PDPU

PANDIT DEENDAYAL PETROLEUM UNIVERSITY

README FILE
OS LAB PROJECT

Team 34: Page Replacement Algorithm with Blady's Anomaly GUI

Leader: Ronak Makwana (18BCP093)
Team Members: Parv Chauhan (18BCP077)
Prachi Doshi (18BCP079)
Pratvi Shah (18BCP083)

Submission Date: 11th December, 2020
Submitted to: Professor Chintan Patel

Table of Contents

Introduction:

Page Replacements algorithm is one of the most important concepts of the Operating System. The major drawback of any operating system is its speed and memory allocation process. The number of memory calls done to access frequently used pages, is tiresome and affects the speed, accuracy and durability of the computer hardware and the other software present in the system.

To overcome this issue, operating system came up with the concept of virtual memory management. Using the concept of storing the pages in the virtual memory, page replacement algorithms is introduced as a solution to paging issues faced in the operating systems.

In this project, we are going to focus on the three most important algorithms of the Page Replacement Topic. We are going to show how the algorithms are implemented, we are going to provide you with algorithm calculator, and a comparison graph simulator, that helps you to compare between the all the algorithm for the similar kind of input.

IDE Installation:

For the successful implementation of the project, our team has used the eclipse IDE and Java Programming Language. Java Programming Language is the most efficient ad secure language and it has major framework or multiple packages for the GUI development.

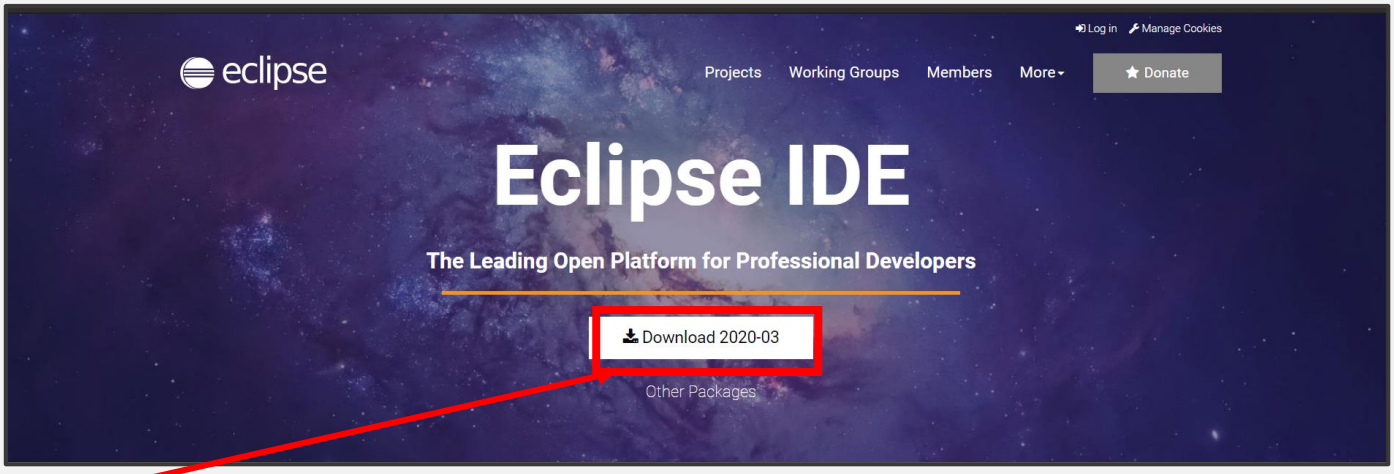
To implement the codes and import the libraries we will need Eclipse IDE.

The installation process include:

