KINGDOM OF SAUDI ARABIA
Ministry of Higher Education
Al-Imam Mohammad University College of Computer & Information Sciences

المملكة العربية السعودية
وزارة التعليم العالي
جامعة الإمام محمد بن سعود الإسلامية
كلية علوم الحاسب والمعلومات

First semester-Fall 2020

Software Engineering (CS- 310)

BSCS- Section: 371

Project-Phase No: 01 , 02 , 03

FYFD

Submitted By

Njood Turki AL-Mukirsh (439020359) – Coordinator

Hind Ahmed Al-Harbi (439017428)

Mayar Weal Al-Shantaf (437000397)

Rand Ibrahim al-Hossaini (439022126)

Reema Nasser Al-Tammami (439021718)

Supervisor

L.Sarah Al-Zahrani

Date: 8 December 2020

Table of Contents

1. Introduction

This section will cover the scope description and overview of everything included in this SRS document, and it also describes all definitions and abbreviations.

1.1 Purpose

The purposeof the Software Requirements Specificationdocumentis to clearly define the system under development and give a detailed description of the requirements for the find your favorite device (FYFD).

The intended audience of this document includes the system administrator and end-users of (FYFD), Other intended audience includes the development team such as the requirements team, requirements analyst, designteam.

1.2 Scope

Our idea is to build an application that helps you to select the best device, a device that meets your need and budget.

Our goal is to make you find the BEST choice of a device among multiple devices in seconds!

How do users benefit from our app?

Our application saves users' time and effort which will be spent in searching for a suitable device for them, by providing them with a list of suitable devices according to their and combine all components in a single convenient page.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
User/ end users	A person is using the app on their mobile phone
API	Application Programming Interface
FYFD	App name find your favorite device
Admin/Administrator	The system administrator who is given specific permission for managing and controlling the system

1.4 References

[1] NotebookCheck, https://www.notebookcheck.net

[2]LaptopMedia, https://laptopmedia.com

[3] Jarir Bookstore, https://www.jarir.com/computers-&-tablets.htm

[4] Newegg, 2001, Newegg Inc., https://www.newegg.com/

2. General Description

2.1 Product Perspective

The FYFD is an application for Android mobile devices. The application will be used to find the appropriate device for the user based on the device specifications selected. The system interfaces with other systems, User email systems, and the Android operating system (OS).

2.2 Product Functions

When the user wants to select a laptop he/she can determine the device type and get the benefit by fill the device form – include price appropriate to its budget and his/her requirements of the device - Which makes the results the best possible. There is two option to select device the first one is a laptop or mobile device.
If the user wants to save one or more results he/she should create an account and save the results.
The user can search for devices without needing to log in, The user can add his favorite device to the favorites icon, and The application should also be free and downloadable through a mobile phone application store or similar and All items will be provided from websites through API, this software requires an internet connection to be able to connect to websites to get all the needed data for this application budget and need.
All this information is stored in the system database.

2.3 User Characteristics

Our application targets anyone in the process of buying a laptop or mobile device, and fall in the targeted age range, (14-60) years old, therefore we have 2 types of users: end-users and administrators, each type will interact with the application differently.

End users, we are interacting with 2 types of end-users.
Advanced level users whose informations about technology are advanced.
The basic level user whose informations about technology are basic will use and navigate the application from their mobile phones to find the best device and there will be a small description of the complicated information like RAM, CPU...etc easily and understandably for the Basic level user.
Admin, they are managing the whole system, prevent it from crashing down, they are also responsible to add the devices and their description in the app.

3. System Analysis

3.1 Functional Requirements

1.Register:

1.1Users should be able to create an account.
1.2The app should be able to add the user's mail.

2.Login:

2.1Users and admin should be able to log in.
2.2There are 3 attempts to log in. If the user makes a mistake in the password 3 times, he/she must go to the "Forget my password".

3. Logout:

3.1Users and admin should be able to logout.

4.Find an item:

Given that a user is logged in to the mobile application:

4.1On the first page, the user will choose to register or log in and will be moved to the next page.

4.2At the top of the next page, there's a search bar for a specific device, and two options: a laptop or a mobile phone, which will be transferred to one of the search options.

4.3The user should be able to search for an appropriate device, according to several search options.

4.4The search options are the Price and specification model.

4.5A user should be able to select multiple search options in one search.

5.Favorite Icon:

5.1The user shall allow placing the appropriate devices for him in this area so that he can refer to it.

5.2This function will automatically take the product price.

5.3And then display it below the product image.

6.Advanced Search:

6.1User should be able to select among various choices to filter the result and fox on what he/she needs.

6.2Filter choices are device type, device capacity, device color…etc.

6.Add devices:

6.1The admin shall add devices and it's characteristics.

6.2The admin shall delete devices that are no longer on market.

6.3 The admin shall update the equipment information price and others if needed.

6.4 The admin shall manage the whole system.

3.2 Non-Functional Requirements

1.Performance Requirements:

The System should update data in the database when is new data arrive within 40 seconds and should be able to respond to more than one hundred users simultaneously.

