

COPY WWW

A Project Work
submitted in partial fulfillment of the
requirements for the degree of

Bachelor of Technology
in
Computer Science and Engineering

Submitted By

KARTIKAY SINGH (1513101270)

APOORVA PANDEY (1515103015)

Under the supervision of
T. GANESH KUMAR
Associate Professor



SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA – 201306
MAY 2019

DECLARATION

Project Title: COPY WWW

Degree for which the project work is submitted: **Bachelor in Technology in Computer Science and Engineering**

I declare that the presented project represents largely my own ideas and work in my own words. Where others ideas or words have been included, I have adequately cited and listed in the reference materials. The report has been prepared without resorting to plagiarism. I have adhered to all principles of academic honesty and integrity. No falsified or fabricated data have been presented in the report. I understand that any violation of the above will cause for disciplinary action by the Institute, including revoking the conferred degree, if conferred, and can also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been taken.

APOORVA PANDEY

Enrollment No. 1515103015

KARTIKAY SINGH

Enrollment No.1513101270

Date: 08-05-2019

CERTIFICATE

It is certified that the work contained in this project entitled “**Copy WWW**” submitted by **Apoorva Pandey (1515103015)**, **Kartikay singh (1513101270)** for the degree of Bachelor in Technology in Computer Science and Engineering is absolutely based on his/her own work carried out under my supervision and this project work has not been submitted elsewhere for any degree.

T.Ganesh Kumar
Associate Professor
School of Computing Science and Engineering
Galgotias University
Greater Noida, UP, India

Date: 08-05-2019

Countersigned by

(Dr. Sanjeev Kumar Pippal)
Professor and Associate Dean
School of Computing Science and Engineering
Galgotias University
Greater Noida, UP, India

ABSTRACT

This project discusses about an application which will copy the files like the operating system but in a different manner. It has some additional features which makes it unique. Often, we copy various large files and we get an error and then we have to start from the starting. Keeping all things in mind this application has features to resume the copy file from where it failed. *Copy WWW* is a graphical copy manager. It provides more comfort and control over files copy operations than basic OS functions.

With *Copy WWW* it is possible to queue several transfers, resume a copy that has failed, limit the speed or pause the copy. If a transfer fails, an error is logged and the rest of the queue is treated. *Copy WWW* tries to include most of the features of Super Copier (Windows-only software) into a completely multi - platform program. In this application various shortcomings of operating system have been covered.

ACKNOWLEDGEMNT

We would like to express my special thanks of gratitude to my Guide Mr T Ganesh as well as our Dean Mr. Sanjeev Pippal who gave us the golden opportunity to do this wonderful project on the topic COPY WWW, which also helped us in doing a lot of Research and came to know about so many new things so we are really thankful to them.

Secondly we would also like to thank our friends who helped us a lot in finalizing this project within the limited time frame.

Apoorva Pandey 1515103015

Kartikay Singh- 1513101270

Table of Contents

DECLARATION.....	i
CERTIFICATE.....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMNT	iv
1.Introduction.....	1
1.1 Overview	1
1.2 Background Motivation	1
2.Overall description.....	2
2.1 Problem Statement	2
2.2 Proposed Statement.....	2
3. Purpose.....	3
4. Motivation and scope.....	3
5. Proposed Model.....	4
6.Website Offline Builder	5
7. Module split-up	6
8. PROJECT SOURCE CODE.....	7
9. Screenshot.....	30
9.1 Homepage	30
9.2 Transfer of file from one location to other.....	30
10.GANTT CHART	31
11.Conclusion	32
12. References.....	33

List of Figures

Figure	Title	Page number
Figure 7.1	Module Split-up	6
Figure 9.1	Screenshot part-1	30
Figure 9.2	Screenshot part-2	30
Figure 10.1	Gantt chart	31
Figure 10.2	Bar Graph	31

1.Introduction

1.1 Overview

This project discusses about the implementation of a copier software that will copy your files in more reliable way. This project is made in java and has been embedded to make an android application. There are various modules present in our project. There are various functions of java used in our project including abstraction, encapsulation and polymorphism. Various core java topics are included in this project. When you choose 'Copy this file' from the move or copy dropdown, you will then be able to select which project to copy it to. You can copy the file to the existing project, or to a different one.

1.2 Background Motivation

Operating system does various tasks for us. It creates a user interface between a user and a software. Apart from doing various input output operation it comes with handy amount of security functions and protection against your data. One more major function of operating system is to copy files from one directory to another. There are various shortcomings when it comes to copy files like, we can't resume the files after cancellation during copy. We can't skip the present ongoing files even though we don't want it to be copied. In this all new software we can eradicate all this shortcoming for a better flow during move/copy operation. What follows is the various function of this software.

- Treatment of the transfers one after the other
- Add new transfers to the queue while a copy is already being processed
- Dynamic management of the queue of remaining transfers
- Pause a copy

These are some function which can help us to transfer data in more reliable way.

2.Overall description

2.1 Problem Statement

While copying files through operating system we face many issues like time consuming, loss of data, can't resume the copy operation in between the process, we can't skip redundant data etc. These problem statements are covered in our proposed statement. While transferring of large files we may get annoyed if we get error and hence it's very time consuming.

2.2 Proposed Statement

Our proposed software includes every features of Super copier (Windows only software). In addition, it contains various features which are as follow.

- Skip current transfer to proceed to the next one
- Resume a copy at the exact point where it failed (no need to start over)
- Choose another name for the target if a file already exists
- Can follow or ignore symbolic links (Unix systems only)
- Set a default behavior if target file already exists
- Keeps track of the transfers which failed.
- Treatment of the transfers one after the other
- Add new transfers to the queue while a copy is already being processed
- Dynamic management of the queue of remaining transfers
- Pause a copy
- Limit the speed of the copy

3. Purpose

The Purpose of this project is to save time while transferring large amount of data. Big companies need such software as they have to transfer large amount of data to their directory. In case if it fails through operating system they have to start from the beginning. Hence this software can be a great asset for them as it will help them to encounter such problem. The installation with Windows should be straightforward. If you use the archive version, there is no need of installation, however you will need the Java Virtual Machine (Java Runtime Environment 5.0 minimum). This software is very reliable if we will compare it with another software. There is no software available in the market which can make the transfer of file easy.

