PROJECT REPORT

बी एच ई एल

BHEL

GOVT. COLLEGE OF ENGINEERING KANNUR

GOVT. COLLEGE OF ENGG
CANNANORE
तमसो मा ज्योतिर्गमय

NAME	BRANCH	SEMESTER
ADARSH J	CSE	S7
BIBIN JAIMON	CSE	S7
MUFEEDA K	CSE	S7
NANDA V	CSE	S7

ACKNOWLEDGEMENT

For a short period, from 6th September 2017 to 15th September 2017, we did an Internship at the Bharat Heavy Electricals Limited (*BHEL*), Bangalore. BHEL owned and founded by the Government of India, is an engineering and manufacturing company based in New Delhi, India. Established in 1964, BHEL is India's largest power plant equipment manufacturer. This internship is a part of our 4-year B.Tech programme which we are pursuing at the Govt. College of Engineering, Kannur.

DECLARATION

We do hereby solemnly declare that the work presented in this Internship Report has been carried out by us in partial fulfilment of our B.Tech programme in Computer Science and Engineering field.

1. Adarsh J
2. Bibin Jaimon
3. Mufeeda K
4. Nanda V

Dept. of Computer Science and Engineering

Govt. College of Engineering, Kannur

CONTENTS

INTRODUCTION

- Backup is an application used to copy files or folders from different locations into a single destination folder.
- This project is developed by a group of computer science students – Adarsh J , Bibin Jaimon , Mufeeda K and Nanda V, studying at Government College Of Engineering Kannur.
- This application is used to reduce the human work and save time. It also provides a very friendly user interface.
- BAMN backup application can be used by type of people who want to copy different files or folders to a location. It is largely useful for people who want to backup large number of files frequently.
- Development environments used for this application are Eclipse IDE and Scene Builder.
- Platform used for this application is JavaFX.

APPLICATION

- BAMN Backup Application is implemented using 2 tools

  1. **Eclipse IDE**

     It is an integrated development environment which is primarily used for developing Java applications. Java FX library is used to provide graphical user interface.

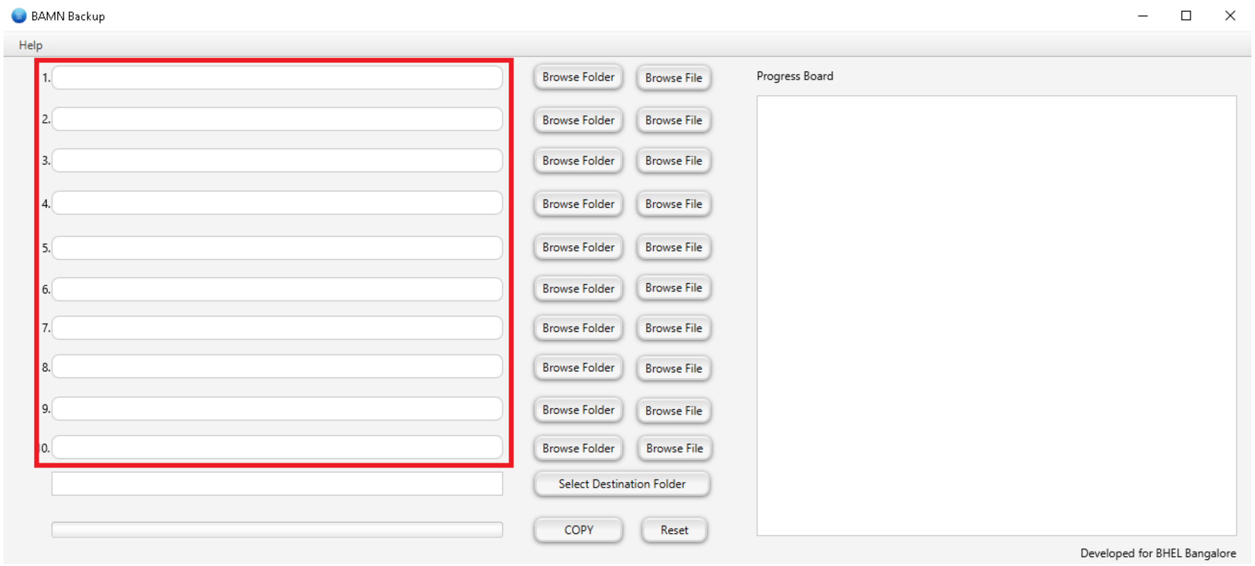
  2. **Scene Builder**

     It is a visual layout tool that lets users quickly design Java FX user interfaces without coding. The result is an FXML file that can be combined with a Java project.