2.Safety Requirements:

The system must automatically log out the user or admin.

3.Security Requirements:

3.1The system should not show customer privacy and making sure the customer's private information is secure.

3.2The system must ask the user for a password that contains 6 numbers, an upper case letter, and a lowercase letter.

4.Reliability Requirements:

This attribute specifies how likely the system or its element would run without a failure for a given time under predefined conditions.

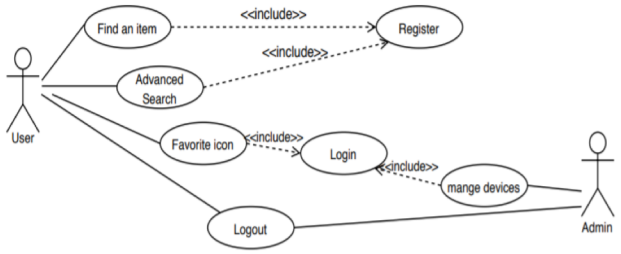
5.Data integrity Requirements:

maintaining and assuring data accuracy and consistency.

6.Responsive interface Requirements:

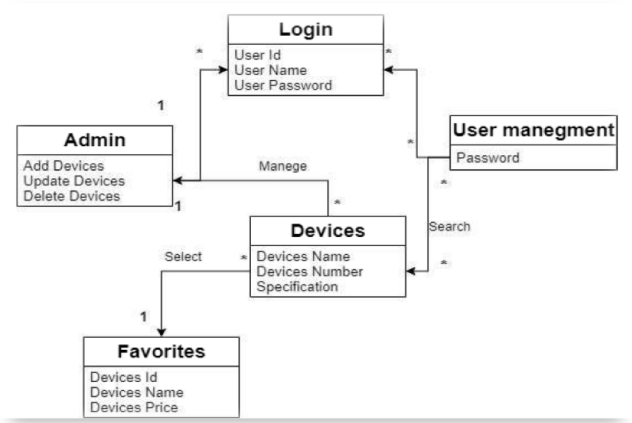
Widescreen, portrait, multi-screen, and tablets will be supported.

