(c+\text{shift}(r,Nb)) \bmod Nb} \text{ for } 0 < r < 4 \text{ and } 0 \leq c < Nb$$

where the shift value $\text{shift}(r, Nb) = \text{shift}(r, 4)$ depends on the row number r as follows:

$$\text{shift}(1, 4) = 1; \text{shift}(2, 4) = 2; \text{shift}(3, 4) = 3;$$

This has the effect of shifting the leftmost bytes around into the rightmost positions over different numbers of bytes in a given row.

1.4.3 MixColumns() Transformation

The MixColumns() transformation operates on the state column-by-column, treating each column as a four-term polynomial over $GF(2^8)$

and multiplied modulo $x^4 + 1$ with a fixed polynomial $a(x)$ as:

$$s'(x) = a(x) \otimes s(x)$$

where $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$, $s(x)$ is the input polynomial and $s'(x)$ is the corresponding polynomial after the MixColumns() transformation.

The matrix multiplication of $s(x)$ is

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \text{ for } 0 \leq c < Nb$$

The four bytes in a column after the matrix multiplication are

$$s'_{0,c} = (\{02\} \cdot s_{0,c}) \oplus (\{03\} \cdot s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$

$$s'_{1,c} = s_{0,c} \oplus (\{02\} \cdot s_{1,c}) \oplus (\{03\} \cdot s_{2,c}) \oplus s_{3,c}$$

$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \cdot s_{2,c}) \oplus (\{03\} \cdot s_{3,c})$$

$$s'_{3,c} = (\{03\} \cdot s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \cdot s_{3,c})$$

1.4.4 AddRoundKey() Transformation

In AddRoundKey() transformation, a round key is added to the state by a simple bitwise XOR operation. Each round key consists of Nb words from the key schedule. These Nb words are added into the columns of the state such that

$$[s'_{0,c}, s'_{1,c}, s'_{2,c}, s'_{3,c}] = [s_{0,c}, s_{1,c}, s_{2,c}, s_{3,c}] \oplus [w_{\text{round} \cdot Nb + c}] \text{ for } 0 \leq c < Nb$$

where $[w_i]$ are the key schedule words, and round is a value in the range $0 \leq \text{round} \leq \text{Nr}$. The initial round key addition occurs when $\text{round} = 0$, prior to the first application of the round function. The application of the AddRoundKey() transformation to the Nr rounds of the cipher occurs when $1 \leq \text{round} \leq \text{Nr}$.

2. LITERATURE REVIEW

In recent time there has been an immense improvement in the field of security. Many new and advanced algorithms have been proposed in-order to improve the security using encryption and decryption. The need and usage of Bluetooth technology in various fields has also seen an impressive growth. This section will be reviewing on various works related to Bluetooth technology and File security algorithms.

2.1 RELATED WORKS

2.1.1 Bluetooth Remote Control

Alhakim.M.M et al. [1] proposed an application based on Bluetooth. They came up with a system which acts as a remote control that can be used to control the target device via Bluetooth. Their system uses the client-server mechanism to establish control of the target device.

2.1.2 Using Visual Tags to Bypass Bluetooth Device Discovery

Scott David, et al. [5] suggested an idea of how Bluetooth technology can be used for other purposes other than file transmission. They proposed a system which can be used in stores for effective interaction between the customer and the shop. The customer can get an update if a product is available or not, using this system. Their system used Bluetooth Device address to identify a customer and communicate with them. Their results proved how efficient and effective the Bluetooth technology is and how it is to be used.

2.1.3 Study of file encryption and decryption system using security key

Gang Hu [2] proposed an idea on Encryption and Decryption based on security key. The system uses both symmetric and asymmetric cryptosystem for

encryption and decryption of files. Symmetric key cryptography is used to encrypt a file while asymmetric key cryptography is used to encrypt the secret key.

2.1.4 File Encryption and Decryption System Based on RSA Algorithm

Suli Wang and Ganlai Liu [6] proposed a paper on file encryption and decryption using asymmetric key cryptosystem. They used RSA cryptographic algorithm for file encryption and decryption. The paper provides a detailed explanation of working and efficiency of RSA algorithm. The result provided the working of public-key cryptosystem and provides various advantages and disadvantages of using RSA algorithm.

2.1.5 Study of Securing Computer Folders with Bluetooth

Nadargi. A. V. et al. [3] proposes a paper on how to secure files in computers using Bluetooth. The proposed system uses Bluetooth as a key for file encryption and decryption. The system uses Rijndael algorithm for encryption and decryption of files. The results proved how effective the Bluetooth technology can be used to improve security.

2.2 EXISTING MECHANISMS AND APPLICATIONS

2.2.1 PASSWORD MECHANISM

A password is a word or string of characters used for user authentication to prove identity or access approval to gain access to a resource, which is to be kept secret from those not allowed access. This is the basic method used for authentication of a user.

2.2.1.1 Issues in Password Mechanism

1. Guessing password is too easy these days. It can either be your name, phone number, your birthday etc.
2. Dictionary attacks can be used to compromise common passwords
3. Username, and passwords are stored in central servers which gives hackers the flexibility to attack a specific server
4. Difficult passwords are written down somewhere to be remembered which makes it more vulnerable
5. Many passwords are reused in other websites. So, if an attacker knows your login details of one website, chances are the attacker can access your website with the same login.

2.2.2 FOLDER LOCK APPLICATIONS

Folder Lock Applications are used to Lock and Hide files and folders within seconds. It enables you to Password Protect and restricts the unwanted eyes from viewing files, folders and drives. Once locked, a folder will be hidden from its previous location and can only be accessed through the software interface.

2.2.2.1 Issues in Folder Lock Applications

1. Provides only one level of authentication.
2. Supports only one user per system.

3. SYSTEM REQUIREMENTS

The software and hardware requirements for proposed application is listed below,

3.1 SOFTWARE REQUIREMENTS

- JAVA JRE
- MySQL (For Database)

3.2 HARDWARE REQUIREMENTS

- A Bluetooth Device
- A Computer embedded with Bluetooth.
- Windows OS X86 or X64