- ***Features Of Application***

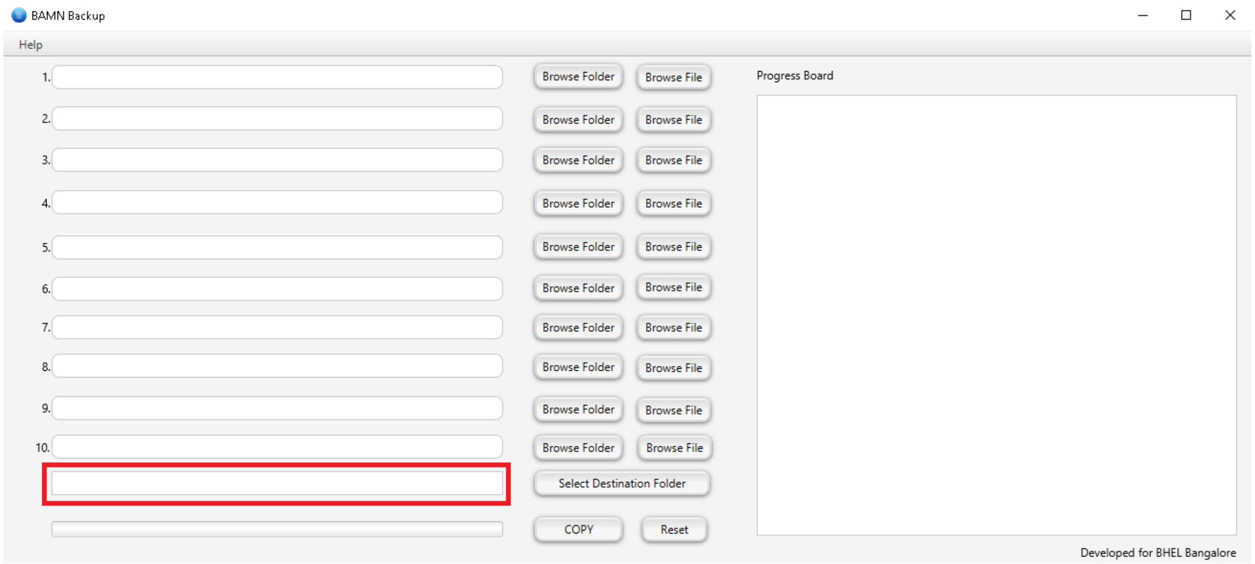
  1. **Fields**

     a. **List View**



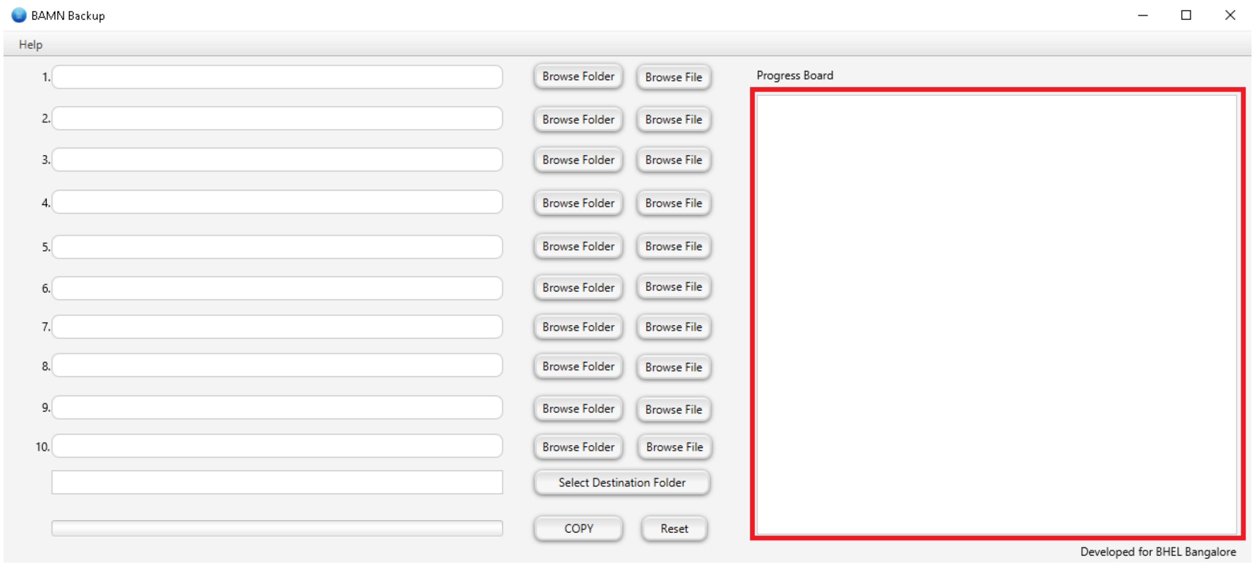
```
<ListView fx:id="listview1" layoutX="50.0" layoutY="36.0" prefHeight="25.0"
prefWidth="470.0" />
```

b. Destination View



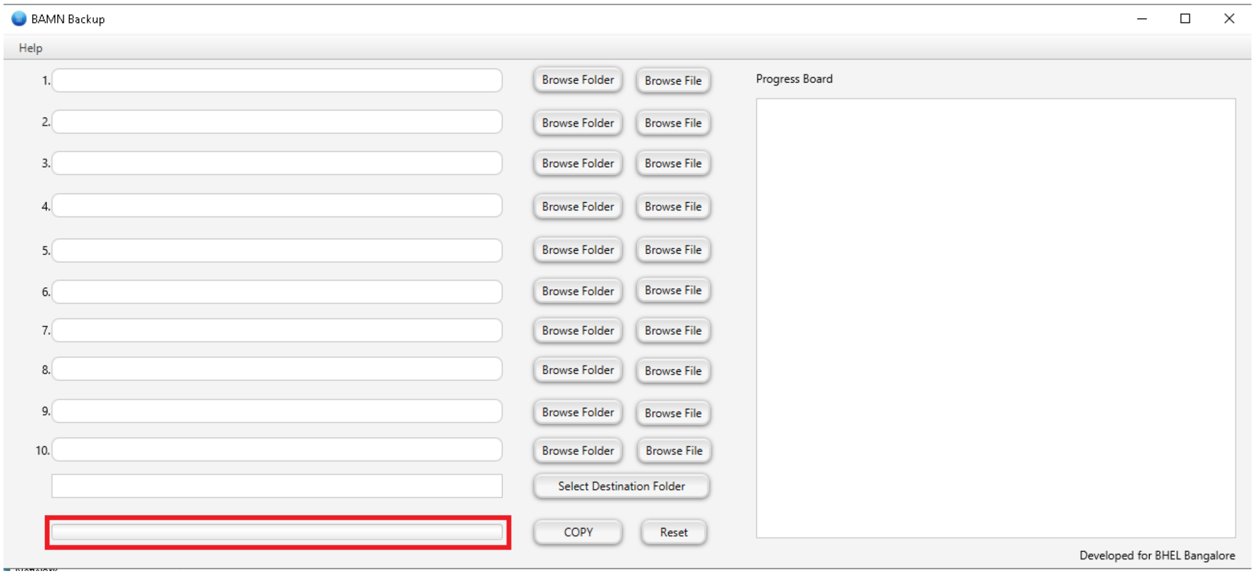
```
<ListView fx:id="destinationview" layoutX="50.0" layoutY="458.0" prefHeight="25.0"
prefWidth="470.0" />
```