3.3 Use Case Diagram



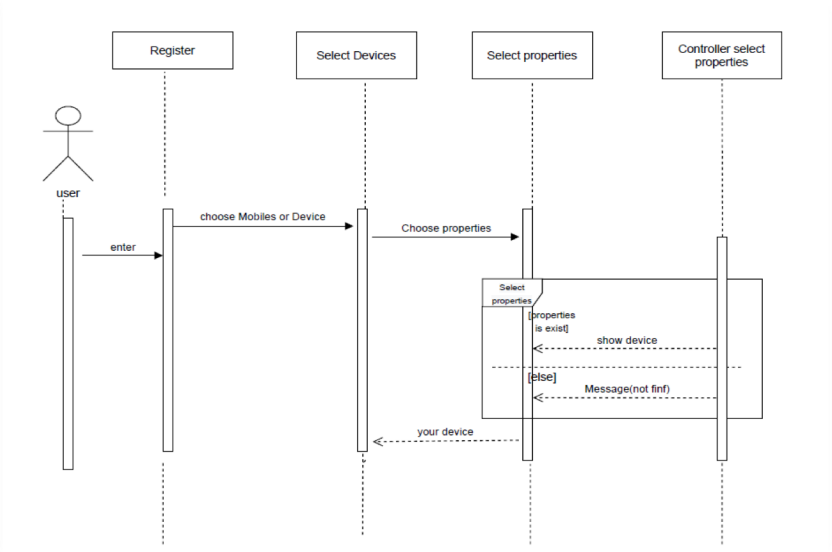
4. System Design

4.1 Class Diagram

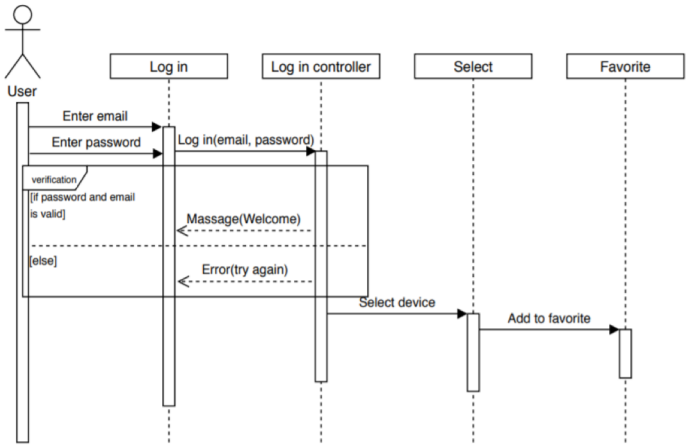


4.2 Sequence Diagram

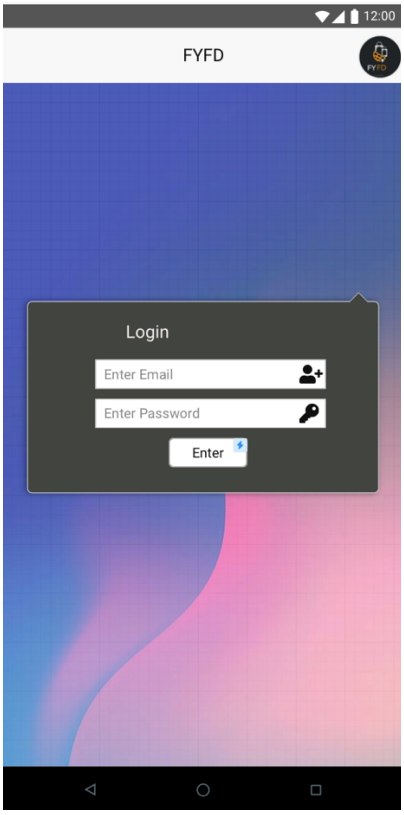
4.2.1. Find an item



4.2.2. Favorite Icon

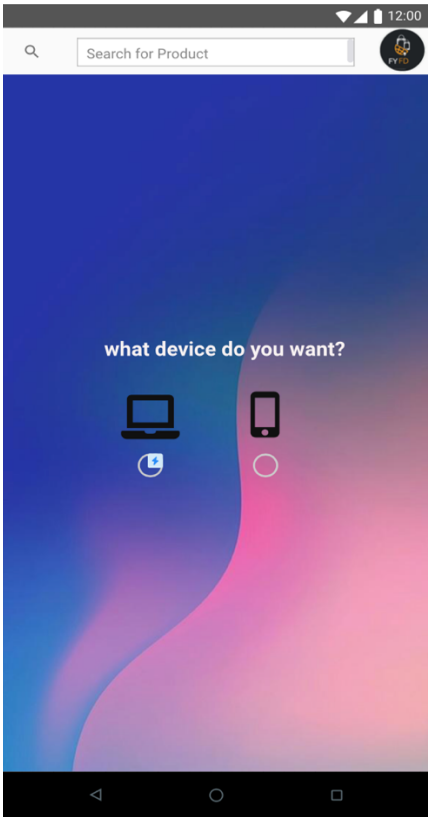


4.3 User Interface Design



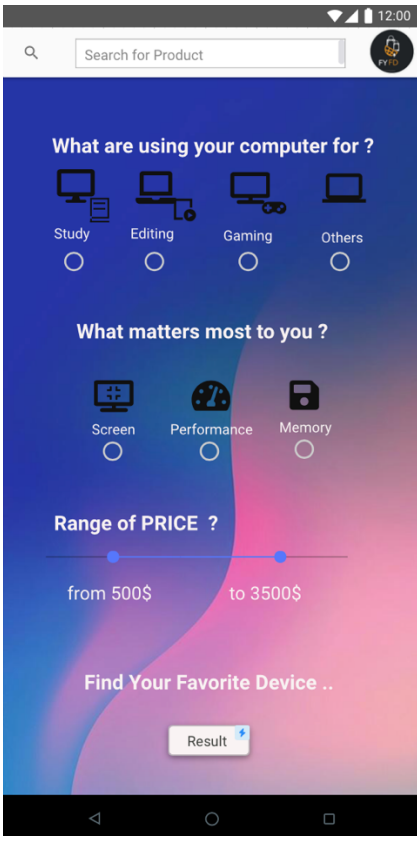
Figuer.1

Home page with login



Figuer.2

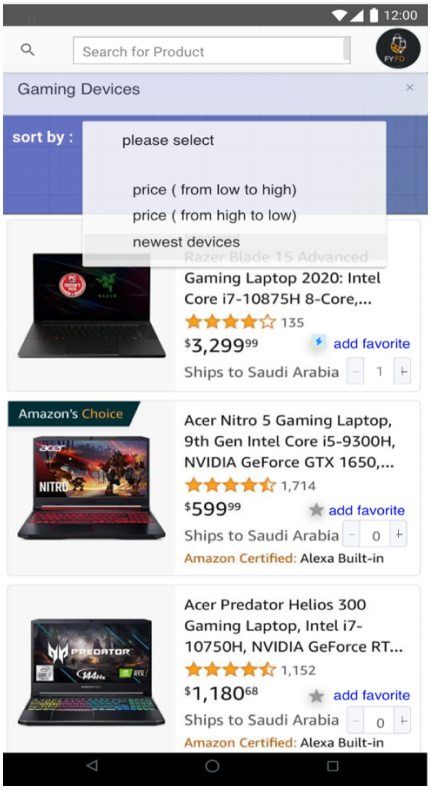
Follows the login, a search bar and a question