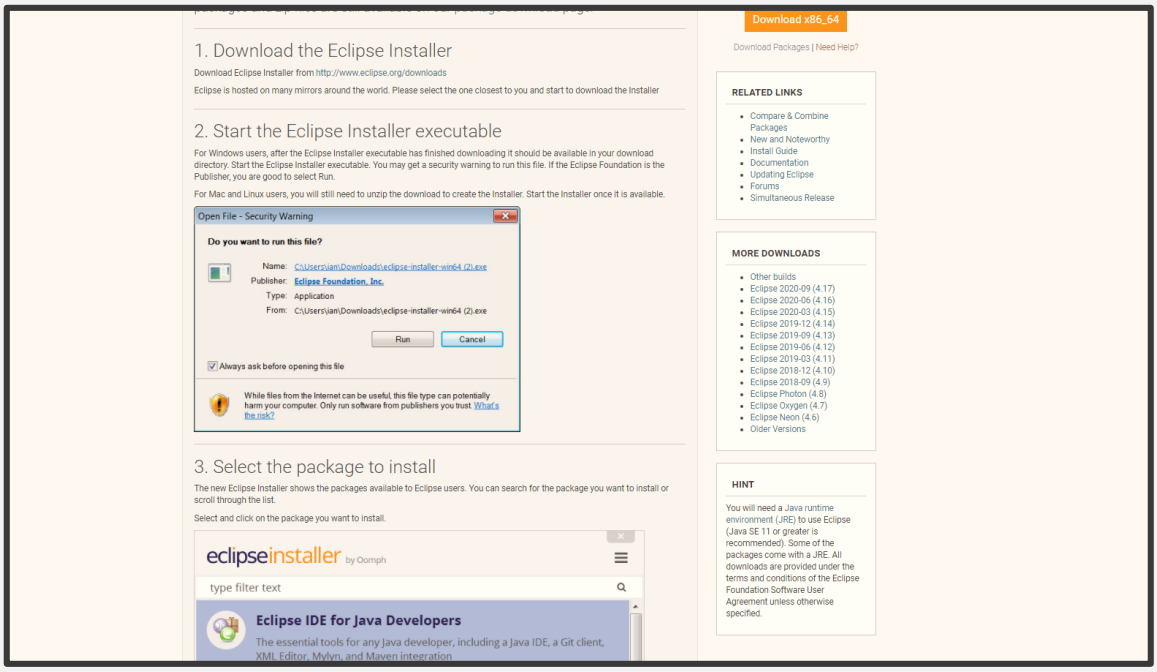
- Install Eclipse IDE, version compatible above 4.13.
- Install Java interpreter with eclipse, version compatible above jdk 11
- Build a java src workspace successfully.
- Creating your first Java Project.

1. **Installing Eclipse IDE**
   We have used Eclipse 2020-03(4.15) which is one of the latest version of eclipse. There have advancement in this version, as every three months they update the system.

   a. Go to the Eclipse 2020-03 homepage to download the software. https://www.eclipse.org/eclipseide/2020-03/



Click here



b. After clicking the button, it will redirect you to another webpage, that describe all the steps for Installation

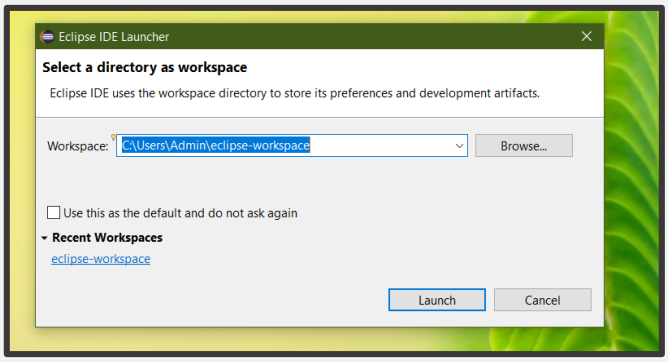
c. While installing the software, in the step 3 make sure you download **Eclipse IDE for Java Developers or the 1st Option.**

d. Save the eclipse software on your C: drive.

e. After installing the software. Launch the software.

f. Once you launch the software you will be asked to set up workspace.