c. Progress Board



```
<ListView fx:id="status" layoutX="784.0" layoutY="67.0" prefHeight="458.0"
prefWidth="500.0" />
```

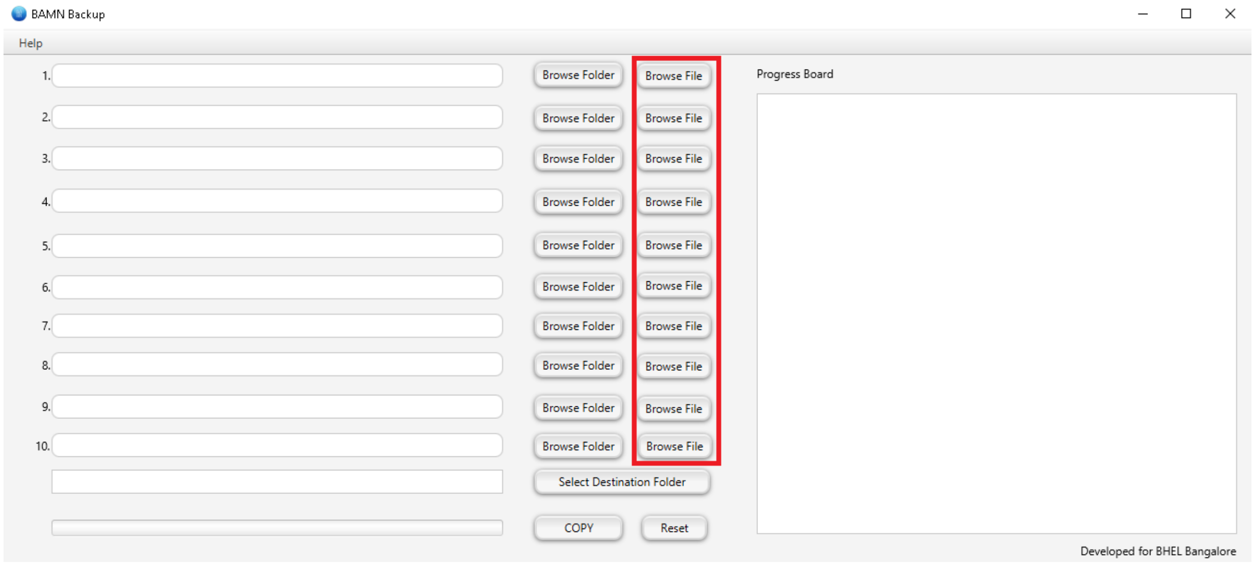
d. Progress Bar



```
<ProgressBar fx:id="pb" layoutX="50.0" layoutY="510.0" prefHeight="18.0"
prefWidth="470.0" progress="0.0" />
```