Figuer.3

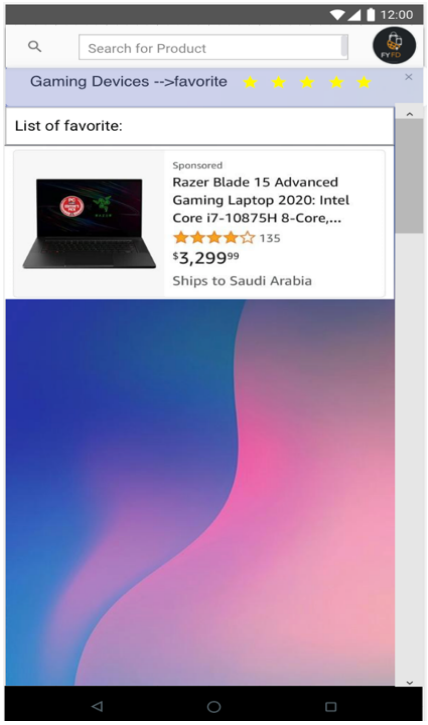
Follows previous page, leads you to this page with more questions

< FYFD >



Figuer.4

Result of the result button, a Sort by and Adding a device to the favorite list options



Figuer.5

The favorite list option

5. Implementation and Testing

Implementation:

1. Login:

```
                Border label_border=BorderFactory.createMatteBorder(1, 1, 1, 1, Color.black);
                jLabel_login.setBorder(label_border);

                 Border label_icons_border=BorderFactory.createMatteBorder(1, 1, 1, 1, new Color (153,153,153));
                jLabel_email.setBorder(label_icons_border);
                 jLabel_Password.setBorder(label_icons_border);


    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jButton1 = new javax.swing.JButton();
        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jButton_Enter = new javax.swing.JButton();
        jPanel3 = new javax.swing.JPanel();
        jLabel_email = new javax.swing.JLabel();
        jTextField_Emile = new javax.swing.JTextField();
        jLabel_Password = new javax.swing.JLabel();
        jPasswordField = new javax.swing.JPasswordField();
```

```
package login;

import java.awt.Color;
import javax.swing.BorderFactory;
import javax.swing.border.Border;

/**
 *
 * @author hind
 */
public class loginform extends javax.swing.JFrame {

    /**
     * Creates new form loginform
     */
    public loginform() {
        initComponents();

        this.setLocationRelativeTo(null);

        Border jPanel_title_border=BorderFactory.createMatteBorder(0, 1, 1, 1, Color.black);


        jPanel_title.setBorder(jPanel_title_border);
                Border label_border=BorderFactory.createMatteBorder(1, 1, 1, 1, Color.black);
                jLabel_login.setBorder(label_border);
```

Source Design History

```
        jButton_Enter.setForeground(new java.awt.Color(255, 255, 255));
        jButton_Enter.setText("Enter");
        jButton_Enter.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton_EnterActionPerformed(evt);
            }
        });

        jPanel3.setBackground(new java.awt.Color(102, 102, 102));

        jLabel_email.setText("*");

        jTextField_Emile.setForeground(new java.awt.Color(153, 153, 153));
        jTextField_Emile.setText("Enter Email");
        jTextField_Emile.addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusGained(java.awt.event.FocusEvent evt) {
                jTextField_EmileFocusGained(evt);
            }
            public void focusLost(java.awt.event.FocusEvent evt) {
                jTextField_EmileFocusLost(evt);
            }
        });
        jTextField_Emile.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jTextField_EmileActionPerformed(evt);
            }
        });

        jLabel_Password.setText("*");
```

```
    private void initComponents() {

        jButton1 = new javax.swing.JButton();
        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jButton_Enter = new javax.swing.JButton();
        jPanel3 = new javax.swing.JPanel();
        jLabel_email = new javax.swing.JLabel();
        jTextField_Emile = new javax.swing.JTextField();
        jLabel_Password = new javax.swing.JLabel();
        jPasswordField = new javax.swing.JPasswordField();
        jPanel_title = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        jLabel_login = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        jLabel_titlel = new javax.swing.JLabel();

        jButton1.setText("jButton1");

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);

        jPanel1.setBackground(new java.awt.Color(204, 204, 255));

        jButton_Enter.setBackground(new java.awt.Color(51, 51, 51));
        jButton_Enter.setFont(new java.awt.Font("Microsoft YaHei", 0, 24)); // NOI18N
        jButton_Enter.setForeground(new java.awt.Color(255, 255, 255));
        jButton_Enter.setText("Enter");
        jButton_Enter.addActionListener(new java.awt.event.ActionListener() {
```