4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

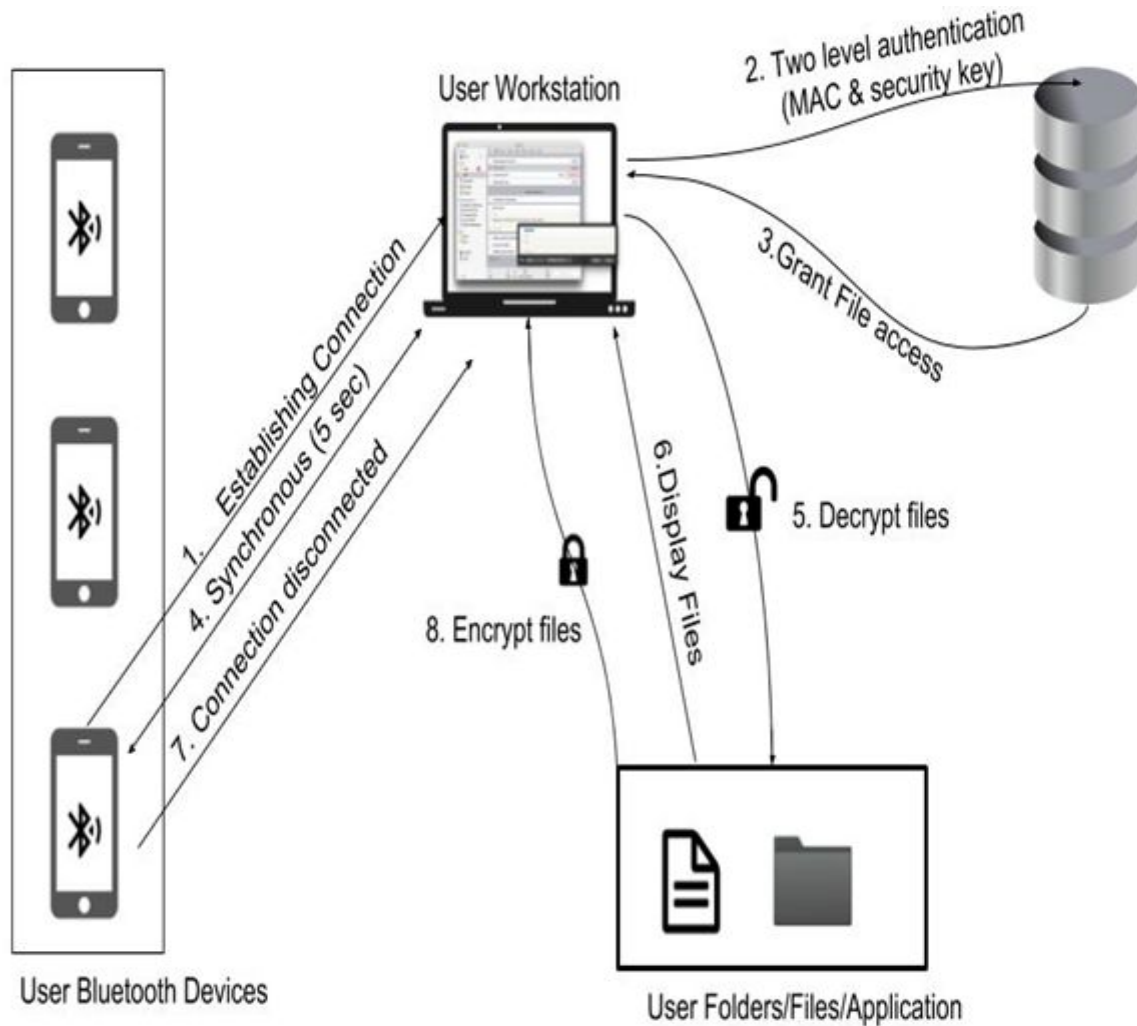


Figure 4.1 – System Architecture

The system architecture for “File and Application Security using Bluetooth” is represented in the Figure.4.1.

The architecture of “File and Application Security using Bluetooth” includes four modules as listed below.

- (i) Connection Establishment and Synchronization
- (ii) Authentication
- (iii) File Security and File Access
- (iv) User Interface

4.1.1 Connection Establishment and Synchronization

This module is responsible for establishing connection between the Bluetooth device and the workstation (users’ computer) and synchronizing the devices. The connection establishment is done using Bluetooth protocols and client-server mechanism. For the connection to be successful, the devices must be within the range of 10-meter radius from the workstation. Once the connection is established, it is important for the devices to be in constant sync. This is achieved by the workstation by running a program in its background which keeps in check if the Bluetooth device is active and in range. The user will not be able to access the system if the Bluetooth device is turned off or in case of the Bluetooth device being out of range.

4.1.2 Authentication

The Authentication module is responsible for checking if a user is legitimate in order to grant access to their files, folders and applications. This module first validates the user by checking the Bluetooth Device Address of the connected device with respect to the Bluetooth Device Address in the database. If a valid device is connected the system prompts for the “Username” and “Password”, and validates it. The successful validation of “Bluetooth Device

Address”, “Username” and “Password” results in granting the access rights to the users files, folders, and applications.

4.1.3 File Security and File Access

File Security and File Access module is responsible for providing all the security and access features for the proposed system. This module gets the users option for the files to be encrypted. If a file or folder is added to the system, then the added files or folders are encrypted using AES algorithm and the access rights for the selected files are revoked till the user presses the release option. When the user removes a file or folder from the system, the selected file or folder is decrypted and the access rights are re-initiated to the user for their access. When the user logouts or when the Bluetooth device moves out of range, the system revokes the user’s rights to add or remove files, folders or applications from the system.

4.1.4 User Interface

User Interface module can be said as the communication medium between the user and the workstation. This module gets the users option on what the system should do and gives the control to the respective modules according to users choice. Once the users request has been processed, this module acquires the result and displays it to the user. This module is also responsible for guiding the user on how to proceed in case of an error.