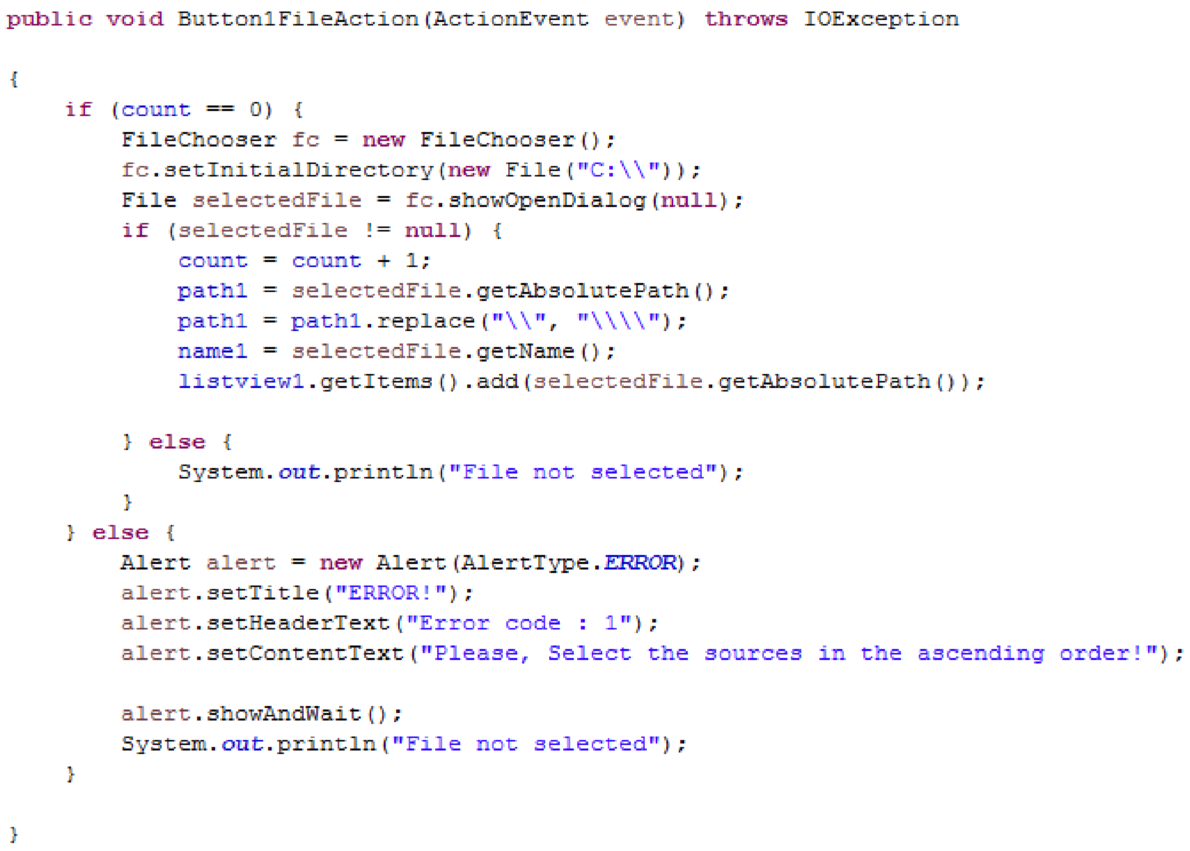
2. Buttons

a. Browse File Button



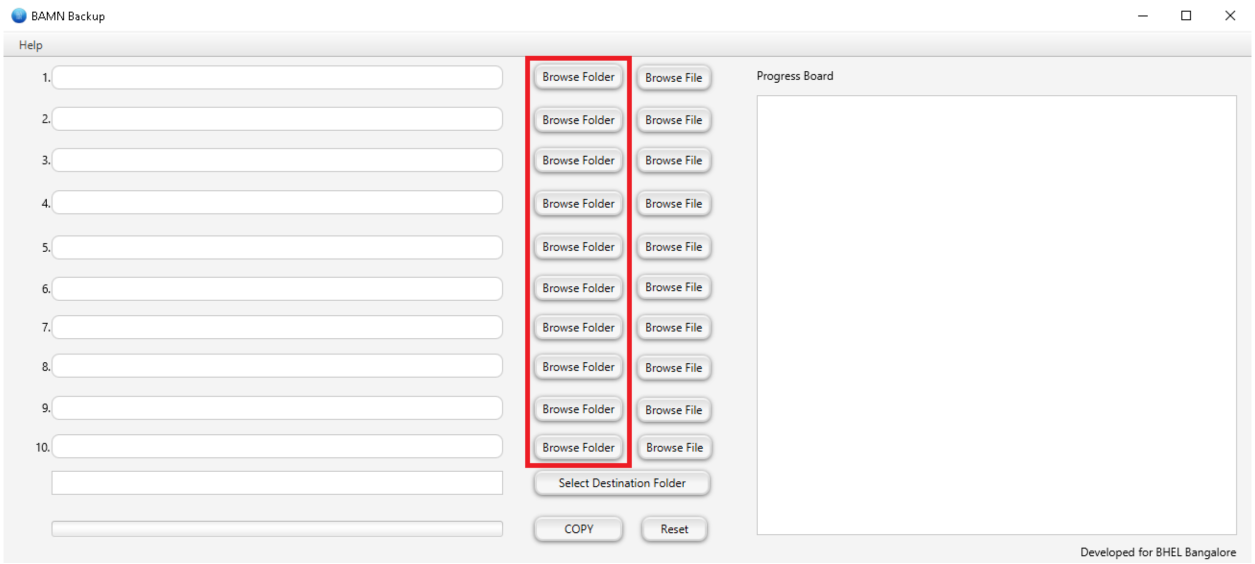
<Button fx:id="btn1file" layoutX="660.0" layoutY="36.0" mnemonicParsing="false" onAction="#Button1FileAction" text="Browse File" />

Action for browse file button



```
public void Button1FileAction(ActionEvent event) throws IOException

{
    if (count == 0) {
        FileChooser fc = new FileChooser();
        fc.setInitialDirectory(new File("C:\\"));
        File selectedFile = fc.showOpenDialog(null);
        if (selectedFile != null) {
            count = count + 1;
            path1 = selectedFile.getAbsolutePath();
            path1 = path1.replace("\\", "\\\\");
            name1 = selectedFile.getName();
            listview1.getItems().add(selectedFile.getAbsolutePath());

        } else {
            System.out.println("File not selected");
        }
    } else {
        Alert alert = new Alert(AlertType.ERROR);
        alert.setTitle("ERROR!");
        alert.setHeaderText("Error code : 1");
        alert.setContentText("Please, Select the sources in the ascending order!");

        alert.showAndWait();
        System.out.println("File not selected");
    }

}
```

