

World Cup

Fabio Oliveira (300275262)

CSIS 1275 - 001

Assignment 4

Gilbert Tsui

Date: April 12, 2018

Index

class WorldCupInfo	3
WorldCupInfo(String match, char stage, int tournament, String pic, double price, boolean avail)	3
class WorldCupList	5
public void createList()	5
class WorldCup extends JFrame implements ActionListener, ItemListener, ListSelectionListener	6
public static void main(String[] args)	7
public void createScreen()	7
public void actionPerformed(ActionEvent e)	12
public void itemStateChanged(ItemEvent e)	15
public boolean searchIndex(int index)	20
public void totalPrice()	20
public void createList(int year, char stage)	20
public void createAllMatchList(int year)	21
public class UpdateFrame extends JFrame implements ActionListener, ItemListener	22
UpdateFrame(WorldCupList a)	22
public void createScreen()	23
public void actionPerformed(ActionEvent e)	25
public void itemStateChanged(ItemEvent e)	26
public void insertPrice()	26

WorldCupInfo.java

```
class WorldCupInfo
{
    private String strWorldCupMatch;
    private char chWorldCupStage;
    private int intWorldCupTournament;
    private String strWorldCupLogo;
    private boolean boolAvail; // available for streaming
    private double doubPrice;

    WorldCupInfo(String match, char stage, int tournament, String pic, double price,
boolean avail )
    {
        strWorldCupMatch = match;
        chWorldCupStage = stage;
        intWorldCupTournament = tournament;
        strWorldCupLogo = pic;

        doubPrice = price;
        boolAvail = avail; // available for streaming
    }

    public void setStrWorldCupMatch(String match)
    {
        strWorldCupMatch = match;
    }
    public String getStrWorldCupMatch()
    {
        return strWorldCupMatch;
    }

    public void setChWorldCupStage(char stage)
    {
        chWorldCupStage = stage;
    }

    public char getChWorldCupStage()
    {
        return chWorldCupStage;
    }
}
```

```

public void setIntWorldCupTournament(int tournament)
{
    intWorldCupTournament = tournament;
}

public int getIntWorldCupTournament()
{
    return intWorldCupTournament;
}

public void setStrWorldCupLogo(String pic)
{
    strWorldCupLogo = pic;
}

public String getStrWorldCupLogo()
{
    return strWorldCupLogo;
}

public void setBoolAvail(boolean avail)
{
    boolAvail = avail;
}

public boolean getIsBoolAvail()
{
    return boolAvail;
}

public void setDoubPrice(double price)
{
    doubPrice = price;
}

public double getDoubPrice()
{
    return doubPrice;
}

}