4. Motivation and scope

In a given time frame, we can induct various features in this project along with all the above features mentioned to make a more reliable software easily. There are various shortcomings when it comes to copy files like, we can't resume the files after cancellation during copy. We can't skip the present ongoing files even though we don't want it to be copied. In this all new software we can eradicate all this shortcoming for a better flow during move/copy operation. You can launch Copy WWW with the script included in the archive. It is also possible, depending on your distribution, to right-click on the MiniCopier.jar file.

Drag and drop files or folders you wish to copy on the basket icon. You can do this several times, send multiple files simultaneously... When you want to paste the basket content, just drag and drop a destination folder on the icon on the right. While queue is being processed, you can, at any time, pause the copy, skip current transfer to proceed to the next one, or manage queue composition.

5. Proposed Model

Our proposed software includes every features of Super copier (Windows only software). In addition, it contains various features which are as follow.

- Skip current transfer to proceed to the next one
- Resume a copy at the exact point where it failed (no need to start over)
- Choose another name for the target if a file already exists
- Can follow or ignore symbolic links (Unix systems only)
- Set a default behavior if target file already exists
- Keeps track of the transfers which failed.
- Treatment of the transfers one after the other
- Add new transfers to the queue while a copy is already being processed
- Dynamic management of the queue of remaining transfers
- Pause a copy
- Limit the speed of the copy

6.Website Offline Builder

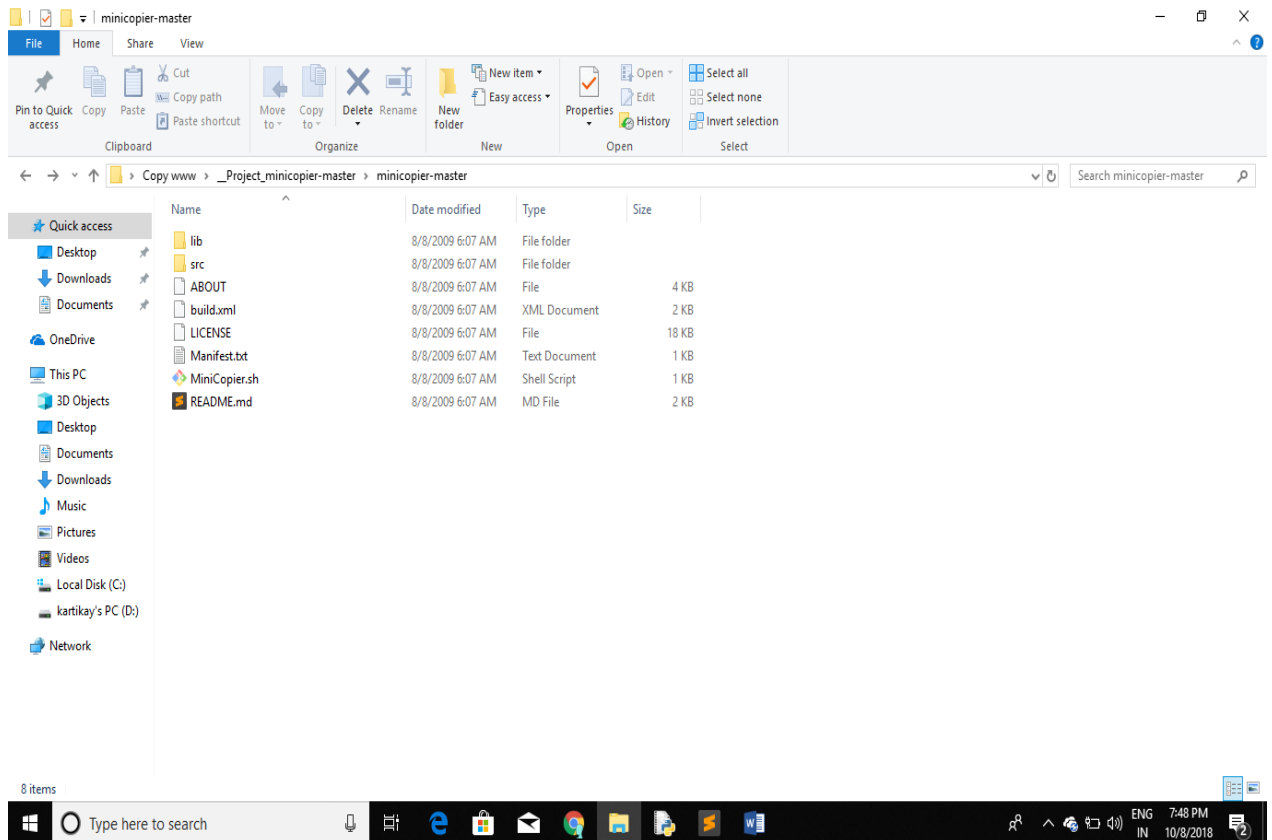
This feature of our project is quite overwhelming. We will try to add a feature along with Copy WWW that is to provide a website offline which is static. When website will be in offline mode you can easily search or browse through it without any internet connection and hence making your tech life quite easily. It has been noticed that where net connection is weak, its difficult to open a website and hence we can't have the information in plight situation. This feature will help people to search or browse the information if they have used this feature previously on some website.

Features of Website offline builder

- It will give you access to search or browse through any webpage without internet connection.
- It saves time and can be easily run irrespective of your mobile network strength.
- Helps to search the data in the minimum time.
- Gives you access to have the data of the website 24*7.

7. Module split-up

Our project is basically an application which is having various java files of different module as well as jar file. Our project contains lib, src, xml, sh script. It also includes GUI in which there are having many modules. Below image shows our project files.



8. PROJECT SOURCE CODE

8.1 lib

lib file contains PgslookAndFeel.jar file- A **JAR** (Java ARchive) is a package **file** format typically used to aggregate many Java class **files** and associated metadata and resources (text, images, etc.) into one **file** for distribution. **JAR files** are archive **files** that include a Java-specific manifest **file**.

8.2 src

src file contains the images with a folder named as *img* used in the software and the main folder named as the *copyWWW* that contain the code.

The files in the main folder are as following: -

8.2.1 gui

gui stand for (Graphical User Interface) which defines some very important files like:

- Credits
- DialogMsg
- FailedPanel
- MailFrame
- OptionsPanel
- QueueJButton
- SupPanel
- TransfersPanel

8.2.2 i18n

Internationalization Support. Internationalization is the process of designing an application so that it can be adapted to various languages and regions without engineering changes. Sometimes the term internationalization is abbreviated as **i18n**, because there are 18 letters between the first "i" and the last "n."

