Appendix A

During my conversation with Franklin, I gained lots of insight on a problem he has at work. This insight allowed me to arrive at a solution. The consultation went as follows:

Me - *"What problem have you been having at work?"*
Franklin - *"I don't have many problems at work, but the main problem I have is that it is hard to keep track of my passwords."*
Me - *"Okay. What are some ways you solve that problem?"*
Franklin - *"I use Notepad right now for storing them. I wish Notepad was more secure though."*
Me - *"So, what else do you wish Notepad did?"*
Franklin - *"I feel like it does its job of being a notebook well, but I would like an app that would help me keep track of my passwords outside of me just putting them in a notebook."*
Me - *"Okay. What would you want a password management app to have?"*
Franklin - *"I would want it to let me easily enter and view my passwords. Also, I want to be able to edit my passwords."*
Me - *"Okay. Are there any other features you can think of for this password manager?"*
Franklin - *"I think the app should let me copy/paste my passwords when I'm looking at them. Also, it would be cool if the app let me open a URL when I'm looking at a password so I can login easily."*
Me - *"Okay. I can definitely make this app for you to not have to remember all your passwords. Do you think these requirements are good for the finished app* [I showed him the success criteria from the Planning section.] *?"*
Franklin - *"Yeah. If you could do this, it would be great."*
Me - *"Okay, thanks."*

Appendix B

Source code for the program:

```
//Swing packages
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

//Read and Write package
import java.io.*;

//Hashing package
import java.security.*;
import java.nio.charset.*;

class Login extends JFrame {

    JLabel prompt;
    JPasswordField masterPassword;
    JButton loginButton;

    JPanel centerPanel = new JPanel(new FlowLayout());
    JPanel mainPanel = new JPanel(new BorderLayout());
    JPanel buttonsPanel = new JPanel(new FlowLayout());

    static Login myFrame;

    static String mp;
    static final String FILEPATH = "Passwords.txt";

    void initialize() {

        prompt = new JLabel("");
        prompt.setFont(new Font("Segoe UI", Font.PLAIN, 12));
        mainPanel.add(prompt, BorderLayout.NORTH);

        masterPassword = new JPasswordField(15);
        masterPassword.setEchoChar('*');
        centerPanel.add(masterPassword);
        centerPanel.setBackground(new Color(229, 229, 234));
        centerPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

        loginButton = new JButton("Login");
        loginButton.setFont(new Font("Segoe UI", Font.BOLD, 12));
```

```
        loginButton.setFocusable(false);
        buttonsPanel.add(loginButton);
        buttonsPanel.setBackground(new Color(229, 229, 234));
        buttonsPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

        setTitle("Login");
        setSize(215, 175);
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);

        mainPanel.setBackground(new Color(229, 229, 234));
        mainPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

        mainPanel.add(buttonsPanel, BorderLayout.SOUTH);
        mainPanel.add(centerPanel, BorderLayout.CENTER);

        add(mainPanel);
        setResizable(false);

        mp = ReadMasterPassword(FILEPATH);
        if(mp == null) {

            prompt.setText("Please create a master password:");

            loginButton.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                  // Convert char[] to String.
                  String password = new String(masterPassword.getPassword());

                  mp = hexDigest(password, "SHA-256");

                  SaveMasterPassword(mp, FILEPATH);

                  JOptionPane.showMessageDialog(myFrame, "Master password
created. We will log you in now.",
                      "Password Manager Ready", JOptionPane.INFORMATION_MESSAGE);

                  myFrame.dispose();
                  Manager newWindow = new Manager();
          }
            });
        }

        else {

            prompt.setText("Please verify your master password:");
```

```
            loginButton.addActionListener(new ActionListener() {

                public void actionPerformed(ActionEvent e) {

                    // Convert char[] to String.
                    String password = new String(masterPassword.getPassword());

                    String verification = hexDigest(password, "SHA-256");

                    if(verification.equals(mp)) {
                        myFrame.dispose();
                        Manager newWindow = new Manager();
                    }
                    else {
                        JOptionPane.showMessageDialog(myFrame, "Invalid Master
Password. Please Try Again.",
                                "Error Message", JOptionPane.INFORMATION_MESSAGE);
                    }
                }
            });
        }

        setVisible(true);
    }

    public static void main(String[]args) {
        myFrame = new Login();
        myFrame.initialize();
    }

    static String ReadMasterPassword(String FilePath) {
        //if there is no file or the file is empty, then return the null string.