```

```

class WorldCupList
{
    WorldCupInfo[] wcList;

    public void createList()
    {
        wcList = new WorldCupInfo[20];
        wcList[0] = new WorldCupInfo("Eng. v Bra.", 'q', 2002, "wc2002.jpg", 9.99, true);
        wcList[1] = new WorldCupInfo("Ger. v USA", 'q', 2002, "wc2002.jpg", 9.99, true);
        wcList[2] = new WorldCupInfo("Spa. v S. Kor.", 'q', 2002, "wc2002.jpg", 9.99, true);
        wcList[3] = new WorldCupInfo("Sen. v Tur.", 'q', 2002, "wc2002.jpg", 9.99, true);
        wcList[4] = new WorldCupInfo("Ger. v S. Kor.", 's', 2002, "wc2002.jpg", 9.99, true);
        wcList[5] = new WorldCupInfo("Bra. v Tur.", 's', 2002, "wc2002.jpg", 9.99, true);
        wcList[6] = new WorldCupInfo("Ger. v Bra.", 'f', 2002, "wc2002.jpg", 19.99, true);

        wcList[7] = new WorldCupInfo("Ger. v Ita.", 's', 2006, "wc2006.jpg", 9.99, true);
        wcList[8] = new WorldCupInfo("Por. v Fra.", 's', 2006, "wc2006.jpg", 9.99, true);
        wcList[9] = new WorldCupInfo("Ita. v Fra.", 'f', 2006, "wc2006.jpg", 19.99, true);

        wcList[10] = new WorldCupInfo("Uru. v Hol.", 's', 2010, "wc2010.jpg", 12.99, true);
        wcList[11] = new WorldCupInfo("Ger. v Spa.", 's', 2010, "wc2010.jpg", 12.99, true);
        wcList[12] = new WorldCupInfo("Hol. v Spa.", 'f', 2010, "wc2010.jpg", 19.99, true);

        wcList[13] = new WorldCupInfo("Fra. v Ger.", 'q', 2014, "wc2014.jpg", 9.99, true);
        wcList[14] = new WorldCupInfo("Bra. v Col.", 'q', 2014, "wc2014.jpg", 9.99, true);
        wcList[15] = new WorldCupInfo("Arg. v Belg.", 'q', 2014, "wc2014.jpg", 9.99, true);
        wcList[16] = new WorldCupInfo("Hol. v Cos.
Rica", 'q', 2014, "wc2014.jpg", 19.99, true);
        wcList[17] = new WorldCupInfo("Bra. v Ger.", 's', 2014, "wc2014.jpg", 19.99, false);
        wcList[18] = new WorldCupInfo("Hol. v Arg.", 's', 2014, "wc2014.jpg", 19.99, true);
        wcList[19] = new WorldCupInfo("Ger. v Arg.", 'f', 2014, "wc2014.jpg", 29.99, true);

    }
}

```

WorldCup.java

```
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.border.*;
import java.awt.*;
import java.awt.event.*;
import java.util.Vector;
import java.text.NumberFormat;

class WorldCup extends JFrame implements ActionListener, ItemListener,
ListSelectionListener
{
    JPanel mainPn, tourPn, imagePn, matchPn, matchPn1, matchPn2, selectionPn,
    selectionPn1, selectionPn2, passwordPn, passwordPn0, passwordPn1, passwordPn2,
    passwordPn3, totalPn;
    JComboBox cbTournament;
    JList lMatches, lYourSelection;
    Vector vtMatch, vtTour, vtYourSelection, vtMatchIndex, vtMatchPrice;
    ImageIcon[] picture = new ImageIcon[4];
    Image[] cupLogo = new Image[4];
    String[] worldCupLogo = {"wc2014.jpg", "wc2010.jpg", "wc2006.jpg", "wc2002.jpg"};
    String[] tourList = {"2014 Tournament", "2010 Tournament", "2006 Tournament", "2002
Tournament"};
    JScrollPane imageScr, scrMatch, scrSelect;
    JButton btnSelect, btnClear, btnClearAll, btnReset, btnEnter, btnSubmit;
    JButton[] btnPass = new JButton[10];
    String[] btnPassArray = {"A", "B", "C", "D", "E", "F", "G", "H", "I", "J"};
    JRadioButton rbAll, rbFinal, rbSemi, rbQuart;
    ButtonGroup rbGroup, nGroup;
    JPasswordField tfTotal;
    JPasswordField pwPassword, pwCredit;
    JLabel keypadLb, imageLb;

    String password = "";
    String keypassword = "F";
    double totalPrice = 0.00;

    Container con;

    WorldCupList matchList;
```

```

NumberFormat dollar = NumberFormat.getCurrencyInstance();

public static void main(String[] args)
{
    WorldCup frame = new WorldCup();
    frame.createScreen();
    frame.setTitle("FABIO DIAS World Cup Matches");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(1600, 900);
    frame.setVisible(true);
}

public void createScreen()
{
    mainPn = new JPanel();
    mainPn.setLayout(new GridLayout(3,2));

    //ComboBox
    tourPn = new JPanel();
    tourPn.setLayout(new BorderLayout());

    vtTour = new Vector();

    for(int i = 0; i < tourList.length; i++)
    {
        vtTour.add(tourList[i]);
        cbTournament = new JComboBox(vtTour);
        cbTournament.setFont(new Font(null,Font.BOLD + Font.ITALIC,14));
        tourPn.add(cbTournament,BorderLayout.NORTH);
        cbTournament.addItemListener(this);
    }
    tourPn.setBorder(new TitledBorder("Select Tournament"));
    mainPn.add(tourPn);

    //Image Panel
    imagePn = new JPanel();
    imagePn.setBackground(Color.YELLOW);

    imageLb = new JLabel("",SwingConstants.CENTER);
    imageLb.setOpaque(true);
    imageLb.setBackground(Color.yellow);
}