< FYFD >

```
        jPanel_titleLayout.setHorizontalGroup(
            jPanel_titleLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel_titleLayout.createSequentialGroup()
                .addGap(52, 52, 52)
                .addComponent(jLabel_login)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 24, Short.MAX_VALUE)
                .addComponent(jLabel2)
                .addGap(140, 140, 140))
        );
        jPanel_titleLayout.setVerticalGroup(
            jPanel_titleLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel_titleLayout.createSequentialGroup()
                .addContainerGap()
                .addGroup(jPanel_titleLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel_login)
                    .addComponent(jLabel2))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

        javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(
            jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jLabel_Password, javax.swing.GroupLayout.PREFERRED_SIZE, 20, javax.swing.GroupLayout.PREFERRED
                    .addComponent(jLabel_email, javax.swing.GroupLayout.PREFERRED_SIZE, 26, javax.swing.GroupLayout.PREFERRED_SIZ
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        jPasswordField.setForeground(new java.awt.Color(153, 153, 153));
        jPasswordField.setText("123456789");
        jPasswordField.addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusGained(java.awt.event.FocusEvent evt) {
                jPasswordFieldFocusGained(evt);
            }
            public void focusLost(java.awt.event.FocusEvent evt) {
                jPasswordFieldFocusLost(evt);
            }
        });
        jPasswordField.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jPasswordFieldActionPerformed(evt);
            }
        });

        jPanel_title.setBackground(new java.awt.Color(94, 89, 89));

        jLabel2.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N

        jLabel_login.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
        jLabel_login.setForeground(new java.awt.Color(255, 255, 255));
        jLabel_login.setText("Login");

        javax.swing.GroupLayout jPanel_titleLayout = new javax.swing.GroupLayout(jPanel_title);
        jPanel_title.setLayout(jPanel_titleLayout);
        jPanel_titleLayout.setHorizontalGroup(
            jPanel_titleLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel_titleLayout.createSequentialGroup()
```

```
        javax.swing.GroupLayout jPanelLayout = new javax.swing.GroupLayout(jPanel);
        jPanel.setLayout(jPanelLayout);
        jPanelLayout.setHorizontalGroup(
            jPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanelLayout.createSequentialGroup()
                .addGroup(jPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanelLayout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jLabel1))
                    .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.Group
                    .addGroup(jPanelLayout.createSequentialGroup()
                        .addGap(143, 143, 143)
                        .addComponent(jButton_Enter, javax.swing.GroupLayout.PREFERRED_SIZE, 154, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        jPanelLayout.setVerticalGroup(
            jPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanelLayout.createSequentialGroup()
                .addGap(239, 239, 239)
                .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayo
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel1)
                .addGap(67, 67, 67)
                .addComponent(jButton_Enter)
                .addContainerGap(173, Short.MAX_VALUE))
        );

        jPanel2.setBackground(new java.awt.Color(255, 255, 255));
```

```
Design  History
                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel3Layout.createSequentialGroup()
                        .addComponent(jTextField_Emile)
                        .addGap(56, 56, 56))
                    .addGroup(jPanel3Layout.createSequentialGroup()
                        .addComponent(jPasswordField, javax.swing.GroupLayout.PREFERRED_SIZE, 333, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addContainerGap(56, Short.MAX_VALUE))))
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel_title, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.Grou
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        jPanel3Layout.setVerticalGroup(
            jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel_title, javax.swing.GroupLayout.PREFERRED_SIZE, 46, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jTextField_Emile, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.sw
                    .addComponent(jLabel_email))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 43, Short.MAX_VALUE)
                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel_Password)
                    .addComponent(jPasswordField, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swin
                .addContainerGap())
        );

        javax.swing.GroupLayout jPanelLayout = new javax.swing.GroupLayout(jPanel);
```

```
        jLabel_title1.setText("                FYFD");

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(94, 94, 94)
                .addComponent(jLabel_title1, javax.swing.GroupLayout.PREFERRED_SIZE, 253, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(138, Short.MAX_VALUE))
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(24, 24, 24)
                .addComponent(jLabel_title1, javax.swing.GroupLayout.PREFERRED_SIZE, 37, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(34, Short.MAX_VALUE))
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

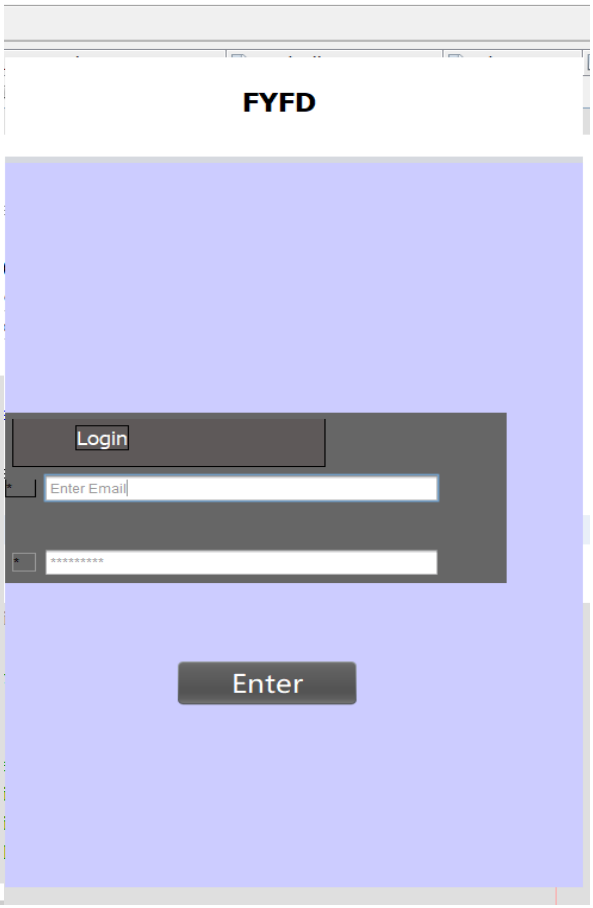
```
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayo
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayo
        );