b. Browse Folder Button

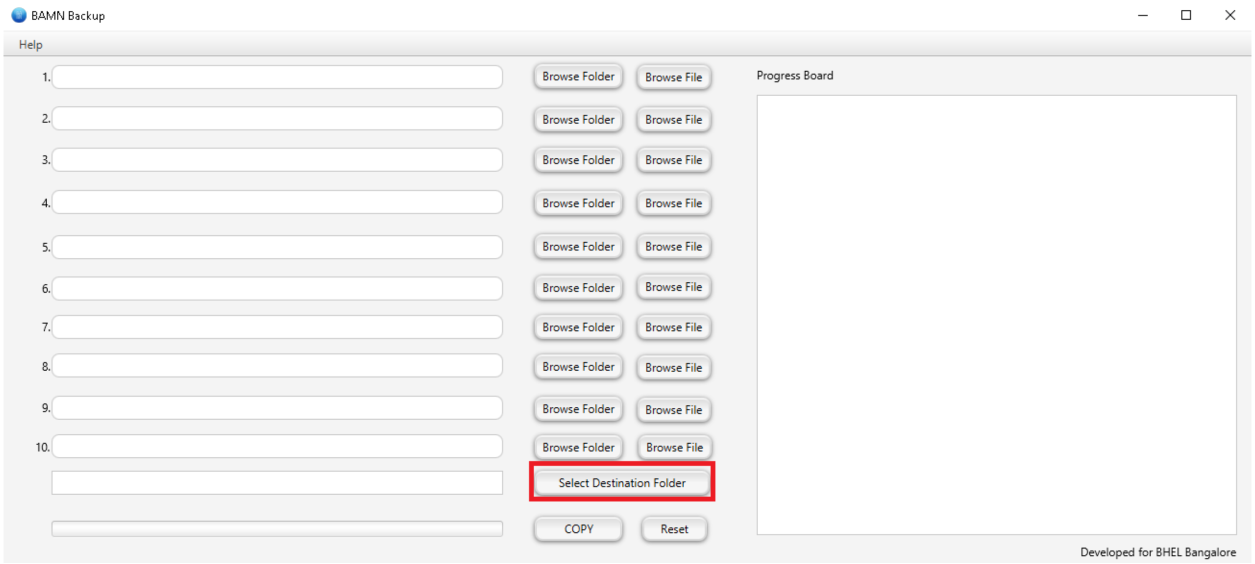


```
<Button fx:id="btn1fold" layoutX="553.0" layoutY="35.0" mnemonicParsing="false"
onAction="#Button1FolderAction" text="Browse Folder" />
```

Action for browse folder button

```
public void Button1FolderAction(ActionEvent event) throws IOException {
    if (count==0) {
        DirectoryChooser directoryChooser = new DirectoryChooser();
        File selectedDir = directoryChooser.showDialog(null);
        if (selectedDir != null) {
            count = count + 1;
            path1 = selectedDir.getAbsolutePath();
            path1 = path1.replace("\\", "\\\\");
            name1 = selectedDir.getName();
            listview1.getItems().add(selectedDir.getAbsolutePath());

        } else {
            System.out.println("File not selected");
        }
    } else {
        Alert alert = new Alert(AlertType.ERROR);
        alert.setTitle("ERROR!");
        alert.setHeaderText("Error code : 1");
        alert.setContentText("Please, Select the sources in the ascending order!");

        alert.showAndWait();
        System.out.println("File not Selected");
    }
}
```

c. Select Destination Button

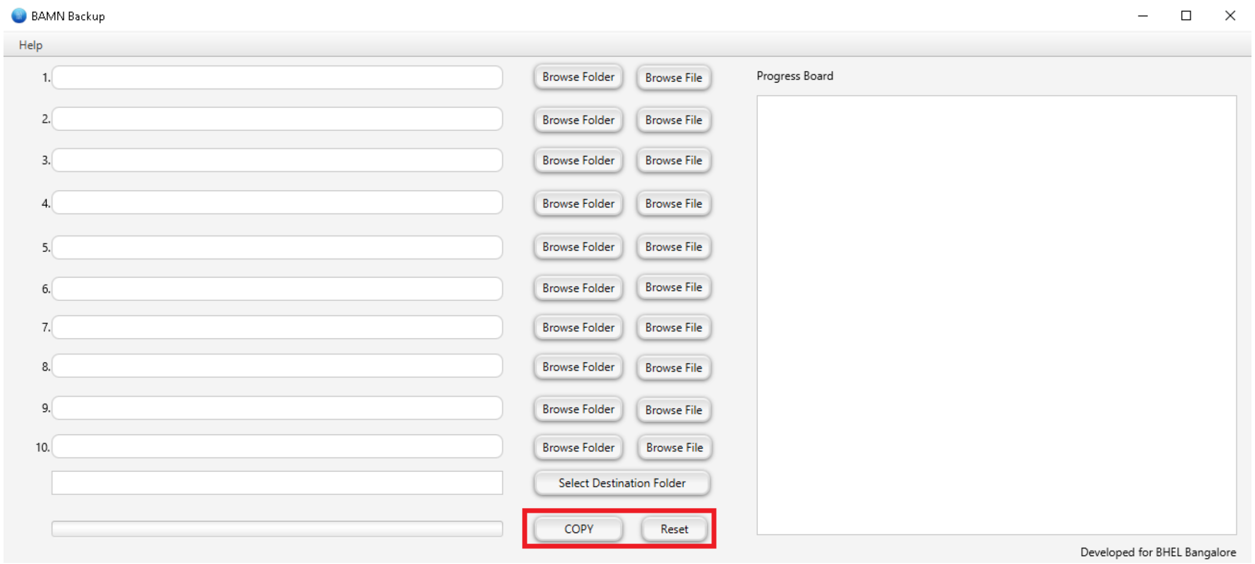


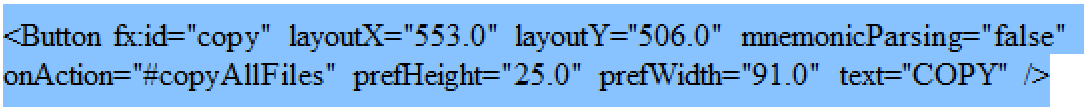
<Button fx:id="destfold" layoutX="553.0" layoutY="458.0" mnemonicParsing="false" onAction="#ButtonDestination" prefHeight="25.0" prefWidth="182.0" text="Select Destination Folder" />