This folder contains the following files: -

- language
- copyWWW
- copyWWW_en
- copyWWW_fr

src file include some other files like:-

- configuration
- copier
- FailedQueue
- FileToTransfer
- MainQueue
- ThroughputlimitInputStream
- TransferTask

The code which are present in these files are given categorically:-

Credits

/*

Credits.java / COPY WWW

Developed by four students (Kartikay Singh, Prashant Kumar Singh, Apoorva Pandey, Vinay Raghav) from galgotias university. Its a final year project which consist of basic operation like copy paste which normal operating system do for us.

This program is free software.

*/

```
package minicopier.gui;
```

```
import java.awt.Component;  
import javax.swing.*;
```

```
import java.awt.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;
```

```
import minicopier.i18n.Language;  
import minicopier.Copier;
```

```

public class Credits extends JDialog {

    private JTextArea credits;

    private JTextArea license;

    private JButton ok;

    public Credits(JFrame owner){
        super(owner,true);

        this.setTitle(Language.get("Credits.Title"));
        this.setSize(360, 300);
        this.setLocationRelativeTo(null);

        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(new BoxLayout(mainPanel,
BoxLayout.Y_AXIS));
        mainPanel.setAlignmentX(Component.CENTER_ALIGNMENT);

        mainPanel.setBorder(BorderFactory.createEmptyBorder(7,7,7,7));

        //Panel with logo and version number
        JPanel logoPanel = new JPanel();
        logoPanel.setLayout(new BoxLayout(logoPanel,
BoxLayout.Y_AXIS));
        logoPanel.setAlignmentX(Component.CENTER_ALIGNMENT);

        JLabel logo = new JLabel("");
        logo.setIcon(new
ImageIcon(MainFrame.frame.imageFromJar("img/minicopier-logo.png")))

        return license.toString();

    }

    public void exit(){
        this.dispose();
    }

    private class ActionOK implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            exit();
        }
    }

}

```

DialogMsg:-


```
/*
DialogMsg.java / COPY WWW
Copyright (C) Galgotias university
```

Developed by four students (Kartikay Singh, Prashant Kumar Singh, Apoorva Pandey, Vinay Raghav) from galgotias university. Its a final year project which consist of basic operation like copy paste which normal operating system do for us.

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License as
published by the Free Software Foundation; either version 2 of
the License, or (at your option) any later version..
*/
```

```
package minicopier.gui;
import javax.swing.JOptionPane;
import minicopier.i18n.Language;
```

```
public class DialogMsg {

    //byte
    private static String b;

    private static String kb; //kB

    private static String mb; //MB

    private static String gb; //GB

    private static String tb; //TB

    private static String bps, kbps, mbps, gbps, tbps;

    private static String day, hour, min, sec;

    public static void init() {
        b = Language.get("DialogMsg.byte");
        kb = Language.get("DialogMsg.kilobyte");
        mb = Language.get("DialogMsg.megabyte");
        gb = Language.get("DialogMsg.gigabyte");
        tb = Language.get("DialogMsg.terabyte");
        bps = Language.get("DialogMsg.byteps");
        kbps = Language.get("DialogMsg.kilobyteps");
        mbps = Language.get("DialogMsg.megabyteps");
        gbps = Language.get("DialogMsg.gigabyteps");
        tbps = Language.get("DialogMsg.terabyteps");
        day = "d";
        hour = "h";
        min = "m";
```

```

        sec = "s";
    }

    /** Conversion of a long representing the size of a file
     * in bytes into a user-friendly String.
     * e.g. : 71430144 -> "68.12 MB"
     * @param s size in bytes
     */
    public static String prettySize(long s){
        String unit = "";
        double cut;

        if (s < 1024) {
            return s + " " + b;
        } else if (s < Math.pow(1024,2)){
            cut = ((double) s) / Math.pow(1024,1);
            unit = " " + kb;
        } else if (s < Math.pow(1024,3)) {
            cut = ((double) s) / Math.pow(1024,2);
            unit = " " + mb;
        } else if (s < Math.pow(1024,4)) {
            cut = ((double) s) / Math.pow(1024,3);
            unit = " " + gb;
        } else {
            cut = ((double) s) / Math.pow(1024,4);
            unit = " " + tb;
        }

        cut *= 100.0;
        cut = Math.floor(cut+0.5);
        cut /= 100.0;

        return cut + unit;
    }

    public static String prettySpeed(long s){
        String unit = "";
        double cut;

        if (s < 1024) {
            return s + " " + bps;
        } else if (s < Math.pow(1024,2)){
            cut = ((double) s) / Math.pow(1024,1);
            unit = " " + kbps;
        } else if (s < Math.pow(1024,3)) {
            cut = ((double) s) / Math.pow(1024,2);
            unit = " " + mbps;
        } else if (s < Math.pow(1024,4)) {
            cut = ((double) s) / Math.pow(1024,3);
            unit = " " + gbps;
        }
    }

```

```

    } else {
        cut = ((double) s) / Math.pow(1024,4);
        unit = " " + tbps;
    }

    cut *= 100.0;
    cut = Math.floor(cut+0.5);
    cut /= 100.0;

    return cut + unit;
}

public static String prettyTime(long t){
    StringBuffer sb = new StringBuffer();
    if (t < 60) { // x sec
        sb.append(t).append(sec);
        return sb.toString();
    }
    if (t < 3600) { // x min y sec
        sb.append(t/60).append(min).append(" ");
        t = t % 60;
        if (t > 0) {
            //if (t<10) sb.append("0");
            sb.append(t).append(sec);
        }

        return sb.toString();
    }
    if (t < 86400) { // x h y min
        sb.append(t/3600).append(hour).append(" ");
        t = t % 3600;
        if (t/60 > 0) {
            //if (t<10) sb.append("0");
            sb.append(t/60).append(min);
        }

        return sb.toString();
    }
    // x days y hours
    sb.append(t/86400).append(day).append(" ");
    t = t % 86400;
    if (t/3600 > 0) {
        sb.append(t/3600).append(hour);
    }

    return sb.toString();
}
}

```

FailedPanel:-

```
/*  
FailedPanel.java / COPY WWW  
Copyright (C) Galgotias University
```

Developed by four students (Kartikay Singh, Prashant Kumar Singh, Apoorva Pandey, Vinay Raghav) from galgotias university. Its a final year project which consist of basic operation like copy paste which normal operating system do for us.