        String MasterPass = null;

        // attempt to read master password from file.
        try {
          BufferedReader in = new BufferedReader(new FileReader(FilePath));
          String s = in.readLine();
          MasterPass = s;
          in.close();
        }
        catch(java.io.FileNotFoundException e) { }
        catch(java.io.IOException t) { }
```

```
        return MasterPass;
      }

      static String hexDigest(String str, String digestName) {
        try {
          MessageDigest md = MessageDigest.getInstance(digestName);
          byte[] digest = md.digest(str.getBytes(StandardCharsets.UTF_8));
          char[] hex = new char[digest.length * 2];
          for (int i = 0; i < digest.length; i++) {
            hex[2 * i] = "0123456789abcdef".charAt((digest[i] & 0xf0) >> 4);
            hex[2 * i + 1] = "0123456789abcdef".charAt(digest[i] & 0x0f);
          }

          return new String(hex);
        } catch (NoSuchAlgorithmException e) {
          throw new IllegalStateException(e);
        }
      }

      private void SaveMasterPassword(String pwd, String FilePath) {
        try
        {
          PrintWriter out = new PrintWriter(new BufferedWriter (new
FileWriter(FilePath)));

          out.println(pwd);
          out.flush();
          out.close();
        }
    catch (IOException e) {}
      }
}
```

```
//Swing packages
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class Manager extends JFrame {

    JFrame frame = new JFrame();

    JLabel label = new JLabel("Click 'Enter' to enter a password, or type the
nickname of the password " +
            "you want to view, then click view:");

    JButton enter = new JButton("Enter");
    JButton view = new JButton("View");
    JTextField nickname = new JTextField(15);

    JPanel centerPanel = new JPanel(new FlowLayout());
    JPanel mainPanel = new JPanel(new BorderLayout());
    JPanel buttonsPanel = new JPanel(new FlowLayout());

    Manager() {

        mainPanel.setBackground(new Color(229, 229, 234));
        mainPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

        label.setFont(new Font("Segoe UI", Font.PLAIN, 12));

        mainPanel.add(label, BorderLayout.NORTH);

        buttonsPanel.setBackground(new Color(229, 229, 234));
        buttonsPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

        enter.setFont(new Font("Segoe UI", Font.BOLD, 12));
        enter.setFocusable(false);
        enter.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                View newViewWindow = new View(false, null);
            }
        });

        buttonsPanel.add(enter);
```

```
        view.setFont(new Font("Segoe UI", Font.BOLD, 12));
        view.setFocusable(false);
        view.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                if(nickname.getText().equals(""))
                    JOptionPane.showMessageDialog(frame, "Please enter a nickname
to view."
                        , "Error Message", JOptionPane.ERROR_MESSAGE);

                else {
                    View newViewWindow = new View(true, nickname.getText());
                }
            }});

        buttonsPanel.add(view);

        mainPanel.add(buttonsPanel, BorderLayout.SOUTH);

        centerPanel.setBackground(new Color(229, 229, 234));
        centerPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

        centerPanel.add(nickname);

        mainPanel.add(centerPanel, BorderLayout.CENTER);

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setTitle("Password Manager");
        frame.setSize(585, 200);
        frame.setResizable(false);
        frame.add(mainPanel);

        frame.setVisible(true);
    }
}
```

```java
import java.util.StringTokenizer;

class UserPassword {

  String nickname;
  String username;
  String password;
  String url;

  UserPassword(String s) {

    SeperateText(s);
  }

  UserPassword() {

  }

  public boolean equals(Object obj) {

    return nickname.equalsIgnoreCase(((UserPassword)obj).nickname);
  }

  private void SeperateText(String s) {

    StringTokenizer tokens = new StringTokenizer(s, ";");

    if (tokens.hasMoreTokens()) nickname = tokens.nextToken();

    if (tokens.hasMoreTokens()) username = tokens.nextToken();

    if (tokens.hasMoreTokens()) password = tokens.nextToken();

    if (tokens.hasMoreTokens()) url = tokens.nextToken();
  }
}
```

```java
//Swing packages
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

//Read/Write package
import java.io.*;
import java.nio.charset.*;

//ArrayList package
import java.util.*;