4.2 UML DIAGRAMS

4.2.1 Use Case Diagram

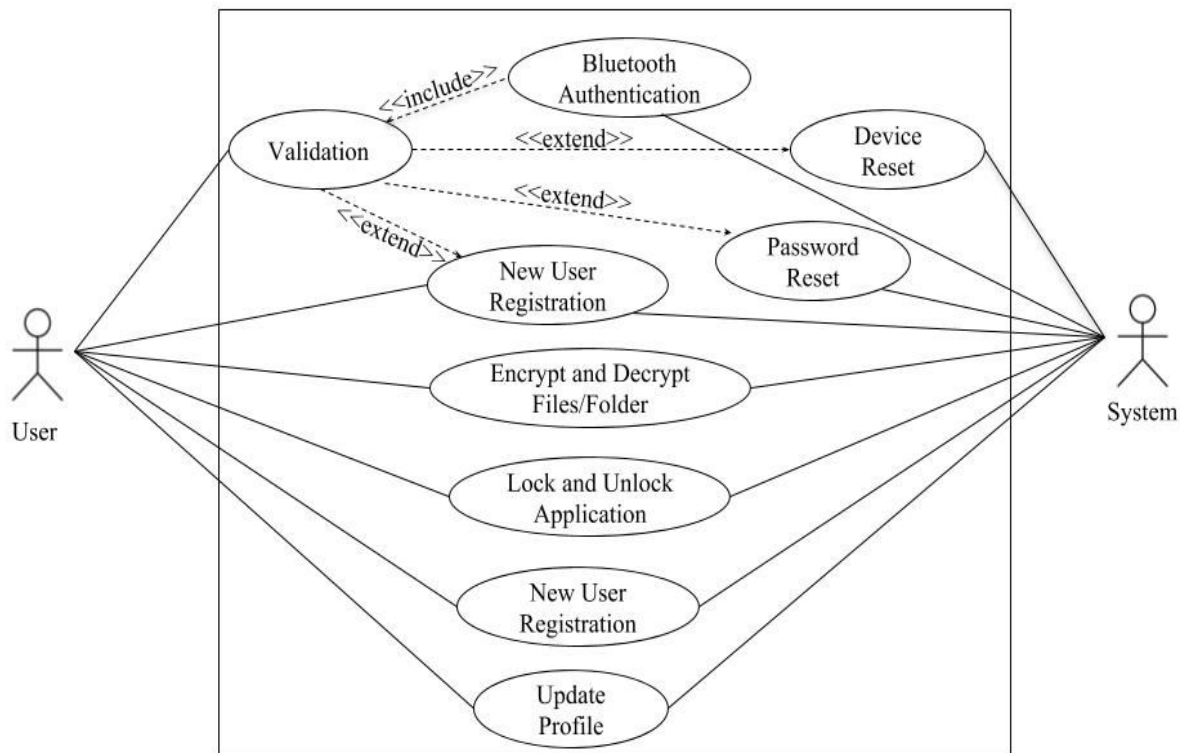


Figure 4.2– System Use-Case Diagram

The Figure 4.2 represents the use-case diagram for “File and Application Security Using Bluetooth”. It represents the user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

