1. **ABSTRACT :**

**JAVA JFRAME --The javax.swing.JFrame class is a type of container which inherits the java.awt.Frame class. JFrame works like the main window where components like labels, buttons, textfields are added to create a GUI.**

**Unlike Frame, JFrame has the option to hide or close the window with the help of setDefaultCloseOperation(int) method.**

**The class JFrame is an extended version of java.awt.Frame that adds support for the JFC/Swing component architecture**.

**Following is the declaration for javax.swing.JFrame class −**

**public class JFrame**

**extends Frame**

**implements WindowConstants, Accessible, RootPaneContainer**

**Java ActionListener Interface--The Java ActionListener is notified whenever you click on the button or menu item. It is notified against ActionEvent. The ActionListener interface is found in java.awt.event**[**package**](https://www.javatpoint.com/package)**.**

**To write an Action Listener, follow the steps given below:**

1. **Declare an event handler class and specify that the class either implements an ActionListener interface or extends a class that implements an ActionListener interface. For example:**
2. **public class MyClass implements ActionListener {**
3. **Register an instance of the event handler class as a listener on one or more components. For example:**
4. **someComponent.addActionListener(instanceOfMyClass);**
5. **Include code that implements the methods in listener interface. For example:**
6. **public void actionPerformed(ActionEvent e) {**
7. **...//code that reacts to the action...**

**}**

**In our project there is a function of public void actionPerformed(ActionEvent e)**

**The following function will allows the the program to react when any buton is clicked by the user the it will either print any data or will allows to open any page related with it .**

1. **INTRODUCTION :**

# SWING --JAVA provides a rich set of libraries to create Graphical User Interface in a platform independent way.

**Swing API is a set of extensible GUI Components to ease the developer's life to create JAVA based Front End/GUI Applications. It is build on top of AWT API and acts as a replacement of AWT API, since it has almost every control corresponding to AWT controls. Swing component follows a Model-View-Controller architecture to fulfill the following criterias.**

* **A single API is to be sufficient to support multiple look and feel.**
* **API is to be model driven so that the highest level API is not required to have data.**
* **API is to use the Java Bean model so that Builder Tools and IDE can provide better services to the developers for use.**

**Classes And Methods in Swing in Java :**

|  |  |  |
| --- | --- | --- |
|  | ***Public Constructors :*** | |
|  | **public AbstractAction ();** |  |
|  | **public AbstractAction (String *name*);** |  |
|  | **public AbstractAction (String *name*, Icon *icon*);** |  |
|  | ***Event Registration Methods (by event name):*** | |
|  | **public void addPropertyChangeListener (java.beans.PropertyChangeListener *listener*);** | ***Implements:Action synchronized*** |
|  | **public void removePropertyChangeListener (java.beans.PropertyChangeListener *listener*);** | ***Implements:Action synchronized*** |
|  | ***Methods Implementing Action :*** | |
|  | **public void addPropertyChangeListener (java.beans.PropertyChangeListener *listener*);** | ***synchronized*** |
|  | **public Object getValue (String *key*);** |  |
|  | **public boolean isEnabled ();** |  |
|  | **public void putValue (String *key*, Object *newValue*);** | ***synchronized*** |
|  | **public void removePropertyChangeListener (java.beans.PropertyChangeListener *listener*);** | ***synchronized*** |
|  | **public void setEnabled (boolean *newValue*);** | ***synchronized*** |
|  | ***Methods Implementing ActionListener :*** | |
|  | **public abstract void actionPerformed (java.awt.event.ActionEvent *e*);** |  |
|  | ***Protected Methods Overriding Object :*** | |
|  | **protected Object clone () throws CloneNotSupportedException;** |  |

**public JCheckBox (String *text*, Icon *icon*, boolean *selected*);**

**public JColorChooser (javax.swing.colorchooser.ColorSelectionModel *model*);**

**Swing is a part of JFC, Java Foundation Classes. It is a collection of packages for creating full featured desktop applications. JFC consists of AWT, Swing, Accessibility, Java 2D, and Drag and Drop. Swing was released in 1997 with JDK 1.2. It is a mature toolkit.**