        pack();
    }// </editor-fold>

    private void jTextField_EmileActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jTextField_EmileFocusGained(java.awt.event.FocusEvent evt) {
        // TODO add your handling code here:
         if(jTextField_Emile.getText().trim().toLowerCase().equals("gg@gmail.com")){
          jTextField_Emile.setText("");
          jTextField_Emile.setForeground(Color.black);
         }
         Border jlable_icon=BorderFactory.createMatteBorder(0, 1, 1, 1, Color.black);


        jLabel_email.setBorder(jlable_icon);
    }

    private void jTextField_EmileFocusLost(java.awt.event.FocusEvent evt) {
```

```
        jLabel_email.setBorder(jlable_icon);
    }

    private void jTextField_EmileFocusLost(java.awt.event.FocusEvent evt) {
        // TODO add your handling code here:
        //if text filde is equale gg@gmail.com or empty
        //we will set Emile in filde
        if(jTextField_Emile.getText().trim().equals("")||
            jTextField_Emile.getText().trim().toLowerCase().equals("gg@gmail.com")){
          jTextField_Emile.setText("gg@gmail.com");
          jTextField_Emile.setForeground(new Color(153,153,153));

         }
        Border label_icons_border=BorderFactory.createMatteBorder(1, 1, 1, 1, new Color (153,153,153));
                jLabel_email.setBorder(label_icons_border);


    }

    private void jPasswordFieldFocusGained(java.awt.event.FocusEvent evt) {
        // TODO add your handling code here:
        String pass=String.valueOf(jPasswordField.getPassword());



        if(pass.trim().toLowerCase().equals("as")){
          jPasswordField.setText("");
          jPasswordField.setForeground(Color.black);
```

```
        // TODO add your handling code here:
        String pass=String.valueOf(jPasswordField.getPassword());
        if(pass.trim().equals("")||
            pass.trim().toLowerCase().equals("as")){
          jPasswordField.setText("as");
          jPasswordField.setForeground(new Color(153,153,153));

         }
        Border label_icons_border=BorderFactory.createMatteBorder(1, 1, 1, 1, new Color (153,153,153));
                jLabel_Password.setBorder(label_icons_border);


    }

    private void jPasswordFieldActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButton_EnterActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
```

< FYFD >

```
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new loginform().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton_Enter;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel_Password;
private javax.swing.JLabel jLabel_email;
private javax.swing.JLabel jLabel_login;
private javax.swing.JLabel jLabel_title1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel_title;
private javax.swing.JPasswordField jPasswordField;
private javax.swing.JTextField jTextField_Emile;
// End of variables declaration
```



What We Will Use To Build This log in ?

- Java Programming Language.

- NetBeans Editor.

What We Will Do In This login ?

- design the two forms using jpanels and borders

- create a jbutton (Enter) check if the email or password- are empty

- Assume a correct value for email, which is(gg@gmail.com)

- Assume a correct value for password , which is(as)

Java Swing Components We Will Use In This log in:

- JPanel - JTextfield - JButton - JLabel

2. Register:

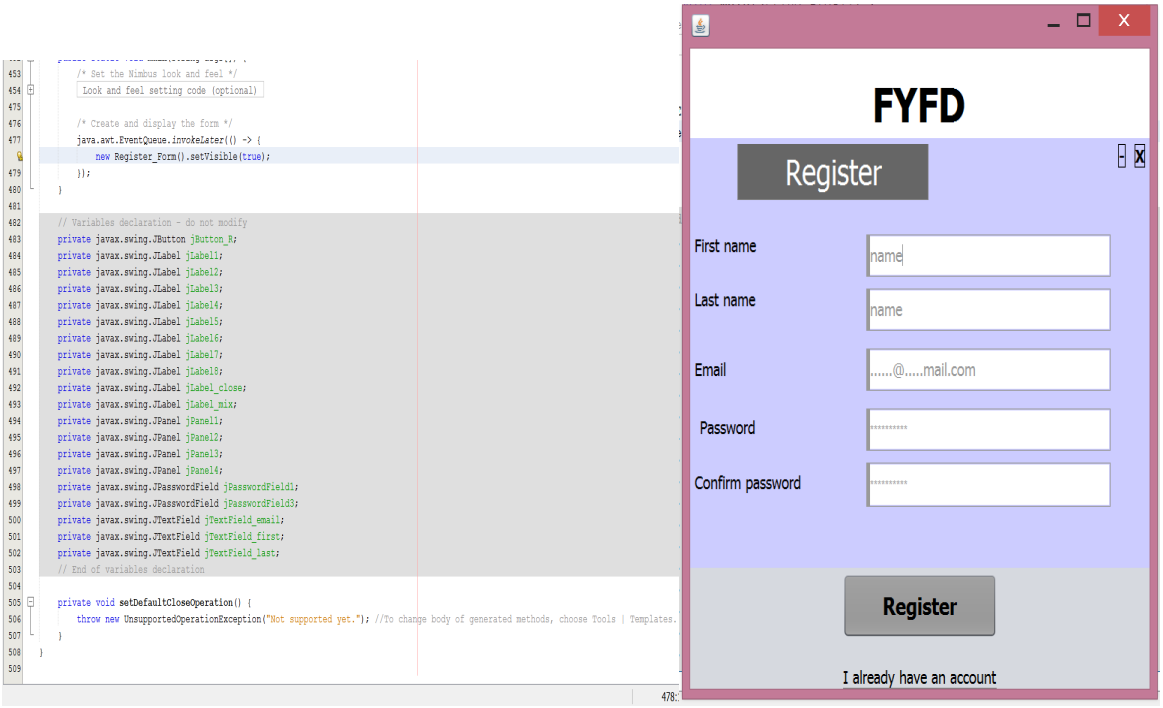
```
package eng;