```

```

imageScr = new JScrollPane();
imageScr.add(imageLb);

for(int i = 0; i < worldCupLogo.length; i++)
{
    picture[i] = new ImageIcon(worldCupLogo[i]);
}

imageLb.setIcon(picture[0]);
imagePn.add(imageLb);
mainPn.add(imagePn);

    //Available Matches Panel

    //Radio Buttons:

rbGroup = new ButtonGroup();

matchPn = new JPanel();
matchPn.setLayout(new GridLayout(1, 3));

matchPn1 = new JPanel();
matchPn1.setLayout(new GridLayout(2,1));

matchPn2 = new JPanel();
matchPn2.setLayout(new GridLayout(3, 2));

rbAll = new JRadioButton("All Matches", true);
rbAll.addItemListener(this);
matchPn1.add(rbAll);
rbGroup.add(rbAll);

rbFinal = new JRadioButton("Final", false);
rbFinal.addItemListener(this);
matchPn1.add(rbFinal);
rbGroup.add(rbFinal);

rbSemi = new JRadioButton("Semi-Final", false);
rbSemi.addItemListener(this);
matchPn1.add(rbSemi);
rbGroup.add(rbSemi);

rbQuart = new JRadioButton("Quart.-Final");
rbQuart.addItemListener(this);

```



```

matchPn1.add(rbQuart);
rbGroup.add(rbQuart);

    //ListBox Available Matches

vtMatch = new Vector();
vtMatchIndex = new Vector();
vtMatchPrice = new Vector();
matchList = new WorldCupList();
matchList.createList();

for(int i = 0; i < matchList.wcList.length; i++)
{
    vtMatch.add(matchList.wcList[i]);
}

lMatches = new JList(vtMatch);
lMatches.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
lMatches.setVisibleRowCount(10);

scrMatch = new JScrollPane(lMatches);
lMatches.addListSelectionListener(this);

btnSelect = new JButton("Select");
btnSelect.setFont(new Font(null,Font.BOLD + Font.ITALIC,18));
btnSelect.addActionListener(this);
matchPn2.add(btnSelect);

matchPn.setBorder(new TitledBorder("Available Matches"));
matchPn.add(matchPn1);
matchPn.add(scrMatch);
matchPn.add(matchPn2);
mainPn.add(matchPn);

    //Your Selection
selectionPn = new JPanel();
selectionPn.setLayout(new GridLayout(1, 2));
selectionPn.setBorder(new TitledBorder("Your Selection"));

selectionPn1 = new JPanel();
selectionPn1.setLayout(new FlowLayout());
selectionPn2 = new JPanel();
selectionPn2.setLayout(new GridLayout(2, 1));