**The Java platform has Java2D library, which enables developers to create advanced 2D graphics and imaging.**

**There are basically two types of widget toolkits:**

* **Lightweight**
* **Heavyweight**

**A heavyweight toolkit uses OS's API to draw the widgets. For example Borland's VCL is a heavyweight toolkit. It depends on WIN32 API, the built-in Windows application programming interface. On Unix systems, we have a GTK+ toolkit, which is built on top of X11 library. Swing is a lightweight toolkit; it paints its own widgets. Similarly does the Qt4 toolkit.**

**JavaFX**

**In 2008, a new Java GUI toolkit was released. It was created in order to address new demands in graphical computing such as advanced animations and mulitouch support.**

**JavaFX is a software platform for developing and delivering rich internet applications (RIAs) that can run across a wide variety of devices. JavaFX is the next generation GUI toolkit for the Java platform. It is fully integrated with recent versions of Java SE Runtime Environment (JRE) and the Java Development Kit (JDK).**

## SWT library

**There is also another third-party GUI library for the Java programming language. It is called the Standard Widget Toolkit (SWT). SWT library was initially developed by the IBM corporation. Now it is an open source project maintained by the Eclipse community. SWT is an example of a heavyweight toolkit**

1. **Literature Survey OR Similar works :**

**Generally in order to give the proper layout and proper description of our project**

**We have gone to our seniours to get how to set our project and co-ordinate with them**

**Learned more about the swing package, classes, methods how to use them in our**

**Project to bring proper layout to our function**

**Also this same format have done in our practical lab first is by using applet got**

**Hoe to use strings ,drawfunction etc and also then GUI in swing by creating**

**The registration form.**

1. **Proposed System :**

**Our Bank management system has a login page where user with card no and pin can access one's account else he orshe can Register i.e can open a bank account by filling Registration form which includes personal detailsafter which a transaction page will open through which customer can go for deposition or withdrawal etc**

**The reason behind chosing Swings in place of applets is that Swings is platform independent where as Applets are platform dependent**

**Swings contains more components than Applets and also Swings contains lightweight components and Applets has heavy weighted components**

1. **Methodology :**

**Swings are platform independent in java where as Applets are platform dependent Swings contains more componets than applets**

**Following are the Swings Components used**

**1. JFrame**

**The class JFrame is an extended version of java.awt.Frame that adds support for the JFC/Swing component architecture.**

**Class Declaration**

**Following is the declaration for javax.swing.JFrame class −**

**public class JFrame**

**extends Frame**

**implements WindowConstants, Accessible**

**2.JButton**

**The JButton class is used to create a labeled button that has platform independent implementation. The application result in some action when the button is pushed. It inherits AbstractButton class.**

**Commonly used Constructors:**

**Constructor Description**

**JButton() It creates a button with no text and icon.**

**JButton(String s). It creates a button with the specified text.**

**JButton(Icon i) It creates a button with the specified icon object**

**3.JLabel**

**The object of JLabel class is a component for placing text in a container. It is used to display a single line of read only text. The text can be changed by an application but a user cannot edit it directly. It inherits JComponent class.**

**Constructor Description**

**JLabel() Creates a JLabel instance with no image and with**

**anemptystringforthetitle.**

**JLabel(String s). Creates a JLabel instance with the specified text.**

**JLabel(Icon i) Creates a JLabel instance with the specified image.**

**4. JComboBox**

**The object of Choice class is used to show popup menu of choices. Choice selected by user is shown on the top of a menu. It inherits JComponent class.**