```
This program is free software; you can redistribute it and/or  
modify it under the terms of the GNU General Public License as  
published by the Free Software Foundation; either version 2 of  
the License, or (at your option) any later version.  
*/
```

```
package minicopier.gui;  
  
import java.awt.BorderLayout;  
import java.awt.Component;  
import java.awt.Dimension;  
  
import javax.swing.*;  
import javax.swing.table.DefaultTableCellRenderer;  
import javax.swing.table.DefaultTableModel;  
  
import minicopier.i18n.Language;  
  
public class FailedPanel extends JPanel {  
  
    private MainFrame mainFrame;  
  
        protected QueueJButton retry;  
  
        protected QueueJButton removeFailed;  
  
        protected DefaultTableModel failedModel;  
  
        protected JTable failedList;  
  
        public FailedPanel(MainFrame f){  
            super();  
  
            this.mainFrame = f;  
  
            this.retry = new QueueJButton("img/retry.gif");  
            this.retry.setToolTipText(Language.get("Tooltip.Failed.Retry"));
```

```

        this.removeFailed = new QueueJButton("img/delete.gif");

        this.removeFailed.setToolTipText(Language.get("Tooltip.Failed.Clear"));

        failedModel = mainFrame.copier.failedItems.getTableModel();
        failedList = new JTable(failedModel);

        //transferList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        failedList.getColumnModel().getColumn(1).setCellRenderer(new
RightTableCellRenderer());
        failedList.getColumnModel().getColumn(1).setMaxWidth(80);
        failedList.getColumnModel().getColumn(1).setMinWidth(80);


        this.setLayout(new BorderLayout());


        JScrollPane jspFailed = new JScrollPane(failedList);
        jspFailed.setViewportView(failedList);
        jspFailed.setPreferredSize(new Dimension(40,40));

        //transferList.setAutoResizeMode(JTable.AUTO_RESIZE_LAST_COLUMN)
;

        //jspTransfer.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL
AL_SCROLLBAR_ALWAYS);


        //Subpanel (west) with queue management buttons
        JPanel queueButtonsPanel = new JPanel();
        queueButtonsPanel.setLayout(new
BoxLayout(queueButtonsPanel, BoxLayout.Y_AXIS));

        queueButtonsPanel.setAlignmentX(Component.CENTER_ALIGNMENT);

        queueButtonsPanel.add(this.retry, Component.CENTER_ALIGNMENT);
        queueButtonsPanel.add(this.removeFailed);


        this.add(queueButtonsPanel, BorderLayout.WEST);
        this.add(jspFailed, BorderLayout.CENTER);

    }

    private class RightTableCellRenderer extends DefaultTableCellRenderer {
        public RightTableCellRenderer() {
            setHorizontalAlignment(RIGHT);

```

```

        setVerticalAlignment(CENTER);
    }
}

```

MainFrame:-

```

/*
MainFrame.java / COPY WWW
Copyright (C) Galgotias University

```

Developed by four students (Kartikay Singh, Prashant Kumar Singh, Apoorva Pandey, Vinay Raghav) from galgotias university. Its a final year project which consist of basic operation like copy paste which normal operating system do for us.

```

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License as
published by the Free Software Foundation; either version 2 of
the License, or (at your option) any later version.
*/

```

```

package minicopier.gui;

import minicopier.i18n.Language;

import java.awt.*;
import java.awt.event.*;
import java.awt.dnd.*;
import java.awt.datatransfer.*;
import java.io.*;
import java.util.*;
import java.util.List;

import javax.swing.*;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;

import minicopier.Configuration;
import minicopier.Copier;

public class MainFrame extends JFrame{

    public static MainFrame frame;

    protected Copier copier;

    private SupPanel supPanel;

    private TransfersPanel transfersPanel;

```

```

private FailedPanel failedPanel;

private OptionsPanel optionsPanel;

public MainFrame(Copier cop) {

    super();

    frame = this;

    DialogMsg.init();

    //Look And Feel

    if (Configuration.lookAndFeel.equals("pgs")){
        try {

            UIManager.setLookAndFeel("com.pagosoft.plaf.PgsLookAndFeel");
        } catch(Exception e) {}

    }

    //Classic swing
    else {
        UIManager.put("swing.boldMetal", Boolean.FALSE);
        if (Configuration.lookAndFeel.equals("native")) {
            try {

                UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
                //SwingUtilities.updateComponentTreeUI(this);
            } catch (Exception e) {}

        }
    }
    this.copier = cop;
    copier.setGui(this);

    this.setTitle(copier.name);
    this.setSize(470, 400);
    //this.setAlwaysOnTop(Configuration.alwaysOnTop);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setLocationRelativeTo(null);
    this.setIconImage(imageFromJar("img/icon.jpg"));
    //this.setUndecorated(true);

    this.supPanel = new SupPanel(this);

    this.transfersPanel = new TransfersPanel(this);

    this.failedPanel = new FailedPanel(this);

```

```

this.optionsPanel = new OptionsPanel(this);

//mainPanel includes a JPanel (top) and a JTabbedPane
JPanel mainPanel = new JPanel();
mainPanel.setLayout(new BoxLayout(mainPanel,
BoxLayout.Y_AXIS));
mainPanel.setAlignmentX(Component.CENTER_ALIGNMENT);

mainPanel.add(this.supPanel);

// Linking the Panel to the controllers
RequestPauseResume actList = new RequestPauseResume();
ActionSkip skipList = new ActionSkip();
ActionCancel cancelList = new ActionCancel();
ActionCredits creditsList = new ActionCredits();
ActionBasket basketList = new ActionBasket();

supPanel.basket.addMouseListener(basketList);
supPanel.pause.addActionListener(actList);
supPanel.skip.addActionListener(skipList);
supPanel.cancel.addActionListener(cancelList);

DropTargetListener dropL = new Drop2Basket();
DropTargetListener dropFL = new Drop2Folder();
DropTarget dropT = new DropTarget(this.supPanel.basket, dropL);
DropTarget dropFT = new DropTarget(this.supPanel.paste, dropFL);

JTabbedPane tabbedPane = new JTabbedPane();