Launching Software

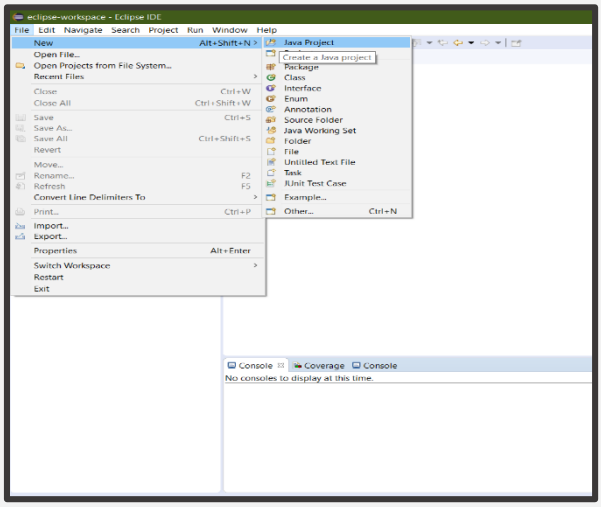


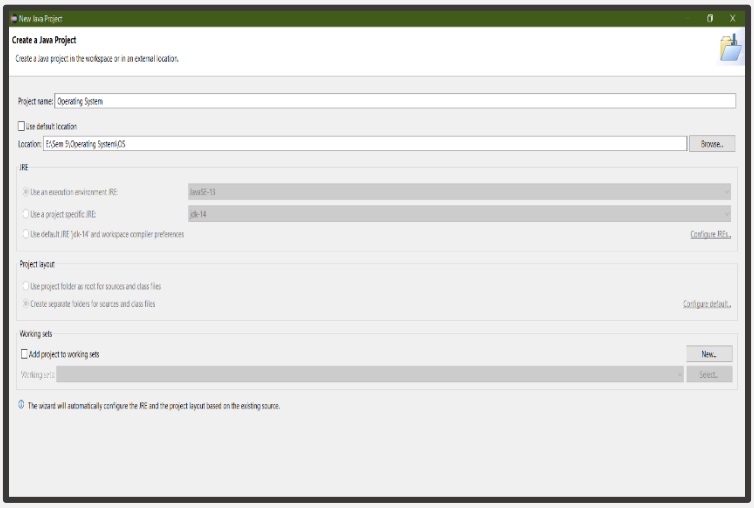
Workspace Setup

g. While setting up the workspace we would recommend you to setup the worksapce in either E: drive or any other place apart from C: drive. This will help your heavy threads and error to be detected smoothly. Also once the GUI is ready it will be easy you to run such heavy file seamlessly

h. Voila you Eclipse IDE is ready to run and now lets create our first project where we are goin gto create our GUI.

2. **Creating your first java project.**
    a. Once you launch the Eclipse IDE, you will receive a blank code area with a predefined default repository.
    b. Go to File -> New -> Project -> A dialog box will pop up





    c. Name the Java Project as **“Operating System”** and save the filein your default repository/workspace and click **“Finish”.**
3. Voila you have created your first java project, now let us write some code into it. But to run this code we have to import some external libraries that do not come predefined with the IDE. So, let’s download those libraries and then set a class path for it.
4. If you are not yet clear with any of the step or are facing some downloading issues. Please check out this YouTube link on [How to downlaod Eclipse IDE 2020-03.] This video will help you to download and help you to navigate through all the stuff easily.

Library Installation :

For this project we have used 3 external libraries, out of which 1 can be directly downloaded from the eclispe IDE interface and other two can be downloaded from their respective web pages and then attach their classpath to our Operating System Workspace.

1. **Instaling SWT Designers kit:**

   This kit is eminent part of our project. We need to download this kit to use the **Windows Builder** feature provided by the Java Developers and can create **UI** of our desktop applicatin with **Drag and Drop** method and then the parser parses the generated codes for us. This methods save a lot of time rather than writing a tedious code.

   a. **To check whether you have pre-installed SWT Designers kit.**
   Go to File → New → Other. The "select a wizard" window pops up.

   Double click in WindowBuilder folder and then to Swing Designer subfolder.

   Click to the Application Window and then click Next.

   Give a Name for your new window and then click Finish. The Source code of the new Window pops up.

   Press Run (the "Play" icon of the toolbar) to run your newly created window.