```

```

btnClear = new JButton("Clear");
btnClear.setFont(new Font(null,Font.BOLD + Font.ITALIC,18));
btnClear.addActionListener(this);

btnClearAll = new JButton("Clear All");
btnClearAll.setFont(new Font(null,Font.BOLD + Font.ITALIC,18));
btnClearAll.addActionListener(this);

vtYourSelection = new Vector();
lYourSelection = new JList(vtYourSelection);
lYourSelection.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
lYourSelection.setVisibleRowCount(10);
scrSelect = new JScrollPane(lYourSelection);

selectionPn2.add(btnClear);
selectionPn2.add(btnClearAll);
selectionPn.add(scrSelect);
selectionPn.add(selectionPn2);
mainPn.add(selectionPn);

    //password:

passwordPn = new JPanel();
passwordPn.setBorder(new TitledBorder("For Employee use ONLY"));
passwordPn.setLayout(new GridLayout(3, 1));
passwordPn.setBackground(Color.RED);

passwordPn0 = new JPanel();
passwordPn0.setLayout(new GridLayout(2, 1));
passwordPn0.setBackground(Color.red);
keypadLb = new JLabel("Use keypad to enter password for settings");
keypadLb.setFont(new Font("Time New Roman", Font.PLAIN, 11));
passwordPn0.add(keypadLb);

passwordPn1 = new JPanel();
passwordPn1.setLayout(new GridLayout(2, 5));

passwordPn2 = new JPanel();
passwordPn2.setLayout(new GridLayout(1, 2));

passwordPn3 = new JPanel();
passwordPn3.setLayout(new GridLayout(1, 2));

for(int i = 0; i < btnPass.length; i++)

```

```

{
    btnPass[i] = new JButton();
    passwordPn1.add(btnPass[i]);
    btnPass[i].addActionListener(this);
    btnPass[i].setText(btnPassArray[i]);
}

pwPassword = new JPasswordField();
pwPassword.setEditable(false);
pwPassword.setEchoChar('*');
passwordPn2.add(pwPassword);

btnReset = new JButton("Reset");
btnReset.addActionListener(this);
passwordPn3.add(btnReset);

btnEnter = new JButton("Enter");
btnEnter.addActionListener(this);
passwordPn3.add(btnEnter);

passwordPn.add(passwordPn0);
passwordPn2.add(passwordPn1);
passwordPn.add(passwordPn2);
passwordPn.add(passwordPn3);

mainPn.add(passwordPn);

//total
totalPn = new JPanel();
totalPn.setLayout(new GridLayout(3, 1));

tfTotal = new JTextField();
tfTotal.setText("Total: ");
tfTotal.setBackground(Color.cyan);
tfTotal.setEditable(false);
tfTotal.setFont(new Font("Time New Roman", Font.BOLD, 30));

//CreditCard

pwCredit = new JPasswordField();
pwCredit.setBorder(new TitledBorder("Enter CC#"));
pwCredit.setEchoChar('*');
totalPn.add(pwCredit);

```

```

btnSubmit = new JButton("Submit");
btnSubmit.setFont(new Font(null,Font.BOLD + Font.ITALIC,18));
btnSubmit.addActionListener(this);

totalPn.add(tfTotal);
totalPn.add(pwCredit);
totalPn.add(btnSubmit);

mainPn.add(totalPn);

    //container
con = getContentPane();
con.add(mainPn);

createAllMatchList(2014);
}

public void actionPerformed(ActionEvent e)
{
    int message;

    if(e.getSource() instanceof JButton)
    {
        if(e.getSource() == btnSelect)
        {
            if(lMatches.getSelectedIndex() != -1)
            {
                int i = (int)vtMatchIndex.elementAt(lMatches.getSelectedIndex());

                if(searchIndex(i) == false)
                {
                    vtMatchPrice.add(i);

vtYourSelection.add(matchList.wcList[i].getStrWorldCupMatch() + " " +
matchList.wcList[i].getChWorldCupStage() + " " +
matchList.wcList[i].getIntWorldCupTournament() + " $" +
matchList.wcList[i].getDoubPrice());
                }
                else
                {
                    JOptionPane.showMessageDialog(this, "Match Already
selected\nPlease re-enter another match","Duplicate",JOptionPane.ERROR_MESSAGE);
                }
                lYourSelection.setListData(vtYourSelection);
            }
        }
    }
}