//Adding the transfers list to the first tab
tabbedPane.addTab(Language.get("MainFrame.TransfersPane.title"),
new ImageIcon(imageFromJar("img/list.png")), transfersPanel);
//Linking the Panel to the controllers
ActionQueueButtons queueButtonsListener = new ActionQueueButtons();
transfersPanel.putFirst.addActionListener(queueButtonsListener);
transfersPanel.putUp.addActionListener(queueButtonsListener);
transfersPanel.putDown.addActionListener(queueButtonsListener);
transfersPanel.putLast.addActionListener(queueButtonsListener);
transfersPanel.delete.addActionListener(queueButtonsListener);

tabbedPane.addTab(Language.get("MainFrame.FailedPane.title"), new
ImageIcon(imageFromJar("img/failed.gif")), failedPanel);
//Linking the Panel to the controllers

```



```

ActionFailedButtons failedButtonsListener = new ActionFailedButtons();
failedPanel.retry.addActionListener(failedButtonsListener);
failedPanel.removeFailed.addActionListener(failedButtonsListener);

//Adding the options panel to the second tab

    tabbedPane.addTab(Language.get("MainFrame.OptionsPane.title"),new
ImageIcon(imageFromJar("img/options.png")),optionsPanel);
        //Linking the Panel to the controllers
        ActionCollision collisionList = new ActionCollision();
        ActionSymbolic symbolicList = new ActionSymbolic();

        optionsPanel.collisionBox.addActionListener(collisionList);
        optionsPanel.symbolicBox.addActionListener(symbolicList);
        optionsPanel.credits.addActionListener(creditsList);
optionsPanel.speedLimitBox.addActionListener(new ActionLimitSpeed());
ActionSpeedChange speedLimitChange = new ActionSpeedChange();
optionsPanel.speedValue.addChangeListener(speedLimitChange);
optionsPanel.speedUnitBox.addActionListener(speedLimitChange);

        mainPanel.add(tabbedPane);

        this.getContentPane().add(mainPanel);
        this.setVisible(true);

        RefreshStats refresh = new RefreshStats(500);
        refresh.start();

    }

    public Image imageFromJar(String path){
        return Toolkit.getDefaultToolkit()
        .getImage(getClass().getClassLoader().getResource(path));
    }

    private class RequestPauseResume implements ActionListener {
        public void actionPerformed(ActionEvent e)
        {
            // to synchro (?)
            if(copier.getPause()) { //currently paused
                copier.unpause();
            } else { //currently copying
                copier.pause();
            }
        }
    }
}

```

```

private class Drop2Basket implements DropTargetListener {
    public void drop(DropTargetDropEvent dtde) {

        dtde.acceptDrop(DnDConstants.ACTION_COPY_OR_MOVE);

        try {

            Transferable tr = dtde.getTransferable();
            DataFlavor[] flavors = tr.getTransferDataFlavors();

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

                DataFlavor fl = flavors[i];

                Object obj = tr.getTransferData(flavors[i]);

                //For Windows and Mac
                if (DataFlavor.javaFileListFlavor.equals(fl)) {
                    List<File> files =
(List<File>)tr.getTransferData(fl);

                    for (File fi : files){

                        System.out.println(fi.toURI().toString());

                        copier.addURIString2basket(fi.toURI().toString());
                    }

                    dtde.dropComplete(true);
                    basketSizeSignal();
                    return;
                }

                //For Linux and Mac
                else if (obj instanceof String && "uri-
list".equals(flavors[i].getSubType())){

                    String allFilePath;
                    allFilePath = (String)obj;

                    Scanner scan =
new
Scanner(allFilePath.trim());

                    while (scan.hasNextLine()) {

```

```

        copier.addURIString2basket(scan.nextLine());
    }

    dtde.dropComplete(true);
    basketSizeSignal();
    return;

}

}

} catch (Exception e) {
    e.printStackTrace();
    dtde.rejectDrop();
}

dtde.rejectDrop();

}

public void dragExit(DropTargetEvent e){ }
public void dropActionChanged(DropTargetDragEvent e){ }
public void dragOver(DropTargetDragEvent e){ }
public void dragEnter(DropTargetDragEvent e){ }

}

private class Drop2Folder implements DropTargetListener {
    public void drop(DropTargetDropEvent dtde) {

        dtde.acceptDrop(DnDConstants.ACTION_COPY_OR_MOVE);

        try {

            Transferable tr = dtde.getTransferable();
            DataFlavor[] flavors = tr.getTransferDataFlavors();

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

                DataFlavor fl = flavors[i];

                Object obj = tr.getTransferData(flavors[i]);

                //For Windows and Mac
                if (DataFlavor.javaFileListFlavor.equals(fl)) {
                    List<File> files =
                        (List<File>)tr.getTransferData(fl);

```

```

files.iterator();

        Iterator<File>         ite         =

        if (files.size() == 1) {

        copier.paste2URIString(ite.next().toURI().toString());
        }

        dtde.dropComplete(true);
        basketSizeSignal();

        return;

    }

    //For Linux and Mac
    else if (obj instanceof String && "uri-
list".equals(flavors[i].getSubType())) {

        String allFilePath;
        allFilePath = (String)obj;

        Scanner         scan         =         new
Scanner(allFilePath.trim());

        String         destinationFolderURIString         =

        scan.nextLine();

        if (!scan.hasNextLine()) {

        copier.paste2URIString(destinationFolderURIString);
        }

        dtde.dropComplete(true);
        basketSizeSignal();

        return;

    }

}

} catch (Exception e) {
    e.printStackTrace();
    dtde.rejectDrop();
}

dtde.rejectDrop();

```

```

    }

    public void dragExit(DropTargetEvent e){ }
    public void dropActionChanged(DropTargetDragEvent e){ }
    public void dragOver(DropTargetDragEvent e){ }
    public void dragEnter(DropTargetDragEvent e){ }