4.2.2 ACTIVITY DIAGRAM

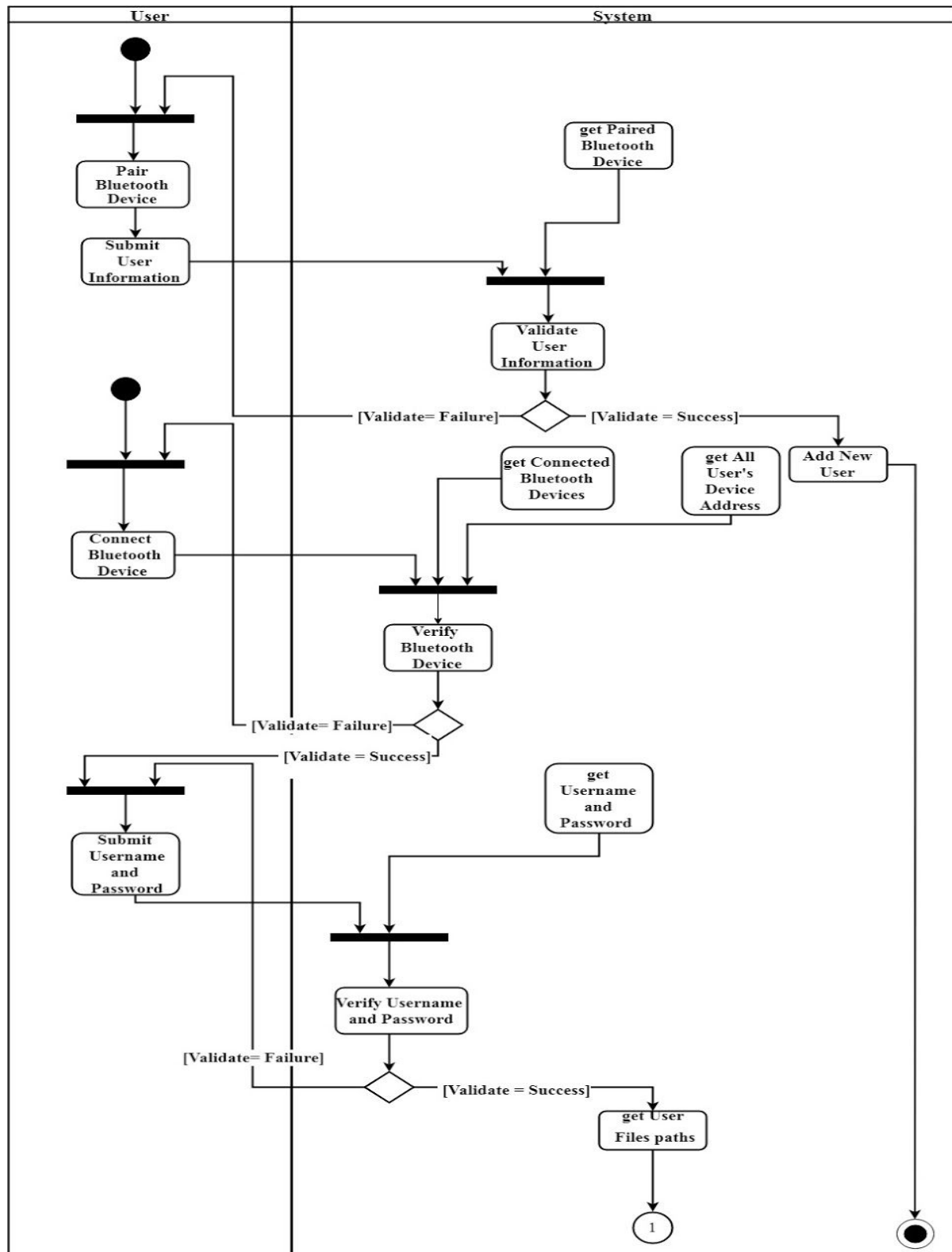


Figure 4.3.1–Activity Diagram for User validation and verification

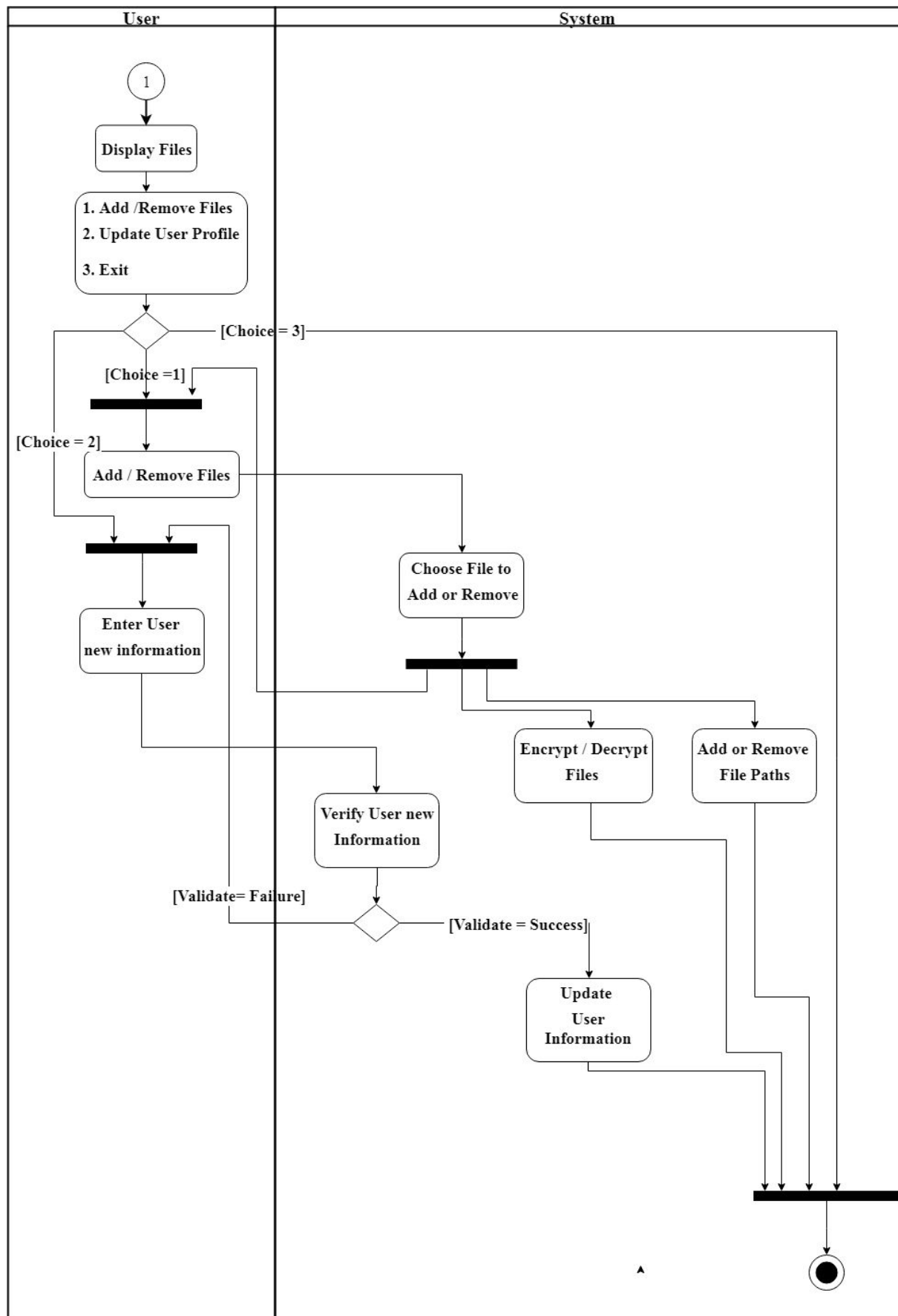


Figure 4.3.2-Activity Diagram for User File Access

4.3. METHODOLOGY

The proposed system is implemented as a java application for working across various operating systems. The system is created using java code for the backend and JavaFX for the frontend. The user detail for authentication is stored in user's individual system using MySQL database.

When the application is executed, it starts off with checking for Bluetooth connection. The system searches for any and all paired connections of registered users. If such known device is found, the Bluetooth Device Address is validated with respect to the data in the database. The validation of Bluetooth Device Address is the first level of authentication of the system. In case of no such user being found, the user is given three options. The user can refresh and check for connection, the user can recover their file in case the connection is not established as the user device is lost, or the user can register with the system as "New user", and create his private profile. This process of connection and establishing bluetooth device is done with the help of the client-server mechanism, where the user's computer acts as client and the users unique bluetooth device acts as the server.

The system registers a new user by getting his "Bluetooth Device Address", "Email-ID", "Username" and "Password". Once the user successfully registers with the system he or she proceeds to the "Login option". Here, the "Username" and "Password" is validated. This validation is the second level of authentication provided by the system.

The successful validation of "Username" and "Password" grants rights to access the system. At the same time, a asynchronous Bluetooth checking program is started, to check if the user is active. If the Bluetooth Device is inactive, the system is terminated.

When the user is granted rights to access the system, the user can add or remove files, folders and applications to the system. Adding files or folders results in its encryption using AES algorithm and revoking of its access rights till it is removed from the user. Removal of the files and folders ends up in decrypting the selected file or folder and granting back the user access rights to those files or folders. Adding an applications to the system results in locking the application by revoking the access rights. The access rights can be reinstated by removing the application from the system.

The system also provides the user with options of editing their profile. The user can change their “Username”, “Password”, “Device” or “Email-ID” using this feature.

The user may have a query on how to recover their files, folders and applications in case their device is lost. The system provides a solution for this query. The system presents an “Device Lost” option to the user. This option takes the user details like “User Name”, “Email-ID” and “Password”. Upon successful validation of user details, the system checks for internet connection. If the internet connection is available, a recovery code is sent to the users Email. The user is to enter the correct recovery code which enables the user to select an alternate Bluetooth device for future access of his or her account.

5. FUNCTIONAL DESCRIPTION

The application has several components that use class-object based entity relationship. Each of these components represents a distinct functionality. This was done so that any changes to a particular component is universal instead having the need to change the individual functionalities. This allows more flexibility and modularity to the application.

The systems are,

1. BLUETOOTH
2. FILE ENCRYPTION AND DECRYPTION
3. DATABASE
4. USER CONTROL
5. GRAPHICAL USER INTERFACE

Each of these components are described in detail below,

5.1 BLUETOOTH

This component provides the system with all the Bluetooth related details. The main functionality of this component is to recover the users Bluetooth Device Address and the device name of the connected devices from the Bluetooth hardware in the users computer.

Initially this component checks for any Bluetooth device paired with the computer. If a device is found, then its Bluetooth Device Address is recorded for user's authentication purpose.

This component is also responsible for checking if the user is in sync. It checks for user's availability repeatedly for each time interval of five seconds. The system is disconnected whenever the users Bluetooth device is turned off or disconnected.

5.1.1 BLUETOOTH CODE SNIPPET

```
public class RemoteDeviceDiscovery {
    public Vector getDevices() {
        /* Create Vector variable */
        final Vector devicesDiscovered = new Vector();
        try {
            final Object inquiryCompletedEvent = new Object();
            /* Clear Vector variable */
            devicesDiscovered.clear();
            /* Create an object of DiscoveryListener */
            DiscoveryListener listener = new DiscoveryListener() {
                public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
                    /* Get devices paired with system or in range(Without Pair) */
                    devicesDiscovered.addElement(btDevice);
                }
                public void inquiryCompleted(int discType) {
                    /* Notify thread when inquiry completed */
                    synchronized (inquiryCompletedEvent) {
                        inquiryCompletedEvent.notifyAll();
                    }
                }
            };
            /* To find service on bluetooth */
            public void serviceSearchCompleted(int transID, int respCode) {
            }
            /* To find service on bluetooth */
            public void servicesDiscovered(int transID, ServiceRecord[] servRecord) {
            }
        };
        synchronized (inquiryCompletedEvent) {
            /* Start device discovery */
            boolean started = LocalDevice.getLocalDevice().getDiscoveryAgent()
                .startInquiry(DiscoveryAgent.GIAC, listener);
            if (started) {
                System.out.println("wait for device inquiry to complete...");
                inquiryCompletedEvent.wait();
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    /* Return list of devices */
    return devicesDiscovered;
}
```

5.2 FILE ENCRYPTION AND DECRYPTION

This component is responsible for providing security for user's files, folders, and applications. Whenever a user file or folder is added, it is encrypted using AES algorithm and decryption is done when the user removes the file or folder from the system. The user's applications are made hidden and the access rights are revoked when they are added to the system.