```

```

        totalPrice();
    }
}
else if(e.getSource() == btnClear)
{
    if(lYourSelection.getSelectedIndex() != -1)
    {
        vtYourSelection.remove(vtYourSelection.elementAt(lYourSelection.getSelectedIndex()));
        vtMatchPrice.remove(vtMatchPrice.elementAt(lYourSelection.getSelectedIndex()));
        lYourSelection.setListData(vtYourSelection);
        totalPrice();
    }
}
else if(e.getSource() == btnClearAll)
{
    if(vtYourSelection.size() != 0)
    {
        message = JOptionPane.showConfirmDialog(this, "Are You SURE?", "WARNING", JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);
        if(message == JOptionPane.YES_OPTION)
        {
            vtYourSelection.removeAllElements();
            vtMatchPrice.removeAllElements();
            lYourSelection.setListData(vtYourSelection);
            totalPrice();
        }
    }
}
else if(e.getSource() == btnReset)
{
    password = "";
    pwPassword.setText(password);
}
else if(e.getSource() == btnEnter)
{
    if(password.equals(keypassword))
    {
        UpdateFrame updateFrame = new UpdateFrame(matchList);
        updateFrame.createScreen();
        updateFrame.setSize(1200, 300);
        updateFrame.setVisible(true);
    }
}

```

```

        rbGroup.clearSelection();
        vtMatch.removeAllElements();
        password = "";
        pwPassword.setText(password);
    }
    else
    {
        JOptionPane.showConfirmDialog(pwPassword, "WRONG Password, please
re-enter", "Not Allowed", JOptionPane.PLAIN_MESSAGE, JOptionPane.ERROR_MESSAGE);
        password = "";
        pwPassword.setText(password);
    }
}
else if(e.getSource() == btnSubmit)
{
    if(vtYourSelection.size() != 0)
    {
        message = JOptionPane.showConfirmDialog(this, "Are You
SURE?", "WARNING", JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);

        if(!pwCredit.getText().equals(""))
        {
            if(message == JOptionPane.YES_OPTION)
            {
                JOptionPane.showConfirmDialog(this, "Total: " +
dollar.format(totalPrice) + " has been charged to your card. Thank you!\nAll selected
Matches will be available for streaming for 1
week", "Confirmation", JOptionPane.PLAIN_MESSAGE, JOptionPane.INFORMATION_MESSAGE);
                vtYourSelection.removeAllElements();
                vtMatchPrice.removeAllElements();
                pwCredit.setText("");
                lYourSelection.setListData(vtYourSelection);
                totalPrice();
            }
        }
        else
        {
            JOptionPane.showMessageDialog(this, "Please Enter a VALID
Credit Card", "Not Allowed", JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

```

        else
        {
            JOptionPane.showMessageDialog(this, "No Match selected, Please
select a match", "Not Allowed", JOptionPane.ERROR_MESSAGE);
        }
    }

    else
    {
        password += e.getActionCommand();
        pwPassword.setText(password);
    }
}

}

public void itemStateChanged(ItemEvent e)
{
    if(e.getSource() instanceof JComboBox)
    {
        if(e.getSource() == cbTournament)
        {
            if(cbTournament.getSelectedIndex() == 0)
            {
                imageLb.setIcon(picture[0]);
                if(rbFinal.isSelected())
                {
                    createList(2014, 'f');
                }
                else if(rbSemi.isSelected())
                {
                    createList(2014, 's');
                }
                else if(rbQuart.isSelected())
                {
                    createList(2014, 'q');
                }
                else if(rbAll.isSelected())
                {
                    createAllMatchList(2014);
                }
            }
        }
    }
}

```