import java.awt.Color;
import javax.swing.border.Border;
import javax.swing.BorderFactory;
import javax.swing.JFrame;



/**
 *
 * @author sy
 */
public class Register_Form extends javax.swing.JFrame {

    /**
     * Creates new form Register_Form
     */
    public Register_Form() {
        initComponents();

        this.setLocationRelativeTo(null);

        Border jPanel4_border=BorderFactory.createMatteBorder(0, 1, 1, 1, Color.gray);
        jPanel4.setBorder(jPanel4_border);

        Border label_border=BorderFactory.createMatteBorder(1, 1, 1, 1, Color.black);
               jLabel_mix.setBorder(label_border);
               jLabel_close.setBorder(label_border);

        Border jLabel8_border=BorderFactory.createMatteBorder(0, 0, 1, 0, Color.gray);
                    jLabel8.setBorder(jLabel8_border);


        Border field_border=BorderFactory.createMatteBorder(1, 5, 1, 1, new Color (153,153,153));
        jTextField_first.setBorder(field_border);
        jTextField_last.setBorder(field_border);
        jTextField_email.setBorder(field_border);
```

```
        jTextField_email.setBorder(field_border);
        jPasswordField1.setBorder(field_border);
        jPasswordField3.setBorder(field_border);



    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void jLabel_mixMouseEntered(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        Border label_border=BorderFactory.createMatteBorder(1, 1, 1, 1, Color.white);
                jLabel_mix.setBorder(label_border);
                jLabel_mix.setForeground(Color.white);
    }

    private void jLabel_mixMouseExited(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        Border label_border=BorderFactory.createMatteBorder(1, 1, 1, 1, Color.black);
                jLabel_mix.setBorder(label_border);
                jLabel_mix.setForeground(Color.black);


    }

    private void jLabel_closeMouseEntered(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        Border label_border=BorderFactory.createMatteBorder(1, 1, 1, 1, Color.white);
                jLabel_close.setBorder(label_border);
                jLabel_close.setForeground(Color.white);
    }

    private void jLabel_closeMouseExited(java.awt.event.MouseEvent evt) {
```

```
    private void jLabel_closeMouseExited(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        Border label_border=BorderFactory.createMatteBorder(1, 1, 1, 1, Color.black);
                jLabel_close.setBorder(label_border);
                jLabel_close.setForeground(Color.black);
    }

    private void jTextField_firstActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButton_RMouseEntered(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
            jButton_R.setBackground(new Color(102,102,102));
    }

    private void jButton_RMouseExited(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        jButton_R.setBackground(new Color(102,102,102));
    }

    private void jLabel_mixMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        this.setState(JFrame.ICONIFIED);
    }

    private void jLabel_closeMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        System.exit(0);
    }

    private void jButton_RActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String fname = jTextField_first.getText();
        String lname = jTextField_last.getText();
        String email = jTextField_email.getText();
        String pass1 = String.valueOf(jPasswordField1.getPassword());
        String pass2 = String.valueOf(jPasswordField3.getPassword());
```

```
        String pass2 = String.valueOf(jPasswordField3.getPassword());

    }

    private void jPasswordField1KeyTyped(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
    }

    private void jLabel8MouseEntered(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
            Border label_border=BorderFactory.createMatteBorder(0, 0, 1, 0, Color.gray);
            jLabel8.setBorder(label_border);

    }

    private void jLabel8MouseExited(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        Border jLabel8_border = BorderFactory.createMatteBorder(0, 0, 1, 0, Color.gray);
        jLabel8.setBorder(jLabel8_border);
    }

    private void jLabel8MouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        Register_Form rf = new Register_Form();
        rf.setVisible(true);
        rf.pack();
        rf.setDefaultCloseOperation();
        this.dispose();
    }