5.2.1 ENCRYPTION AND DECRYPTION CODE SNIPPETS

File Encryption:

```
private void encrypt(String filePath)
/* Takes filePath and password as argument and creates a encrypted file by name filePath.enc*/
    try
    {
        try (FileInputStream fis = new FileInputStream(filePath) // file to be encrypted
            ; FileOutputStream fos = new FileOutputStream(filePath+".enc") // encrypted file
        ) {
            byte[] keyBytes = getKey(); // get md5 hash of the password
            SecretKeySpec key = new SecretKeySpec(keyBytes,"AES"); // secret key for aes
            Cipher cipher = Cipher.getInstance("AES"); // get cipher
            cipher.init(Cipher.ENCRYPT_MODE,key); // init cipher
            try (CipherInputStream cis = new CipherInputStream(fis,cipher) // get cipher
                //input stream, to read the file
            ) {
                byte buf[] = new byte[1024];
                int len;
                while( (len = cis.read(buf,0,buf.length)) != -1) // read till end
                {
                    fos.write(buf,0,len); // write out to encrypted file
                }
            }
        }
    }
    catch (IOException | InvalidKeyException | NoSuchAlgorithmException |
        NoSuchPaddingException e) // catch all exceptions #TODO proper cleanup
    {
        System.out.println("Error in encryption encrypt(String filePath) " + e); } }
```

File Decryption:

```
private void decrypt(String filename)
{ /*Takes filename and password as arguments and decrypts file "filename.enc" stores it as filename*/
    try
    { try (FileInputStream fis = new FileInputStream(filename+".enc") // file to be decrypted
        ; FileOutputStream fos = new FileOutputStream(filename) // decripted file
        ) {
            byte[] keyBytes = getKey(); // get md5 hash of the password
            SecretKeySpec key = new SecretKeySpec(keyBytes,"AES"); // secret key for aes
            Cipher cipher = Cipher.getInstance("AES"); // get cipher
            cipher.init(Cipher.DECRYPT_MODE,key); // init cipher
            try (CipherInputStream cis = new CipherInputStream(fis,cipher) // get chiper input
                //stream, to read teh file
            ) {
                byte buf[] = new byte[1024];
                int len;
                while( (len = cis.read(buf,0,buf.length)) != -1) // read till end
                {
                    fos.write(buf,0,len); // write out to encrypted file
                }
            }
        }
    }
    catch (IOException | InvalidKeyException | NoSuchAlgorithmException | NoSuchPaddingException
    e) // catch all exceptions #TODO proper cleanup
    {
        System.out.println("Error in decryption decrypt"+e);
    }
}
```

Get Key:

```
private byte[] getKey()
{
    /*
     * Takes a string as input and returns its MD5 in byte[]
     */
    try
    {
        MessageDigest md5 = MessageDigest.getInstance("MD5"); // get the MD5 instance
        return md5.digest(password.getBytes()); // get md5 of key ( byte[] )
    }
    catch(NoSuchAlgorithmException e)
    {
        System.out.println("Error in decryption getKey"+e);
    }
    return new byte[1];
}
```


Make ZIP file:

```
private void encryptFile(String filePaths) throws ZipException, IOException {
    try{
        String filePath=filePaths+".zip";
        Path path = Paths.get(filePath+".enc");
        ZipFile zipFile = new ZipFile(filePath);
        ZipParameters parameters = new ZipParameters();
        parameters.setCompressionMethod(Zip4jConstants.COMP_DEFLATE);
        // Set the compression level
        parameters.setCompressionLevel(Zip4jConstants.DEFLATE_LEVEL_NORMAL);
        parameters.setEncryptFiles(true);
        parameters.setEncryptionMethod(Zip4jConstants.ENC_METHOD_AES);
        parameters.setAesKeyStrength(Zip4jConstants.AES_STRENGTH_256);
        parameters.setPassword(password);
        // Add folder to the zip file
        if(filePaths.indexOf(".")>=0)
        {
            File file=new File(filePaths);
            zipFile.addFile(file, parameters);
            FileUtils.forceDelete(new File(file.toString()));
        }
    }
    else{
        String folderToAdd = filePaths;
        zipFile.addFolder(folderToAdd, parameters);
        FileUtils.forceDelete(new File(folderToAdd));
    }

    encrypt(filePath);
    Process process= runtime.exec("cmd.exe /c start attrib +s +h +a "+filePath+".enc");
    FileUtils.forceDelete(new File(filePath));
    System.out.print("Done encryption");
}
catch (IOException | ZipException e) {
    System.out.println("Error in encryption encryptFile(String filePaths) " + e);
}
}
```

5.3 DATABASE

This component provides an important functionality for storing and retrieving the users' progress. The application data is stored and retrieved using this component. This component is implemented using MySQL database. The database contains User data like 'User Name', 'Password' and 'User Device Address' in addition to file and application details like 'Path' in which the file is stored and the 'User' to whom the file belongs to.

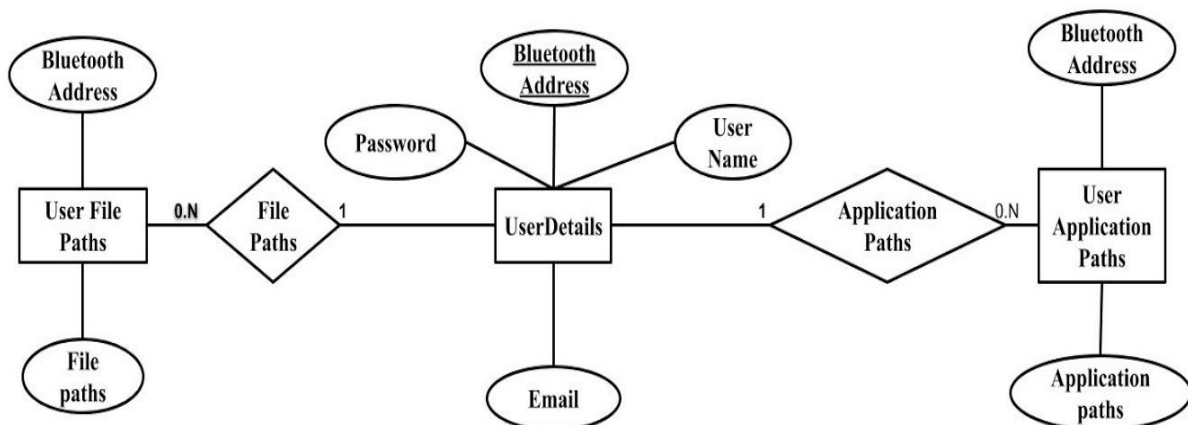


Figure 5.1– ER -diagram

5.4 USER CONTROL

This provides the most important functionality of the system. This is the one which integrates the user's actions in the user interface with the background functionalities. This acts as an intermediate medium between the user and the system.