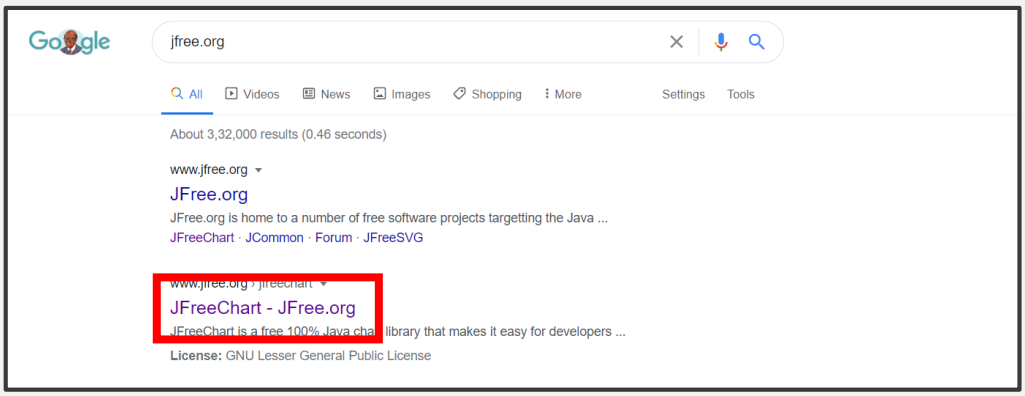
   b. **If you don't have pre-installed SWT Designer kit.**
   Go to Help → Install New Software → From the **work with** drop select any one Option → Find for SWT Designer and install all the files
   Voila you are done setting up the your SWT designer.

2. **Installing and Building Path for JfreeChart**

   The JFreeChart kit or set of jar files is one of eminent part of the visualisation portion of our project. This is an opensource .jar file created by the Java Developers for the ease of the graph visualitions and dataset analysis and comparision charts etc.

   a. **The method to download the JFreeChart .jar file.**

   Search for jfree.org online → Click on the 2nd link(that says JFreeChart – jfree.org)

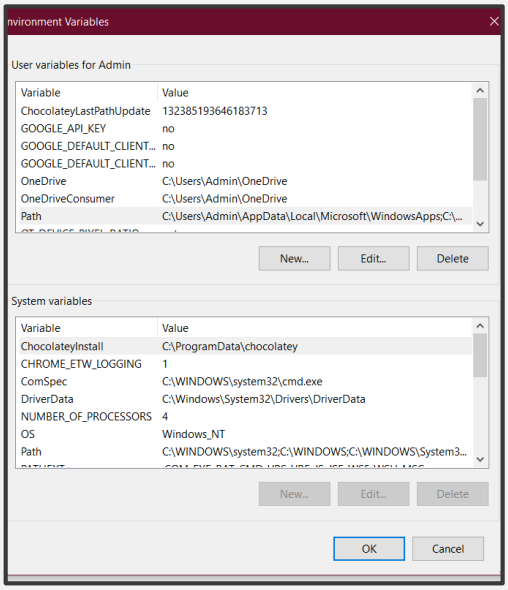


Click on the Download tab → It will redirect you to the Github repository → Click on the file with **(.zip)** extension → Save the dowloaded file anywhere in your computer.

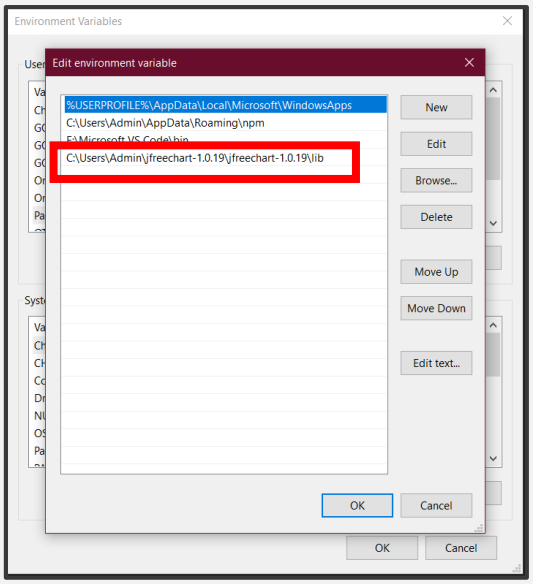
Extract those (.zip) file → Extract it at the place where you have stored your project for easy fetching

b. **Building the path for this referenced library**

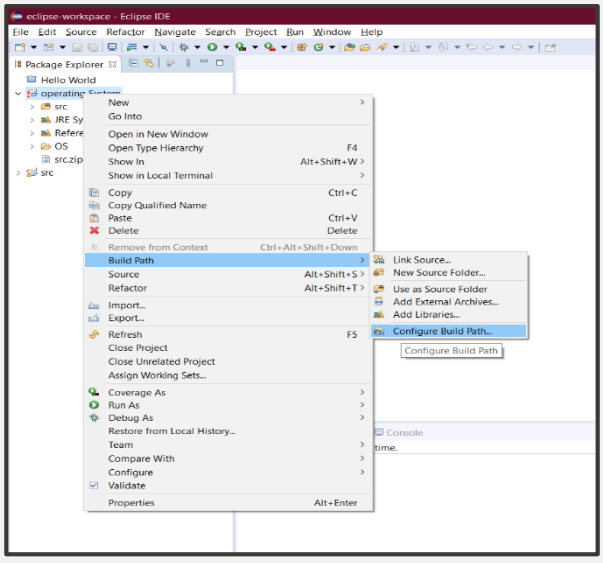
Type Environment Variables in the search tab of your laptop → A screen similar to the one shown below will show up.