**Constructor Description**

**JComboBox() Creates a JComboBox with a default data model.**

**JComboBox(Object[] items) Creates a JComboBox that contains the elements in the specified array.**

**JComboBox(Vector<?> items) Creates a JComboBox that contains the elements in the specified Vector.**

**5. JOptipnPane**

**The JOptionPane class is used to provide standard dialog boxes such as message dialog box, confirm dialog box and input dialog box. These dialog boxes are used to display information or get input from the user. The JOptionPane class inherits JComponent class.**

**constructor Description**

**JOptionPane() It is used to create a JOptionPane with a test**

**JOptionPane(Object message) It is used to create an instance of JOptionPane to**

**display a message.**

**6. ACTION LISTENER**

**The Java ActionListener is notified whenever you click on the button or menu item. It is notified against ActionEvent. The ActionListener interface is found in java.awt.event package. It has only one method: actionPerformed().**

**actionPerformed() method**

**The actionPerformed() method is invoked automatically whenever you click on the registered component.**

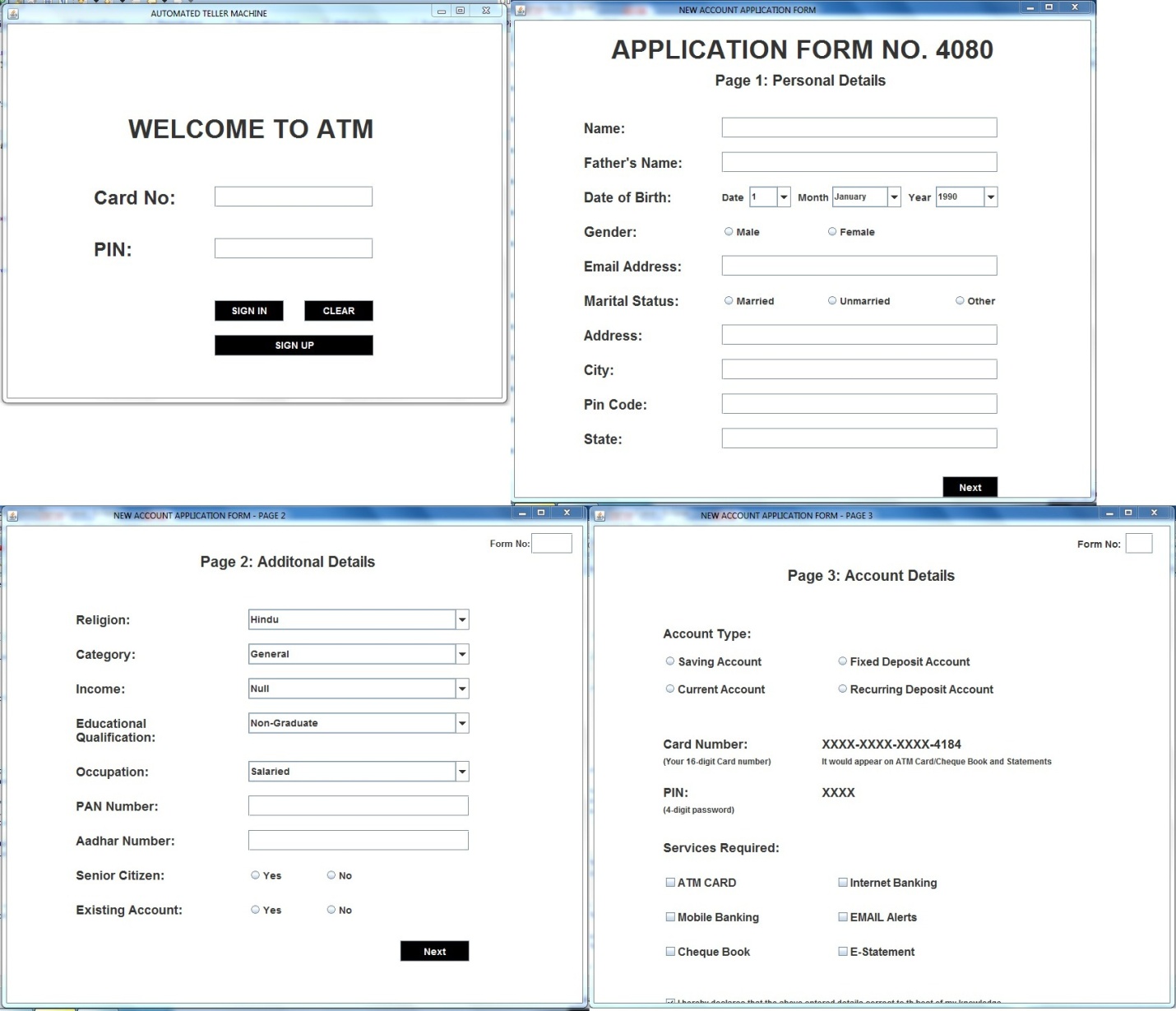
**public abstract void actionPerformed(ActionEvent e);**