5.5 GRAPHICAL USER INTERFACE

This is the frontend of the system. This is achieved by using JavaFx. This is the interface using which the user communicates with the system. The user can control the system using this interface by just clicking or entering details and the output for users input is displayed in this interface.

6. IMPLEMENTATION AND RESULTS

The core implementation of the project is done using Asynchronous Tasks since the events require a stable Bluetooth connection. This is because applications continuously checks for Bluetooth availability in the background and hence the UI cannot be updated or refreshed while this is happening.

6.1 CHECKING FOR BLUETOOTH CONNECTIVITY

The application performs a check for Bluetooth connectivity during its start up. This is done to ensure that the user has synced a Bluetooth device with the application. If a valid device is connected the system proceeds to user password validation else the user is requested to check the connectivity and refresh it.

The user is also provided with an option of changing the device in case of device being lost and a new user option for the users to create an account with the system.

The user interface is implemented using a thread where the frontend shows the load screen while the backend checks for Bluetooth device. Figure 6.1 shows the Loading screen, which is displayed when the system is searching and establishing connection with a Bluetooth device in the background.

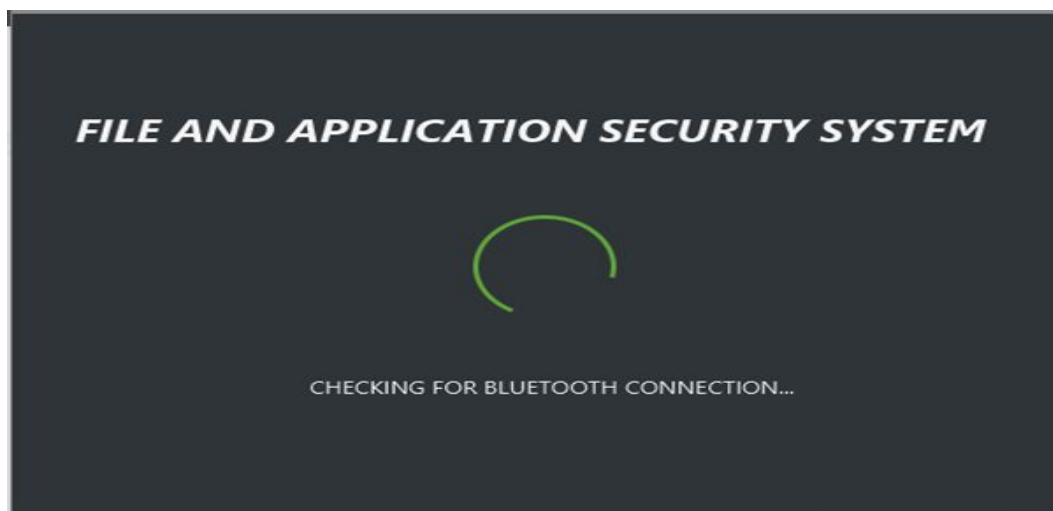
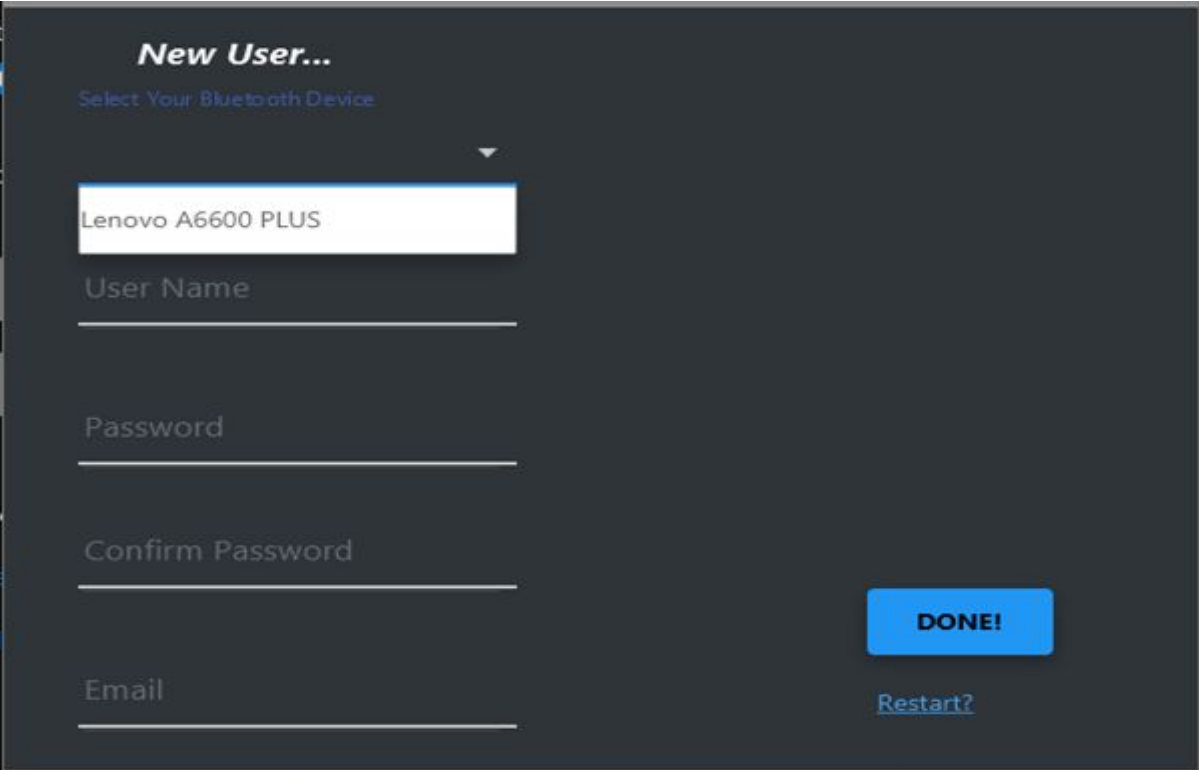


Figure 6.1– Checking for Bluetooth Page

6.2 NEW USER

A user can sign up with the system using this option. The user is requested to select his or her device from a list of connected device, and then the user is required to enter their 'Username', 'Email' and 'Password'. Once the user signs up they are ready to use the system. Figure 6.2 shows the new user page of the system.