Action for Select Destination Button

```
public void ButtonDestination(ActionEvent event) throws IOException {
    DirectoryChooser directoryChooser = new DirectoryChooser();
    selectedDest = directoryChooser.showDialog(null);
    if (selectedDest != null) {
        dest = selectedDest.getAbsolutePath();
        int length = dest.length();
        --length;
        char buff = dest.charAt(length);
        if (buff == '\\') {
            dest = dest.substring(0, 2);
        } else {
            dest = dest.replace("\\", "\\\\");
        }
        destinationview.getItems().clear();
        destinationview.getItems().add(selectedDest.getAbsolutePath());

    } else {
        Alert alert = new Alert(AlertType.ERROR);
        alert.setTitle("ERROR!");
        alert.setHeaderText("Error code : 2");
        alert.setContentText("Please Select a destination!");

        alert.showAndWait();
        System.out.println("File not selected");
    }
}
```

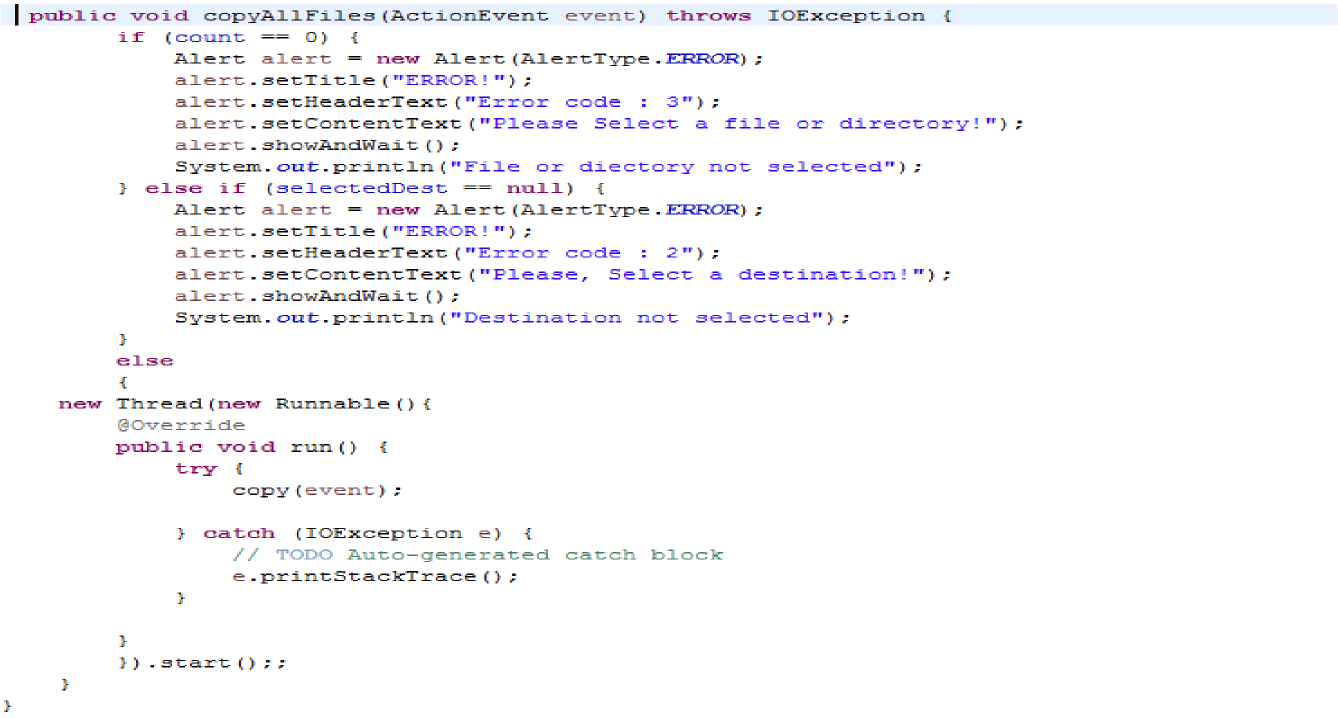
d. Copy Button





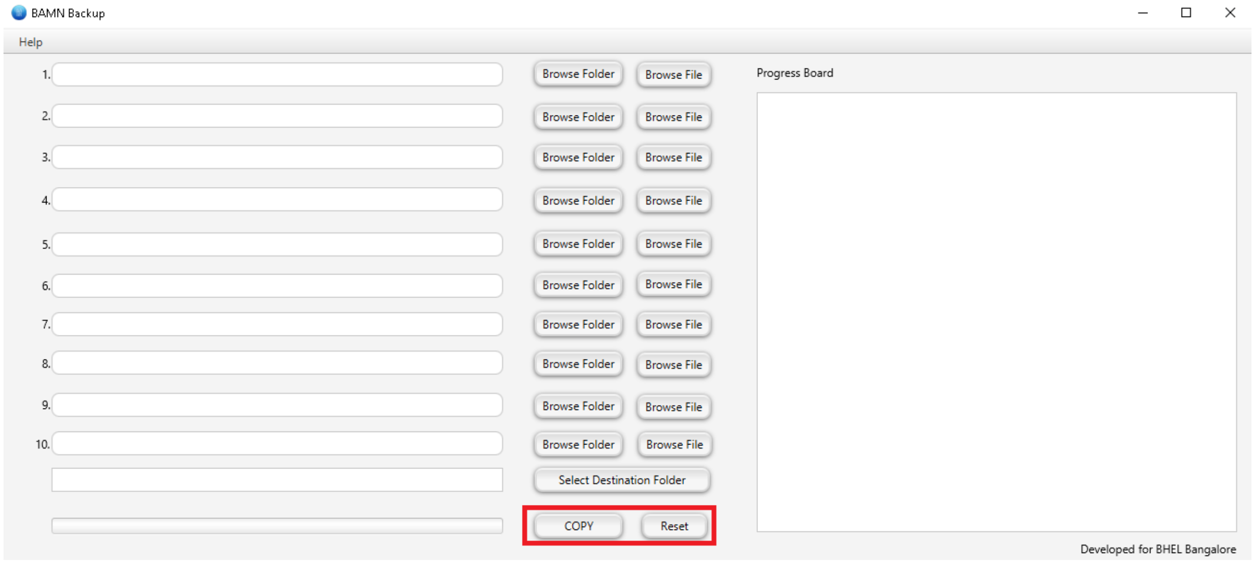
```
<Button fx:id="copy"  layoutX="553.0"  layoutY="506.0"  mnemonicParsing="false"
onAction="#copyAllFiles"  prefHeight="25.0"  prefWidth="91.0"  text="COPY"  />
```