//Password encryption package (AES Encryption Algorithm)
import javax.crypto.*;
import javax.crypto.spec.SecretKeySpec;

//Byte Array to String Array package
import javax.xml.bind.DatatypeConverter;

// Browser Launching package
import java.net.URI;

class View extends JFrame {

    static final String ENC_KEY = "0e329232ea6d0d73";

    static UserPassword upobject;

    JFrame frame = new JFrame();

    JLabel label = new JLabel("Enter the credentials and click 'Enter' to save
it:");

    JLabel nicknameLabel = new JLabel("Nickname:");
    JLabel usernameLabel = new JLabel("User Name:");
    JLabel passwordLabel = new JLabel("Password:");
    JLabel urlLabel = new JLabel("URL:");

    JButton enter = new JButton("Enter");
    JButton cancel = new JButton("Cancel");
    JButton visit = new JButton("Browse");

    JTextField nicknameField = new JTextField(15);
    JTextField usernameField = new JTextField(15);
    JTextField passwordField = new JTextField(15);
```

```
    JTextField urlField = new JTextField(15);

    JPanel centerPanel;
    JPanel mainPanel = new JPanel(new BorderLayout());
    JPanel buttonsPanel = new JPanel(new FlowLayout());

    View(boolean readOnly, String inputNickname) {
try {
       ArrayList<UserPassword> testList = new ArrayList<UserPassword>();

       ReadPasswords(testList, Login.FILEPATH);

       mainPanel.setBackground(new Color(229, 229, 234));
       mainPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

       label.setFont(new Font("Segoe UI", Font.PLAIN, 12));

       nicknameLabel.setFont(new Font("Segoe UI", Font.PLAIN, 12));
       usernameLabel.setFont(new Font("Segoe UI", Font.PLAIN, 12));
       passwordLabel.setFont(new Font("Segoe UI", Font.PLAIN, 12));
       urlLabel.setFont(new Font("Segoe UI", Font.PLAIN, 12));

       mainPanel.add(label, BorderLayout.NORTH);

       centerPanel = new JPanel();

       GroupLayout layout = new GroupLayout(centerPanel);
       centerPanel.setLayout(layout);
       layout.setAutoCreateGaps(true);
       layout.setAutoCreateContainerGaps(true);

       layout.setHorizontalGroup(
        layout.createSequentialGroup()
           .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
                .addComponent(nicknameLabel)
                .addComponent(usernameLabel)
                .addComponent(passwordLabel)
                .addComponent(urlLabel)
                )
           .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
                .addComponent(nicknameField)
                .addComponent(usernameField)
                .addComponent(passwordField)
                .addComponent(urlField)
                )
```

```
    );
    layout.setVerticalGroup(
       layout.createSequentialGroup()
          .addGroup(layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
             .addComponent(nicknameLabel)
             .addComponent(nicknameField))
          .addGroup(layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
             .addComponent(usernameLabel)
             .addComponent(usernameField))
          .addGroup(layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
             .addComponent(passwordLabel)
             .addComponent(passwordField))
          .addGroup(layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
             .addComponent(urlLabel)
             .addComponent(urlField))
    );

      buttonsPanel.setBackground(new Color(229, 229, 234));
      buttonsPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

      enter.setFont(new Font("Segoe UI", Font.BOLD, 12));
      enter.setFocusable(false);

      visit.setFont(new Font("Segoe UI", Font.BOLD, 12));
      visit.setFocusable(false);
      visit.addActionListener(new ActionListener() {
       public void actionPerformed(ActionEvent e) {

           if(urlField.getText() == "") JOptionPane.showMessageDialog(frame,
"Please provide a URL to browse.",
             "Error Message", JOptionPane.ERROR_MESSAGE);

           else {
             try {

               URI uri = new URI("https://" + urlField.getText());
               java.awt.Desktop.getDesktop().browse(uri);
             } catch(Exception f) {f.printStackTrace();}
           }
       }
   });

      if(readOnly) {

        enter.setVisible(false);
```

```
        nicknameField.setEnabled(false);

        UserPassword search = new UserPassword();

        search.nickname = inputNickname;

        int itemIndex = testList.indexOf(search);

        if(itemIndex == -1) {

          JOptionPane.showMessageDialog(frame, "The nickname you have entered is