New User...
Select Your Bluetooth Device

Lenovo A6600 PLUS

User Name

Password

Confirm Password

Email

DONE!

[Restart?](#)

Figure 6.2– New User Page

6.3 PASSWORD VALIDATION

The user needs to provide his or her username and password after which the username and password is compared with the connected Bluetooth device for user validation. The validation is done with respect to the data stored in the database. The user is also provided with an option of forgot password which is

recovered by validating a secret code sent through Email. Figure 6.3 depicts the details mentioned above.

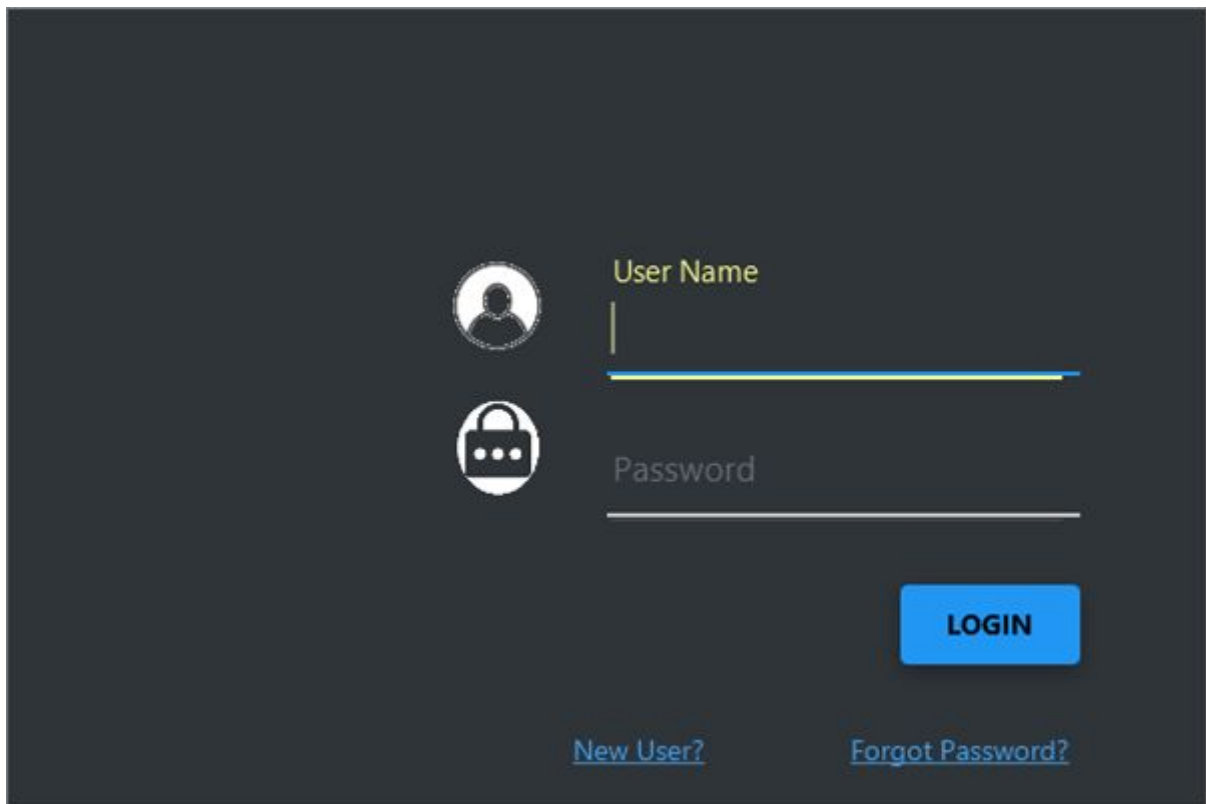


Figure 6.3– Login Page

6.4 SYSTEM ACCESS

This is the page which shows all the actions that can be performed by the user. This page is in form of tab view where the user can perform actions on File/Folder, Applications, and Users Profile.

File/Folder tab provides the user with ‘Add Files’ option, ‘Add Folder’ option and ‘Remove’ option. These enable the user to add and remove files or folders from the system.

Applications tab is used for locking the applications. The user can add Applications to lock them or remove to unlock them.

User Profile tab provides users with option to edit their user details. The edited details are reflected in the database ones the user action is performed. Figure 6.4.1, 6.4.2 and 6.4.3 shows the users System Access page.