}

public void changeCurrentFileSignal() {

    supPanel.currentFileSource.setText(this.copier.getCurrentFileSource());

    supPanel.currentFileDestination.setText(this.copier.getCurrentFileDestination(
));
}

public void changePauseStateSignal(boolean b) {
    if (b) { //Copier has been paused
        setTitle(Language.get("MainFrame.paused")+" " + getTitle());

        supPanel.pause.setText(Language.get("MainFrame.SupPanel.ResumeButton")
);
    } else { //Copier has been resumed
        //setTitle(Copier.name);

        supPanel.pause.setText(Language.get("MainFrame.SupPanel.PauseButton"));
    }
}

public void totalSizeSignal() {
    String totalSize = DialogMsg.prettySize(copier.getQueueTotalSize());
    supPanel.barTotalSize.setText(totalSize);
}

public void currentSizeSignal() {
    String currentSize = DialogMsg.prettySize(copier.getFileTotalSize());
    supPanel.barCurrentSize.setText(currentSize);
}

public void basketSizeSignal() {

    supPanel.basketContent.setText(Language.get("MainFrame.SupPanel.BasketC
ontent")+" "+copier.basketLength());
}

private class RefreshStats extends Thread {

```

```

//Delay between two displays (ms)
private int refreshRate;

public RefreshStats(int l){
    this.refreshRate = l;
}

public void setRefreshRate(int l){
    this.refreshRate = l;
}

public void run() {
    Vector<Long> historyBytes = new Vector<Long>();
    int maxSize = 20;

    while (true) {
        long timeInit = System.currentTimeMillis();
        if (!copier.getPause()){

            historyBytes.add(0,copier.readAndInitBytesCounter());
            if (historyBytes.size() > maxSize)
                historyBytes.removeElementAt(historyBytes.size()-1);
            double average = 0;
            for (long l : historyBytes){
                average += l;
            }
            try{
                average /=
(historyBytes.size()*refreshRate/1000f);

                //System.out.println(DialogMsg.prettySize((long)average));

                String totalETA = "%";
                String currentETA = "%";
                String currentSpeed = "";
                if (average > 0){
                    if (copier.getTotalBytesRemaining()>0)
                    {
                        totalETA += " - "
"+DialogMsg.prettyTime((long)(copier.getTotalBytesRemaining()/average));
                        currentSpeed = "
("+DialogMsg.prettySpeed((long)average)+")";
                    }
                    if
(copier.getCurrentBytesRemaining()>0)
                        currentETA += " - "
"+DialogMsg.prettyTime((long)(copier.getCurrentBytesRemaining()/average));
                }
            }
        }
    }
}

```

```

supPanel.mainBar.setValue(copier.getTotalPercent());

supPanel.fileBar.setValue(copier.getCurrentPercent());

supPanel.mainBar.setString(copier.getTotalPercent()+totalETA);

supPanel.fileBar.setString(copier.getCurrentPercent()+currentETA);

setTitle(copier.getTotalPercent()+" "+currentSpeed+" MiniCopier");
        } catch (Exception e){e.printStackTrace();}
    }
    try
        { Thread.sleep(refreshRate-
(int)(System.currentTimeMillis()-timeInit));}
    catch (Exception e){}

}

}

}

```

OptionPanel:-

```

/*
OptionsPanel.java / COPY WWW
Copyright (C) Galgotias University

```

Developed by four students (Kartikay Singh, Prashant Kumar Singh, Apoorva Pandey, Vinay Raghav) from galgotias university. Its a final year project which consist of basic operation like copy paste which normal operating system do for us.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

```

*/

```

```

package minicopier.gui;

import java.awt.Dimension;

import javax.swing.*;

import minicopier.Configuration;
import minicopier.i18n.Language;

public class OptionsPanel extends JPanel {

    private MainFrame mainFrame;

```

```

protected JComboBox collisionBox;

protected JComboBox symbolicBox;

protected JCheckBox speedLimitBox;

protected JSpinner speedValue;

protected JComboBox speedUnitBox;

protected JButton credits;

public OptionsPanel(MainFrame f) {
    super();

    this.mainFrame = f;

    this.credits = new JButton(Language
        .get("MainFrame.OptionsPane.creditsButton"));

    String[] collisionStrings = {
        Language.get("MainFrame.OptionsPane.collision.Ask"),
        Language.get("MainFrame.OptionsPane.collision.Cancel"),
        Language.get("MainFrame.OptionsPane.collision.Overwrite"),
        Language.get("MainFrame.OptionsPane.collision.OverwriteOld"),
        Language.get("MainFrame.OptionsPane.collision.Resume") };
    collisionBox = new JComboBox(collisionStrings);
    // DefaultListCellRenderer rightRenderer = new
    // DefaultListCellRenderer();
    // rightRenderer.setHorizontalAlignment(DefaultListCellRenderer.RIGHT);
    // collisionBox.setRenderer(rightRenderer);
    collisionBox.setSelectedIndex(Configuration.collisionAction);

    this.speedLimitBox = new
JCheckBox(Language.get("MainFrame.OptionsPane.throughput.Sentence"));
    this.speedLimitBox.setSelected(false);
    SpinnerModel speedModel = new SpinnerNumberModel(1,1,9999,1);
    this.speedValue = new JSpinner(speedModel);
    this.speedValue.setEnabled(false);
    String[] unitValues = {
        Language.get("DialogMsg.byteps"),
        Language.get("DialogMsg.kilobyteps"),
        Language.get("DialogMsg.megabyteps"),
        Language.get("DialogMsg.gigabyteps") };
    speedUnitBox = new JComboBox(unitValues);
    speedUnitBox.setSelectedIndex(2);
    speedUnitBox.setEnabled(false);

    String[] symbolicStrings = {
        Language.get("MainFrame.OptionsPane.symLinks.Follow"),

```



```

        Language.get("MainFrame.OptionsPane.symLinks.Ignore" ) };
symbolicBox = new JComboBox(symbolicStrings);

symbolicBox.setSelectedIndex(Configuration.symbolicLinkAction);

this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));

// Collision option
JPanel collisionPanel = new JPanel();
collisionPanel.setLayout(new BoxLayout(collisionPanel, BoxLayout.X_AXIS));
collisionPanel.add(new
JLabel(Language.get("MainFrame.OptionsPane.collision.Sentence")));

collisionPanel.add(Box.createHorizontalGlue());
collisionBox.setMaximumSize(collisionBox.getPreferredSize());
collisionPanel.add(collisionBox);

collisionPanel.setBorder(BorderFactory.createTitledBorder(Language.get("MainFram
e.OptionsPane.collision.Title")));