not in your passwords.",
           "Error Message", JOptionPane.ERROR_MESSAGE);

          frame.dispose();

          return;
        }

        else {

         label.setText("Credentials:");
          search = testList.get(itemIndex);

          nicknameField.setText(search.nickname);
          usernameField.setText(DecryptText(search.username, ENC_KEY, "AES"));
          passwordField.setText(DecryptText(search.password, ENC_KEY, "AES"));
          urlField.setText(search.url);
        }
      }

      enter.addActionListener(new ActionListener() {
          public void actionPerformed(ActionEvent e) {

              upobject = new UserPassword();

              upobject.nickname = nicknameField.getText();
              upobject.username = EncryptText(usernameField.getText(), ENC_KEY,
"AES");
              upobject.password = EncryptText(passwordField.getText(), ENC_KEY,
"AES");
              upobject.url = urlField.getText();

              int idx = testList.indexOf(upobject);
```

```
                if(idx > -1) testList.remove(upobject);

                testList.add(upobject);

                SavePasswords(testList, Login.FILEPATH);
                JOptionPane.showMessageDialog(frame, "Successful Entry", "Entry
Message", JOptionPane.INFORMATION_MESSAGE);

                frame.dispose();
            }
        });

        buttonsPanel.add(enter);
        buttonsPanel.add(visit);

        cancel.setFont(new Font("Segoe UI", Font.BOLD, 12));
        cancel.setFocusable(false);
        cancel.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {

             frame.dispose();
         }
     });

        buttonsPanel.add(cancel);

        mainPanel.add(buttonsPanel, BorderLayout.SOUTH);

        centerPanel.setBackground(new Color(229, 229, 234));
        centerPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

        mainPanel.add(centerPanel, BorderLayout.CENTER);

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setTitle("Enter or View");
        frame.setSize(400, 300);
        frame.setResizable(false);
        frame.add(mainPanel);

        frame.setVisible(true);

} catch (Exception e) {
    System.out.println(e.getMessage());
}
```

```
    }

static void ReadPasswords(ArrayList<UserPassword> L, String path) {
    try {
      BufferedReader in = new BufferedReader(new FileReader(path));

      //This "readLine();" is meant to skip the master password.
      String s = in.readLine();

      s = in.readLine();
      while(s != null) {
        UserPassword up = new UserPassword(s);

        L.add(up);
        s = in.readLine();
      }
      in.close();
    }
    catch(java.io.FileNotFoundException e)
{System.out.println("FileNotFoundException");}
    catch(java.io.IOException t) {System.out.println("IOException");}
  }

  private static String EncryptText(String plainText, String key, String
algorithm) {
    String hex = null;
    try {
      Cipher c = Cipher.getInstance(algorithm);
      SecretKeySpec secretKeySpec = new
SecretKeySpec(key.getBytes(StandardCharsets.UTF_8), algorithm);
      c.init(Cipher.ENCRYPT_MODE, secretKeySpec);

      byte[] encBytes = c.doFinal(plainText.getBytes(StandardCharsets.UTF_8));
      hex = DatatypeConverter.printHexBinary(encBytes);
    } catch (Exception ex) {
      System.out.println("Error encrypting data");
      ex.printStackTrace();
    }
    return hex;
  }

  private static void SavePasswords(ArrayList<UserPassword> UPList, String
FilePath) {
    try
    {
```

```
      PrintWriter out = new PrintWriter( new BufferedWriter (new
FileWriter(FilePath)));

      //save the master password on the first line of the file.
      out.println(Login.mp);

      for (UserPassword p : UPList) {
        out.println(p.nickname + ";" + p.username + ";" + p.password + ";" +
p.url);
      }
      out.flush();
      out.close();
    }
    catch (IOException e) { System.out.println("Problem Saving Passwords"); }
  }

  private static String DecryptText(String cipherText, String key, String
algorithm) {
    String decStr = null;
    try {
      Cipher c = Cipher.getInstance(algorithm);
      SecretKeySpec secretKeySpec = new
SecretKeySpec(key.getBytes(java.nio.charset.StandardCharsets.UTF_8), algorithm);
      c.init(Cipher.DECRYPT_MODE, secretKeySpec);

      byte[] encBytes = DatatypeConverter.parseHexBinary(cipherText);
      decStr = new String(c.doFinal(encBytes),
java.nio.charset.StandardCharsets.UTF_8);
    } catch (Exception ex) {
      System.out.println("Error encrypting data");
      ex.printStackTrace();
    }
    return decStr;
  }
}
```