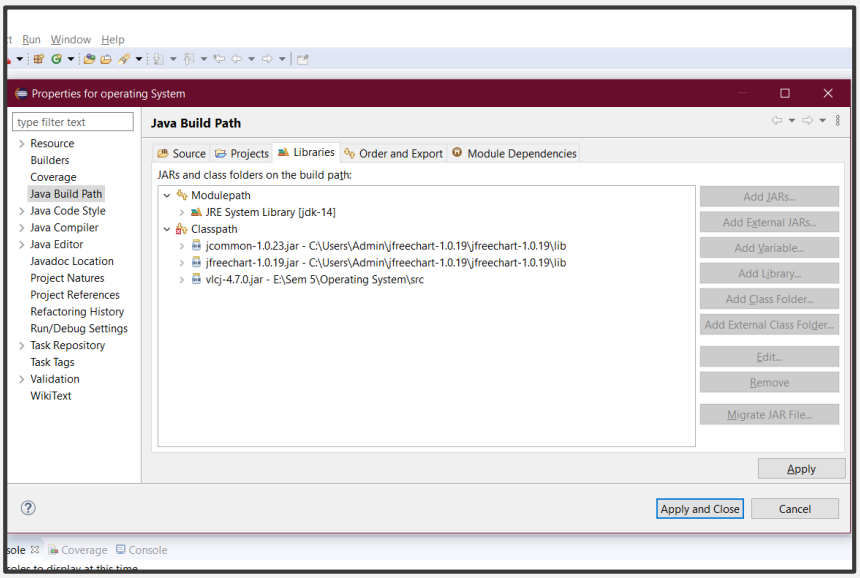
Choose Path → Click new → Go to place where you have extracted your files → select jfreechart-1.0.19 → select jfreechart-1.0.19 → select lib → click on Open.



Now launch Eclipse IDE → Right Click the project → Click Build Path → Click Configure Build Path



A dialogue box appear → Click on Add External Jar Files → Execute the same process as we did for setting up the Environment Variable Path.



Voila, your JFreeChart is added to the project to be use. For crosscheck, Click the dropdown arrow on the left of the project → Click the drop down arrow on the left of **Referenced Libraries** → JFreeChart Libraries would have been installed.

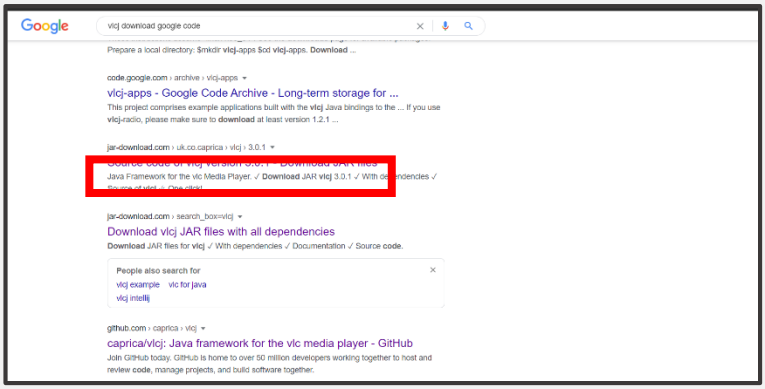
3. **Installing and Building Path for vlcj library**

   vlcj library is an extension to embed your customized vlc media player in java. We have used this library to show simulation in the simulation tab and to attach the voice over. This library helps to attach videos, animations music and customize the control buttons.