    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
         Look and feel setting code (optional)
```

< FYFD >



What We Will Use To Build This Register?

- Java Programming Language.

- NetBeans Editor.

What We Will Do In This Register?

-design it by using jpanels and borders

- create a jbutton (Register)

- create a JTextfield (first/last name,email,password,confirm password)

Java Swing Components We Will Use In This Register:

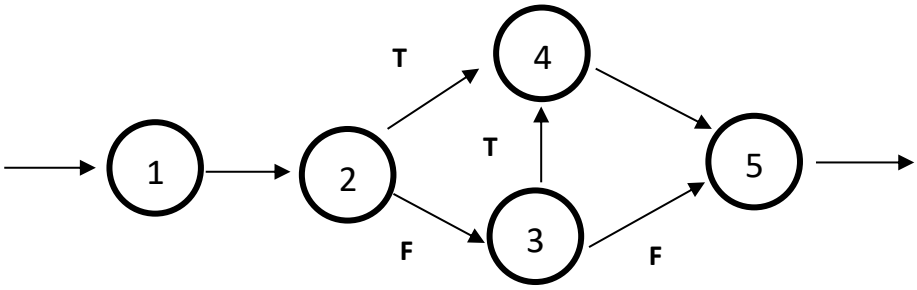
- JPanel - JTextfield - JButton - JLabel -Jpasswordfiled

Testing:

1.White box testing

```
private void jPasswordFieldFocusLost(java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
    String pass=String.valueOf(jPasswordField.getPassword()); //1
    if(pass.trim().equals("")||pass.trim().toLowerCase().equals("as")){ // 2 & 3
   jPasswordField.setText("as"); //4
   jPasswordField.setForeground(new Color(153,153,153)); //4
    }
    Border label_icons_border=BorderFactory.createMatteBorder(1, 1, 1, 1, new Color (153,153,153)); //5
        jLabel_Password.setBorder(label_icons_border); } //5
```

Step1: draw the CFG for the code fragment



Step 2: compute the cyclomatic complexity

Number of nodes and edges: N = 5, E = 6

E – N + 2 = 6 – 5 + 2 = 3

cyclomatic complexity = 3

Step 3: find the basic paths set

[1 – 2 – 4 – 5]

[1 – 2 – 3 – 5]

[1 – 2 – 3 – 4 – 5]

Step4: test case for each independent path

Path [1 – 2 – 3 – 4 – 5]:

Test case: First it will start at line No.1 then it will enter the IF statement, it will check statement No.2 (“pass.trim().equals(“”)), and it will be true; therefore we will go to line No.4 then line No.5

Path [1 – 2 – 3 – 4 – 5]:

Test case: First it will start at line No.1 then it will enter the IF statement, it will check statement No.2 “pass.trim().equals(“”)), and it will be false; therefore we will go to line No.3 (pass.trim().toLowerCase().equals("as")) it will be true then line No.4 at the end will go to line No.5.

Path [1 – 2 – 3 – 4 – 5]:

Test case: First it will start at line No.1 then it will enter the IF statement, it will check statement No.2 (“pass.trim().equals(“”)), and it will be false; therefore we will go to line No.3 (pass.trim().toLowerCase().equals("as")) it will be false then line No.5.

2.Black box testing

Equivalence partitions analysis

Functional Requirements
Login:

1. Users and admin should be able to log in.
2. There are 3 attempts to log in. If the user makes a mistake in the password 3 times, he/she must go to the "Forget my password".

Equivalence classes

Attempts < 3	Attempts >= 3
Valid (login)	Invalid (forget my password)

Test Cases:

1. Attempt to login 2 times – valid

2. Attempt to login 4 times – invalid

6. Futurework and Conclusion

Futurework:

1. Add the search feature for the nearest store that provides the appropriate device.
2. Providing buyer service, which is that the user assigns us the task of providing the device to him.
3. Performance and speed up.

Conclusion: Nowadays, everyone search for anything from his device and collect important information about what he wants to buy. Thus, we provide details for FYFD. It helps users to search about desired devices laptop or mobile device. We start from analyze the system by determine both functional and nonfunctional requirements and use case diagram. Then, we designed the system by provide class diagram, sequence diagram and user interface design. After that, we implement some functions and test them. Finally, we determine some features to improve our system by add them to it.

7. Team Members Contributions

Student	What tasks she performed
Hind Ahmed Al-Harbi	• Introduction • Purpos • Scope • Definitions, Acronyms, and Abbreviations • References • Major edit of product function • Functional Requirements • Non-Functional Requirements • Add Devices Function • Use Case Diagram • Sequence Diagram • Implementation for login
Mayar Weal Al-Shantaf	• Product Functions • Functional Requirements • Non-Functional Requirements • Compiling & formatting information in one document • Check the spelling and grammar • Fixed phase 01 again with phase 02 • End The Class Diagram • Implementation for Register
Njood Turki AL-Mukirsh	• User Characteristics • Functional Requirements • Non-Functional Requirements • User Interface Design • Testing • Futurework and Conclusion
Rand Ibrahim Al-Hossaini	• Product Perspective • Functional Requirements • Non-Functional Requirements • Use Case Diagram • Start The Class Diagram • Sequence Diagram • Testing • Futurework and Conclusion
Reema Nasser Al-Tammami	• Introduction • Purpose • Scope • Definitions, Acronyms, and Abbreviations • References • Functional Requirements • Non-Functional Requirements • Logo • User Interface Design • Testing