```

else if(cbTournament.getSelectedIndex() == 1)
{
    imageLb.setIcon(picture[1]);
    if(rbFinal.isSelected())
    {
        createList(2010,'f');
    }
    else if(rbSemi.isSelected())
    {
        createList(2010,'s');
    }
    else if(rbQuart.isSelected())
    {
        createList(2010,'q');
    }
    else if(rbAll.isSelected())
    {
        createAllMatchList(2010);
    }
}
else if(cbTournament.getSelectedIndex() == 2)
{
    imageLb.setIcon(picture[2]);
    if(rbFinal.isSelected())
    {
        createList(2006,'f');
    }
    else if(rbSemi.isSelected())
    {
        createList(2006,'s');
    }
    else if(rbQuart.isSelected())
    {
        createList(2006,'q');
    }
    else if(rbAll.isSelected())
    {
        createAllMatchList(2006);
    }
}
else
{
    imageLb.setIcon(picture[3]);
    if(rbFinal.isSelected())

```



```

        {
            createList(2002,'f');
        }
        else if(rbSemi.isSelected())
        {
            createList(2002,'s');
        }
        else if(rbQuart.isSelected())
        {
            createList(2002,'q');
        }
        else if(rbAll.isSelected())
        {
            createAllMatchList(2002);
        }
    }
    lMatches.setListData(vtMatch);
}
}
else if(e.getSource() instanceof JRadioButton)
{

    if(rbFinal.isSelected())
    {
        if(cbTournament.getSelectedIndex() == 0)
        {
            createList(2014,'f');
        }
        else if(cbTournament.getSelectedIndex() == 1)
        {
            createList(2010,'f');
        }
        else if(cbTournament.getSelectedIndex() == 2)
        {
            createList(2006,'f');
        }
        else if(cbTournament.getSelectedIndex() == 3)
        {
            createList(2002,'f');
        }
    }
    else if(rbSemi.isSelected())
    {
        if(cbTournament.getSelectedIndex() == 0)

```

```

        {
            createList(2014,'s');
        }
        else if(cbTournament.getSelectedIndex() == 1)
        {
            createList(2010,'s');
        }
        else if(cbTournament.getSelectedIndex() == 2)
        {
            createList(2006,'s');
        }
        else if(cbTournament.getSelectedIndex() == 3)
        {
            createList(2002,'s');
        }
    }
    else if(rbQuart.isSelected())
    {

        if(cbTournament.getSelectedIndex() == 0)
        {
            createList(2014,'q');
        }
        else if(cbTournament.getSelectedIndex() == 1)
        {
            createList(2010,'q');
        }
        else if(cbTournament.getSelectedIndex() == 2)
        {
            createList(2006,'q');
        }
        else if(cbTournament.getSelectedIndex() == 3)
        {
            createList(2002,'q');
        }
    }
}
else
{
    if(cbTournament.getSelectedIndex() == 0)
    {
        createAllMatchList(2014);
    }
}

```

```

        else if(cbTournament.getSelectedIndex() == 1)
        {
            createAllMatchList(2010);
        }
        else if(cbTournament.getSelectedIndex() == 2)
        {
            createAllMatchList(2006);
        }
        else if(cbTournament.getSelectedIndex() == 3)
        {
            createAllMatchList(2002);
        }
    }
    lMatches.setListData(vtMatch);
}
}

public void valueChanged(ListSelectionEvent e)
{
    if(lMatches.getSelectedIndex() != -1)
    {
        int i = (int)vtMatchIndex.elementAt(lMatches.getSelectedIndex());

        if(matchList.wcList[i].getChWorldCupStage() == 'f')
        {
            lMatches.setSelectionBackground(Color.red);
            lMatches.setSelectionForeground(Color.white);
        }
        else if(matchList.wcList[i].getChWorldCupStage() == 's')
        {
            lMatches.setSelectionBackground(Color.blue);
            lMatches.setSelectionForeground(Color.white);
        }
        else //( 'q' )
        {
            lMatches.setSelectionBackground(Color.yellow);
            lMatches.setSelectionForeground(Color.black);
        }
    }
}
}

```