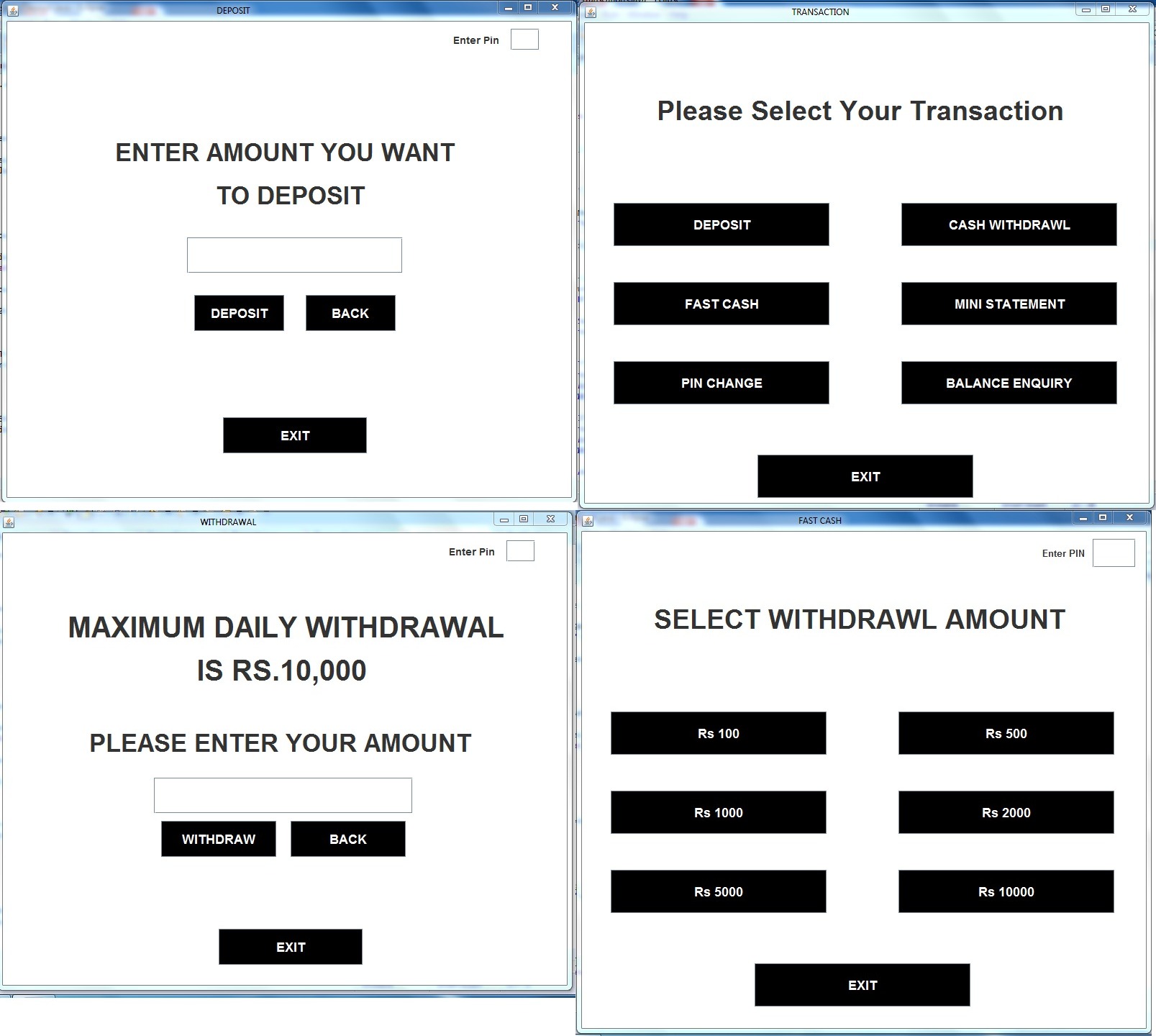
1. **Conclusion :**

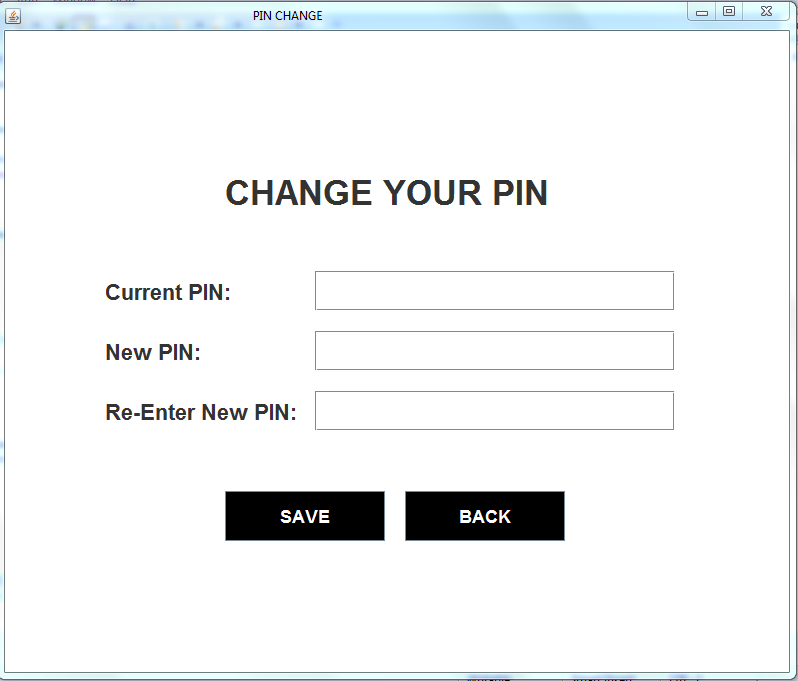
**Bank Management System from this project we have learned that how the java is such a vast language it is not only used to do the basic calculations ,display or to do other things it is used to relate with our basic life like how we used to use the websites , applications in our daily life as we have got an overview that how we think that they are used and designed are the two different parts of a coin like it is hard to implement and easy to use**

**Also we learned the different packages of swing in java its uses and different classes to use and how they are worked and their implementation like to create a frame first we declare a object of JFrame and pass parameter, to display ,to draw ,to change the color of foreground ,background, change the size ,font location of any thing used in java ,use of action listener in java that how it react on clicking mouse buttons or keyboard that’s how we have got an idea and knowlwdge.**

1. **Results or Snapshots :**

****

****

****

1. **References :**

[**http://www.oracle.com/technetwork/java/index.html**](http://www.oracle.com/technetwork/java/index.html)**).**

[**http://java.net**](http://java.net/)**.**

[**http://download.oracle.com/javase/7/docs/**](http://download.oracle.com/javase/7/docs/)**.**

[**http://docs.oracle.com/javase/tutorial/**](http://docs.oracle.com/javase/tutorial/).