Action for copy button



```
  public void copyAllFiles(ActionEvent event) throws IOException {
        if (count == 0) {
            Alert alert = new Alert(AlertType.ERROR);
            alert.setTitle("ERROR!");
            alert.setHeaderText("Error code : 3");
            alert.setContentText("Please Select a file or directory!");
            alert.showAndWait();
            System.out.println("File or diectory not selected");
        } else if (selectedDest == null) {
            Alert alert = new Alert(AlertType.ERROR);
            alert.setTitle("ERROR!");
            alert.setHeaderText("Error code : 2");
            alert.setContentText("Please, Select a destination!");
            alert.showAndWait();
            System.out.println("Destination not selected");
        }
        else
        {
    new Thread(new Runnable(){
        @Override
        public void run() {
            try {
                copy(event);

            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

        }
        }).start();;
    }
}
```

e. Reset Button



```
<Button fx:id="reset" layoutX="665.0" layoutY="506.0" mnemonicParsing="false"
onAction="#ButtonReset" prefHeight="25.0" prefWidth="67.0" text="Reset" />
```

Action for Reset button

```
public void ButtonReset(ActionEvent event) {
     System.out.println("Reset");
     count = 0;
     temp = 1;
     progress = 0;
     dest=null;
     listview1.getItems().clear();
     listview2.getItems().clear();
     listview3.getItems().clear();
     listview4.getItems().clear();
     listview5.getItems().clear();
     listview6.getItems().clear();
     listview7.getItems().clear();
     listview8.getItems().clear();
     listview9.getItems().clear();
     destinationview.getItems().clear();
     status.getItems().clear();
     pb.setProgress(0.0);
}
```

- **Commands**

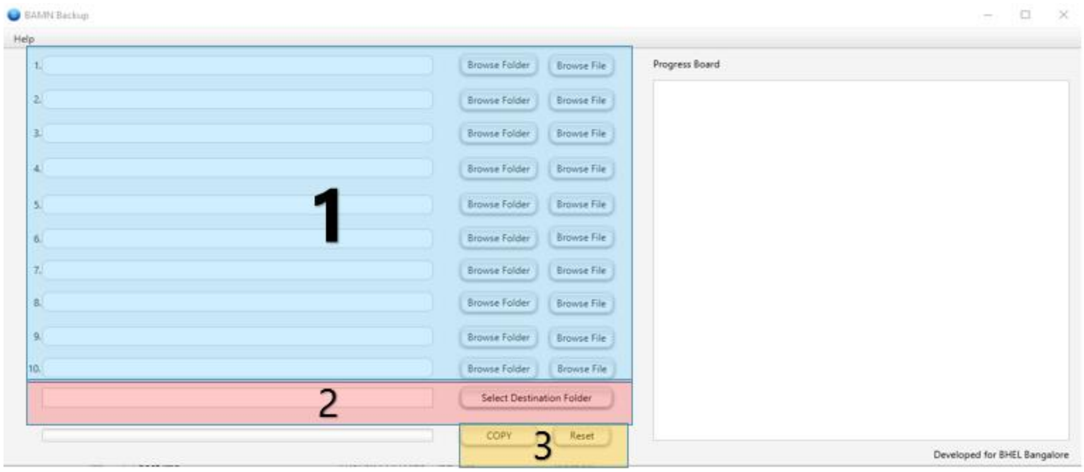
a. Browse Command

```
// Action of Browse Folder 1
public void Button1FolderAction(ActionEvent event) throws IOException {
        if (count==0) {
                DirectoryChooser directoryChooser = new DirectoryChooser();
                File selectedDir = directoryChooser.showDialog(null);
                if (selectedDir != null) {
                        count = count + 1;
                        path1 = selectedDir.getAbsolutePath();
                        path1 = path1.replace("\\", "\\\\");
                        name1 = selectedDir.getName();
                        listview1.getItems().add(selectedDir.getAbsolutePath());

                } else {
                        System.out.println("File not selected");
                }
        } else {
                Alert alert = new Alert(AlertType.ERROR);
                alert.setTitle("ERROR!");
                alert.setHeaderText("Error code : 1");
                alert.setContentText("Please, Select the sources in the ascending
order!");

                alert.showAndWait();
                System.out.println("File not Selected");
        }
}
```