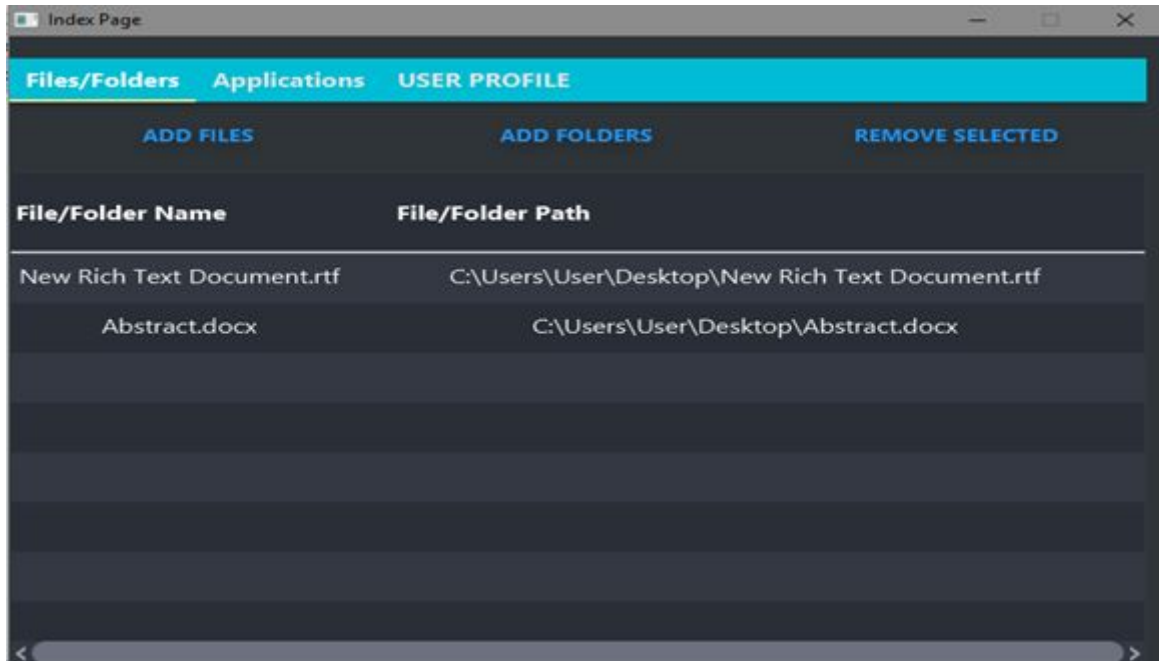


Figure 6.4.1-Files/Folders Page

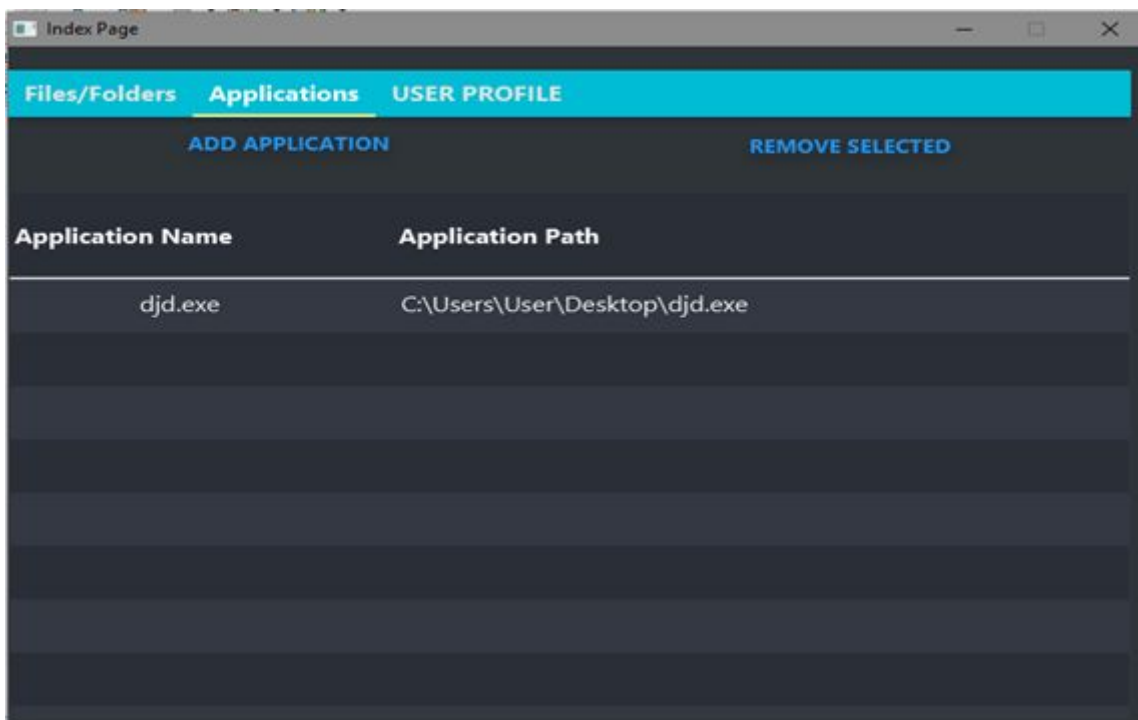


Figure 6.4.2-Applications Page

The screenshot shows a web application window titled "Index Page". It features a navigation bar with three tabs: "Files/Folders", "Applications", and "USER PROFILE", with the latter being the active tab. The main content area is a user profile form with the following fields and controls:

- Current Device:** A text input field containing "shrikara" and a blue "CHANGE" button to its right.
- User Name:** A text input field containing "shriks".
- Email ID:** A text input field containing "shrikara1996@gmail.com".
- Current Password:** A text input field with masked characters (dots). To its right are two buttons: a blue "CHANGE" button and a grey "SET NEW PASSWORD" button.
- New Password:** A text input field.
- Confirm Password:** A text input field.
- Bottom Controls:** Two blue buttons, "CANCEL" and "APPLY", are positioned at the bottom right of the form.

Figure 6.4.3-Users Profile Page

6.5 RECOVER DEVICE CONTENT

This option is triggered by user in-order to recover their files, in case of the device being lost. The user needs to provide the system with their username, password and Email-ID. The system checks for internet connection and validates the input given by the user. In case of successful validation a recovery code is sent to user's mail using which he or she can set a new device.

Figure 6.5.1 and 6.5.2 shows the validation and verification for recovering device content.

DEVICE RECOVERY

User Name

Email ID

Password

REFRESH **VALIDATE**

Figure 6.5.1-Validate User Page

DEVICE RECOVERY VALIDATION

RECOVERY CODE _____ **VERIFY**

SELECT YOUR NEW DEVICE ▼ **APPLY**

Figure 6.5.2-Device Recovery Page

6.6 DISCUSSION

The application is, however, not bug free with minor issues that pop up when the Bluetooth signal is low. These are not irrecoverable and hence are handled using exception wherever necessary.

The major issue that occurred during the testing of the application under critical circumstances when a large file or folder is required to be encrypted or decrypted. This process is time consuming hence the application being unresponsive till the encryption or decryption is done.

7. CONCLUSION AND FUTURE ENHANCEMENTS

Security of files is an important issue in places where multiple users share the same computer. The proposed system would play an important role such places as an effective file security system. This system will safeguard a user's privacy over their files, folders and applications. The proposed system overcomes the disadvantages in the password mechanism and provides maximum security to one's files, folders, and application.

The proposed system can be further improved by making a user's files available in remote workstations over internet instead of availability in a single workstation. Security can be further improved by using more advanced cryptographic techniques.

REFERENCES

- [1] Alhakim.M.M, I. Al-Kittani, A. Bakleh, M. Swidan, N. Zarka, “*Bluetooth Remote Control*”. Information and Communication Technologies, 2006. ICTTA '06.
- [2] Gang Hu, “*Study of file encryption and decryption system using security key*”. Computer Engineering and Technology (ICCET), 2010 2nd International Conference.
- [3] Nadargi, A. V., ApurvaDalmiya, SonaliJadhav, and Gajendra Singh Solanki. "Study of Securing Computer Folders with Bluetooth." International Advanced Research Journal in Science, Engineering and Technology Vol 2; February-2015
- [4] Rhee, Man Young. “*Internet security: cryptographic principles, algorithms and protocols.*” John Wiley & Sons, 2003.
- [5] Scott, David, et al. "Using visual tags to bypass Bluetooth device discovery." *ACM SIGMOBILE Mobile Computing and Communications Review* 9.1 (2005): 41-53.
- [6] Suli Wang, Ganlai Liu, “*File Encryption and Decryption System Based on RSA Algorithm*”. Computational and Information Sciences (ICCIS), 2011 International Conference.