```

public boolean searchIndex(int index)
{
    boolean found = false;
    for(int i = 0; i < vtMatchPrice.size(); i++)
    {
        if((int)vtMatchPrice.elementAt(i) == index)
        {
            found = true;
            break;
        }
    }
    return found;
}

public void totalPrice()
{
    totalPrice = 0.00;
    for(int i = 0; i < vtMatchPrice.size(); i++)
    {
        totalPrice +=
matchList.wcList[(int)vtMatchPrice.elementAt(i)].getDoubPrice();
    }
    tfTotal.setText("Total: " + dollar.format(totalPrice));
}

public void createList(int year, char stage)
{
    vtMatch.removeAllElements();
    vtMatchIndex.removeAllElements();
    for(int i = 0; i < 20; i++)
    {
        if(matchList.wcList[i].getIntWorldCupTournament() == year)
        {
            if(matchList.wcList[i].getChWorldCupStage() == stage)
            {
                if(matchList.wcList[i].getIsBoolAvail() == true)
                {
                    vtMatchIndex.add(i);
                    vtMatch.add(matchList.wcList[i].getStrWorldCupMatch() + "
" + matchList.wcList[i].getChWorldCupStage() + " " +
matchList.wcList[i].getIntWorldCupTournament() + " $" +
matchList.wcList[i].getDoubPrice());
                }
            }
        }
    }
}

```

```

        }
    }
}

public void createAllMatchList(int year)
{
    vtMatch.removeAllElements();
    vtMatchIndex.removeAllElements();
    for(int i = 0; i < matchList.wcList.length; i++)
    {
        if(matchList.wcList[i].getIntWorldCupTournament() == year)
        {
            if(matchList.wcList[i].getIsBoolAvail() == true)
            {
                vtMatchIndex.add(i);
                vtMatch.add(matchList.wcList[i].getStrWorldCupMatch() + " " +
matchList.wcList[i].getChWorldCupStage() + " " +
matchList.wcList[i].getIntWorldCupTournament() + " $" +
matchList.wcList[i].getDoubPrice());
            }
        }
    }
}
}

```

UpdateFrame.java (Password = F)

```
import javax.swing.*;
import javax.swing.border.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.Font;
import java.util.Vector;
import java.text.DecimalFormat;

public class UpdateFrame extends JFrame implements ActionListener,
ItemListener
{

    JComboBox cbMatchList;
    JTextField tfPrice;
    JList jlInventory;
    JTextArea taInventory;
    String txInventory;
    JScrollPane scrInventory;
    JButton btnUpdated;
    JMenuBar mb;
    JMenu menuFr;
    JMenuItem fullList, clear;
    JCheckBox chbAvailable;
    Vector vtMatch;
    JPanel mainPn, matchPn, priceAvailPn, pricePn, availablePn, updatedPn,
inventoryPn;

    DecimalFormat decimal = new DecimalFormat("0.00");

    WorldCupList matchList;

    Container con;

    UpdateFrame(WorldCupList a)
    {
        matchList = a;
    }
}
```

```

public void createScreen()
{

    mb = new JMenuBar();
    menuFr = new JMenu("Match List");
    menuFr.setMnemonic('M');

    fullList = new JMenuItem("Full List");
    fullList.setMnemonic('F');
    fullList.addActionListener(this);

    clear = new JMenuItem("Clear");
    clear.setMnemonic('C');
    clear.addActionListener(this);

    menuFr.add(fullList);
    menuFr.add(clear);

    mb.add(menuFr);
    setJMenuBar(mb);

    //Main UpdateFrame Panel
    mainPn = new JPanel();
    mainPn.setLayout(new GridLayout(2,2));

    //ComboBox MatchList

    String[] listItem = new String[matchList.wcList.length];
    for(int i = 0; i < matchList.wcList.length; i++)
    {
        listItem[i] = matchList.wcList[i].getStrWorldCupMatch() + " " +
matchList.wcList[i].getChWorldCupStage() + " " +
matchList.wcList[i].getIntWorldCupTournament();
    }

    matchPn = new JPanel();
    matchPn.setLayout(new BorderLayout());
    cbMatchList = new JComboBox(listItem);
    cbMatchList.setFont(new Font(null,Font.BOLD + Font.ITALIC,14));
    cbMatchList.addItemListener(this);
    matchPn.add(cbMatchList, BorderLayout.NORTH);