b. Copy Command

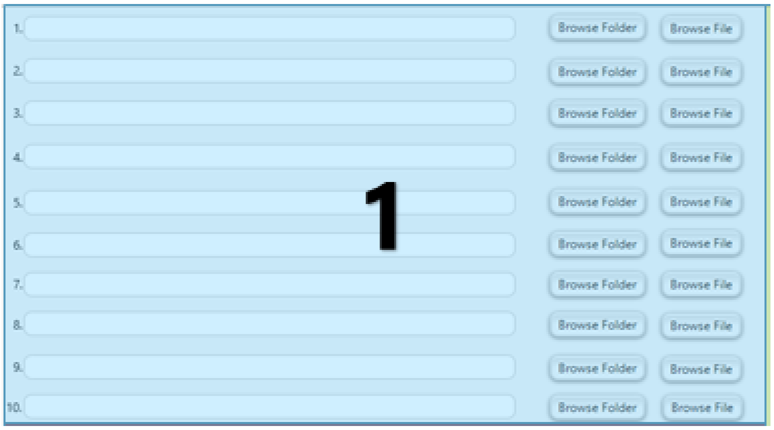
```
        // Copy function
        private static void copyFolder(File sourceFolder, File destinationFolder, ListView
status, ProgressBar pb,
                        double progress) throws IOException {
                // Check if sourceFolder is a directory or file
                // If sourceFolder is file; then copy the file directly to new location
                if (sourceFolder.isDirectory()) {
                        // Verify if destinationFolder is already present; If not then
                        // create it
                        if (!destinationFolder.exists()) {
                                destinationFolder.mkdir();
                                System.out.println("Directory created :: " + destinationFolder);
                        }
                        // Get all files from source directory
                        String files[] = sourceFolder.list();
                        // Iterate over all files and copy them to destinationFolder one by
                        // one
                        for (String file : files) {
                                File srcFile = new File(sourceFolder, file);
                                File destFile = new File(destinationFolder, file);
                                // Recursive function call
                                copyFolder(srcFile, destFile, status, pb, progress);
                        }
                } else {
                        // Copy the file content from one place to another
                        Files.copy(sourceFolder.toPath(), destinationFolder.toPath(),
StandardCopyOption.REPLACE_EXISTING);
                        pb.setProgress(progress);
                        System.out.println("File Copied ::" + destinationFolder);
                        status.getItems().add(destinationFolder.toPath());
                        status.refresh();
                }
        }
```

- **Procedures** : 3 steps



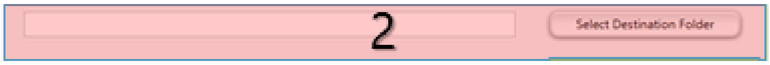
a. Select File or Folder

Select the files or folders which is to be copied.



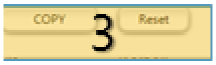
b. Select Destination Folder

Select the destination folder where the selected files are to be copied.



c. Click Copy or Reset Button

Click the copy button to copy the selected files or folders to the selected destination or click reset button to reset the application.

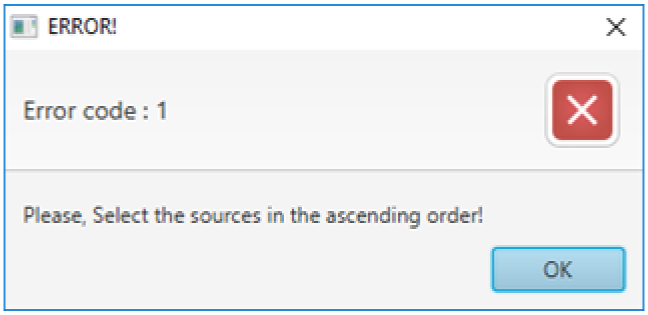


VALIDATION

- There are 3 types of errors in the validation phase. They are

1. **Error code 1**

   User should select sources in the ascending order . Otherwise an error popup will be shown.



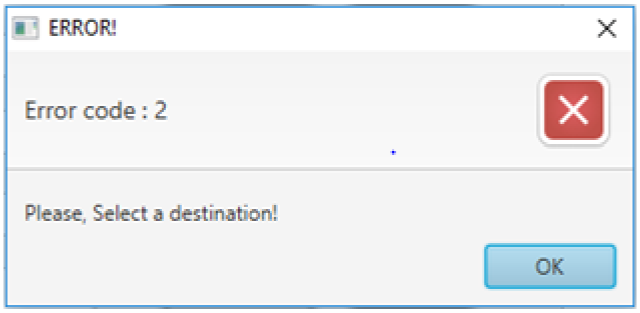
ERROR!

Error code : 1

Please, Select the sources in the ascending order!

2. **Error code 2**

   User should select a destination folder. Otherwise an error popup will be shown.



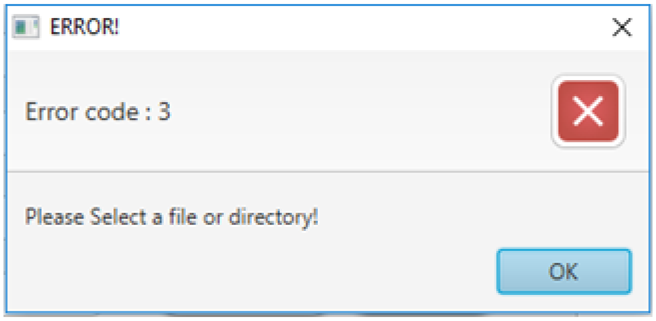
ERROR!

Error code : 2

Please, Select a destination!

3. Error code 3

User should select atleast one file or directory. Otherwise an error popup will be shown.



Error code : 3

Please Select a file or directory!

Bibliography

1. *https://www.youtube.com/watch?v=9YrmON6nlEw*

2. *https://docs.oracle.com/javafx/2/fxml_get_started/jfxpub-fxml_get_started.pdf*