// Speed option
JPanel speedPanel = new JPanel();
speedPanel.setLayout(new BoxLayout(speedPanel, BoxLayout.X_AXIS));

speedPanel.add(this.speedLimitBox);
speedLimitBox.setMaximumSize(speedLimitBox.getPreferredSize());
speedPanel.add(Box.createHorizontalGlue());
speedPanel.add(this.speedValue);
speedValue.setMaximumSize(new Dimension(
    speedValue.getMinimumSize().width,
    speedUnitBox.getMinimumSize().height)
);
speedPanel.add(Box.createRigidArea(new Dimension(5,5)));
speedPanel.add(speedUnitBox);
speedUnitBox.setMaximumSize(speedUnitBox.getMinimumSize());

speedPanel.setBorder(BorderFactory.createTitledBorder(Language.get("MainFrame.
OptionsPane.throughput.Title")));

// Symbolic link option
JPanel symbolicPanel = new JPanel();
symbolicPanel.setLayout(new BoxLayout(symbolicPanel,
BoxLayout.X_AXIS));
symbolicPanel.add(new JLabel(Language
    .get("MainFrame.OptionsPane.symLinks.Sentence")));
symbolicPanel.add(Box.createHorizontalGlue());

```

```

symbolicBox.setMaximumSize(symbolicBox.getPreferredSize());
// symbolicBox.setMinimumSize(collisionBox.getPreferredSize());

symbolicPanel.add(symbolicBox);
symbolicPanel.setBorder(BorderFactory.createTitledBorder(Language
    .get("MainFrame.OptionsPane.symLinks.Title")));

JPanel optionsBottom = new JPanel();
optionsBottom.setLayout(new BoxLayout(optionsBottom,
BoxLayout.X_AXIS));
optionsBottom.add(this.credits);
optionsBottom.add(Box.createHorizontalGlue());

this.add(collisionPanel);
this.add(speedPanel);
this.add(symbolicPanel);
this.add(Box.createVerticalGlue());
this.add(optionsBottom);
}

}

```

QueueJbutton:-

```

/*
QueueJButton.java / COPY WWW
Copyright (C) Galgotias University

```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

```

*/

package minicopier.gui;

import java.awt.Dimension;

import javax.swing.ImageIcon;
import javax.swing.JButton;

public class QueueJButton extends JButton{

    public QueueJButton(String imagePath) {
        super();
        this.setIcon(new
ImageIcon(MainFrame.frame.imageFromJar(imagePath)));
        this.setMaximumSize(new Dimension(28,28));
        this.setPreferredSize(new Dimension(28,28));
    }
}

```

```

    }

    //Source file path
    private String sourcePath;

    //File's copy destination path (folder)
    private String destinationPath;

    //Destination File Path
    private String destinationFilePath;

    //Name of source file
    private String name;

    //Size of source file
    private long size;

    /** Creation of a FileToTransfer
     * @param _path2source system path to the source file
     * @param _path2destinationfolder system path to the destination folder
     */
    public FileToTransfer(String _path2source,
                          String _path2destinationfolder) {

        //Using the argument Strings directly work, but the
        //display is not fine for Windows, so we create Files
        //and get the path (OS-dependent).

        //this.sourcePath = _path2source;

        File sourceFile = new File(_path2source);

        this.sourcePath = sourceFile.getPath();
        this.name = sourceFile.getName();
        this.size = sourceFile.length();

        //this.destinationPath = _path2destinationfolder;

        File destFolder = new File(_path2destinationfolder);
        this.destinationPath = destFolder.getPath();

        this.destinationFilePath = this.destinationPath
        + File.separator + this.name;

    }

    public String getSourcePath() {
        return this.sourcePath;
    }

```

```

    }

    public File getSourceFile() {
        return new File(this.sourcePath);
    }

    public String getDestinationFolderPath() {
        return this.destinationPath;
    }

    public File getDestinationFolder() {
        return new File(this.getDestinationFolderPath());
    }

    public String getDestinationFilePath() {
        return this.destinationFilePath;
    }

    public File getDestinationFile() {
        return new File(this.getDestinationFilePath());
    }

    public String getName() {
        return this.name;
    }

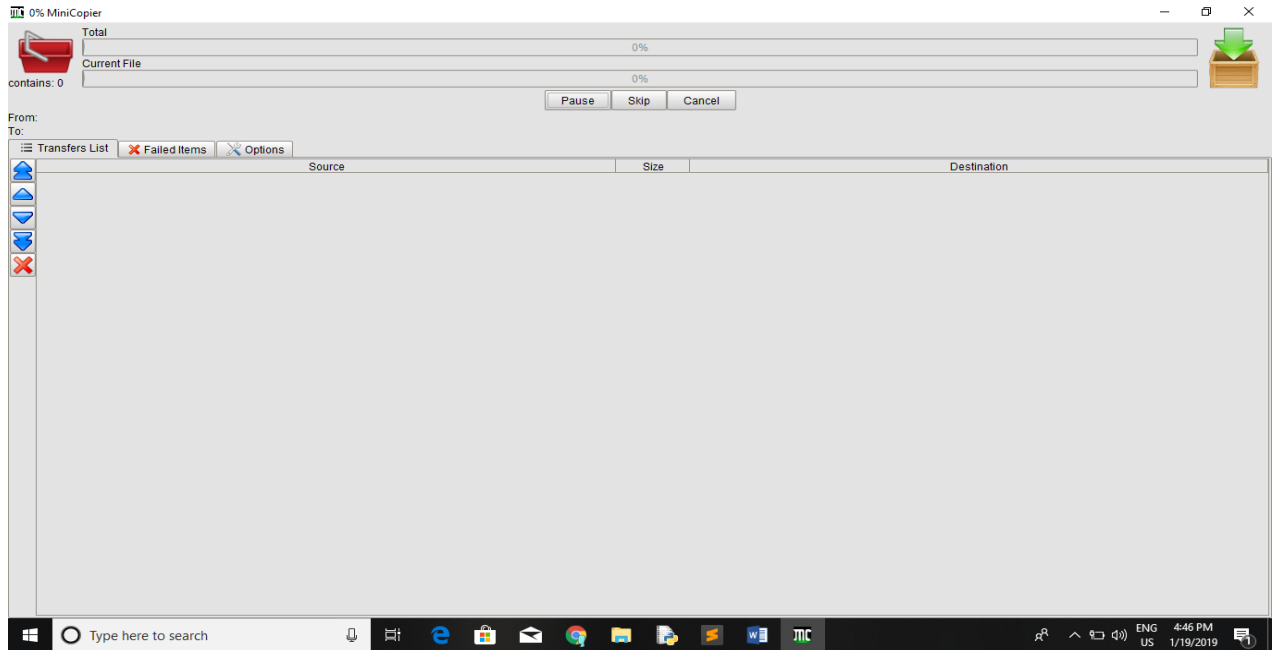
    public long getSize() {
        return this.size;
    }