    //CheckBox Available and TextField Price

```

```

//Price
priceAvailPn = new JPanel();
priceAvailPn.setLayout(new GridLayout(2, 1));

pricePn = new JPanel();
tfPrice = new JTextField();
tfPrice.setBackground(Color.yellow);
TitledBorder border = new TitledBorder("Match Price");
tfPrice.setBorder(border);

//Available
availablePn = new JPanel();
availablePn.setLayout(new GridLayout(1, 1));

chbAvailable = new JCheckBox("Available");
chbAvailable.addActionListener(this);

availablePn.add(chbAvailable);

priceAvailPn.add(tfPrice);
priceAvailPn.add(availablePn);

//Button Updated

updatedPn = new JPanel();
updatedPn.setLayout(new BorderLayout());
btnUpdated = new JButton("Update Current Match");
btnUpdated.setFont(new Font(null,Font.BOLD + Font.ITALIC,18));
btnUpdated.addActionListener(this);
updatedPn.add(btnUpdated);

//TextArea Inventory

inventoryPn = new JPanel();
inventoryPn.setLayout(new BorderLayout());
TitledBorder title = new TitledBorder("Inventory");
title.setTitleFont(new Font(null,Font.BOLD + Font.ITALIC,20));
title.setTitlePosition(TitledBorder.CENTER);
inventoryPn.setBorder(title);
taInventory = new JTextArea();
taInventory.setFont(new Font("Helvetica",Font.BOLD + Font.ITALIC,14));
taInventory.setBackground(Color.ORANGE);
scrInventory = new JScrollPane(taInventory);
inventoryPn.add(scrInventory);

```



```

//All Panels in one
mainPn.add(matchPn);
mainPn.add(priceAvailPn);
mainPn.add(updatedPn);
mainPn.add(inventoryPn);

con = getContentPane();
con.add(mainPn);

insertPrice();

}

public void actionPerformed(ActionEvent e)
{
    if(e.getSource() instanceof JButton)
    {
        if(e.getSource() == btnUpdated)
        {
            int i = cbMatchList.getSelectedIndex();
            double newValue = Double.parseDouble(tfPrice.getText());

            tfPrice.setText(decimal.format(newValue));
            newValue = Double.parseDouble(tfPrice.getText());
            matchList.wcList[i].setDoubPrice(newValue);
            matchList.wcList[i].setBoolAvail(chbAvailable.isSelected());
        }
    }

    if(e.getSource() instanceof JMenuItem)
    {
        if (e.getSource() == fullList)
        {
            txInventory = "";
            for(int i = 0; i < matchList.wcList.length; i++)
            {
                txInventory += matchList.wcList[i].getStrWorldCupMatch() + "
" + matchList.wcList[i].getChWorldCupStage() + " " +
matchList.wcList[i].getIntWorldCupTournament() + " ==== $" +
matchList.wcList[i].getDoubPrice() + " " + matchList.wcList[i].getIsBoolAvail() +
"\n";
            }
            taInventory.setText(txInventory);
        }
    }
}

```

```

        }
        else
        {
            if (e.getSource() == clear)
                txInventory = "";
        }
        taInventory.setText(txInventory);
    }
}

public void itemStateChanged(ItemEvent e)
{
    if(e.getSource() == cbMatchList)
    {
        if(cbMatchList.getSelectedIndex() != -1)
        {
            insertPrice();
        }
    }
}

public void insertPrice()
{
    int i = cbMatchList.getSelectedIndex();

    String price = decimal.format(matchList.wcList[i].getDoubPrice());
    boolean getIsBoolAvail = matchList.wcList[i].getIsBoolAvail();

    tfPrice.setText(price);

    if(getIsBoolAvail == true)
    {
        chbAvailable.setSelected(true);
    }
    else
    {
        chbAvailable.setSelected(false);
    }
}
}

```