    public void changeTargetName(String newName){
        this.destinationFilePath = this.destinationPath + File.separator +
            newName;
    }
}

```

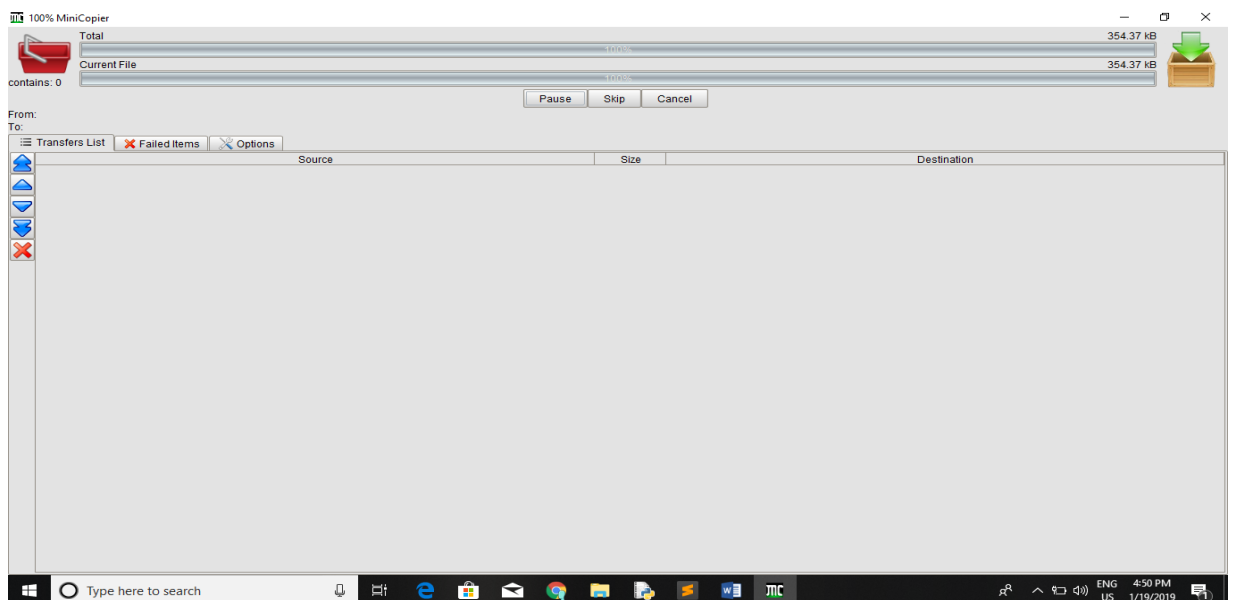
9. Screenshot

9.1 Homepage



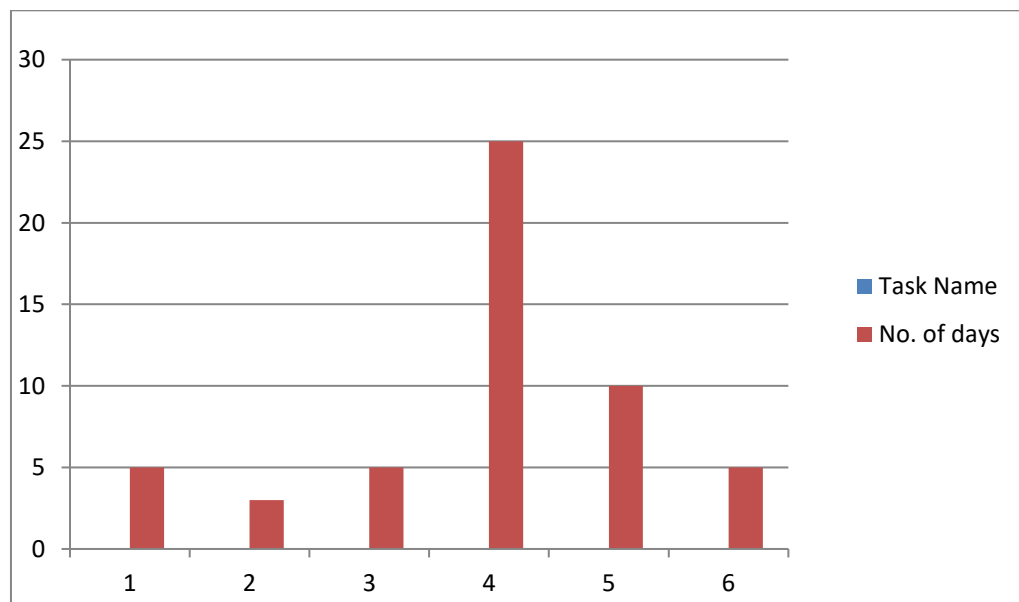
9.2 Transfer of file from one location to other

For Transfer of files from source to destination you need to drag and drop the file in the red basket. The destination folder should be drag and drop in the green basket.



10.GANTT CHART

TASK	Task Name	No. of days
TASK 1	Requirement and Analysis	5
TASK 2	Design	3
TASK 3	Function Requirement	5
TASK 4	Coding	25
TASK 5	Testing	10
TASK 6	Debugging	5



11.Conclusion

Copy WWW is just a software right now which just copy files from source folder to destination folder with some unique features which are not present in any other software till now. As discussed above the main features of this software is –

- Treatment of the transfers one after the other
- Add new transfers to the queue while a copy is already being processed
- Dynamic management of the queue of remaining transfers
- Pause a copy

In addition, we are thinking to add an overwhelming feature which will work like a website offline builder with the help of which a user can download the offline version of website quite easily. This will help the user in many ways, one of the main cases where this feature is going to help user is when the user is on low network, they can take help of the offline website to search information. The static website will be easily downloaded on the user system which the user can use it for the reference later.

Some of the main features will be-

- It will give you access to search or browse through any webpage without internet connection.
- It saves time and can be easily run irrespective of your mobile network strength.
- Helps to search the data in the minimum time.

12. References

- 1- JavaTpoint- <https://www.javatpoint.com/>
- 2- <https://www.youtube.com/watch?v=yeeuNx5xJw8>
- 3- Java Concurrency in practice- Brian Goetz (Book Name)
- 4- <https://www.geeksforgeeks.org/java/>