   a. The method to download the **vlcj .jar** file

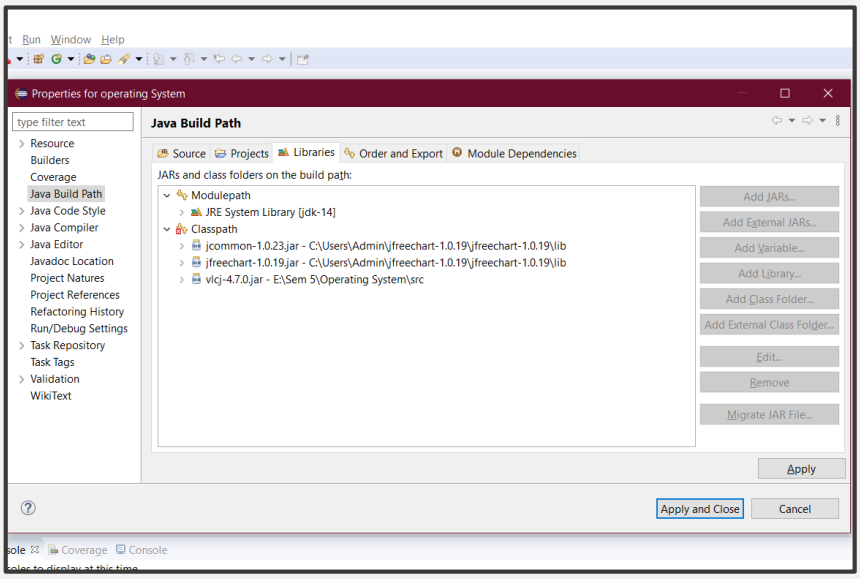
      Search for vlcj download google code online → Click on the link that says download vlcj with all dependencies → Check whether the version is 4.7.0 else do not download. → Save the (.zip) file in the anywhere.

      Note: Before installing vlcj make sure, you have **VLC** media player x64-bit version downloaded in your laptop



Download the first link → After saving the .zip file; extract it in the folder where the project is saved

Launch Eclipse IDE → Follow the same steps that you did to build path for the JFreeChart → Just make sure that this time you are attaching the **vlcj** files and not the JFreeChart files



Voila, all the libraries are set up and your project is ready to extract all the .java files now.

Other than, above mentioned External .jar files we have used some inbuilt libraries:

1. **Javax Swing Package** for creating and viewing desktop app
2. **Java AWT Package** for drawing and setting dimension and external layout in the Desktop App
3. **Javax Windows Builder** method to create the visuals in the GUI
4. **Java language Exception package** to handle all the possible exceptions in the code.
5. **Java language Thread class** to change between multiple JFrames smoothly and effortlessly.

Extract and Run:

Now you have imported and attached all the libraries that will be needed to run the project.

Now extract the .zip file → Go to src folder → Copy paste all the files to the src folder of your project.

Launch the Eclipse IDE → Compile all the codes one by one → Check whether there is some error or not → Run

OR

Create two packages **images** and **mywork** in the **src** folder of your project.

Create Application window for **homepage.java, fifo_sim.java, OPR_sim.java and lru_sim.java**

While for the remaining file create a JFrame class.

After that open all the java file in our project in the Notepad and copy-paste all the codes to their respective classes that you just created in your IDE

Compile the code → Run the code.

Note: Before running the code change the location of VIDEO_PATH in all the simulator files i.e. fifo_sim, opr_sim and lru_sim. Change it according to the places you have saved the video.

Conclusion :

All the above algorithms and tabs are running properly with no errors in the code.
Through this project our team got the chance to learn following things given below:

- We learnt that with team work we ca achieve our goal in limited time
- We learnt how to implement different libraries of java and connect all the frames to each other for seamless user experience
- We learned the concept of page replacement algorithm thoroughly.
- We learned to implement animationa and Desktop App with video, voicevers, charts, graphs and content.

Hence, team 34 have implemented all the algorithms successfully within the given time frame and have submitted their video potraying the working of the code properly.

Thanking Chitan sir and Samir Sir for giving us this eminent opportunity to create this GUI and guiding us throughout out project. We are pleased to announce that we have completed our work and will be waiting for your suggestions and constant support in the future.

References :

1. https://www.geeksforgeeks.org/page-replacement-algorithms-in-operating-systems/#:~:text=Optimal%20page%20replacement%20is%20perfect,can%20be%20analyzed%20against%20it.&text=In%20this%20algorithm%20page%20will%20be%20replaced%20which%20is%20least%20recently%20used. (For the content in the report)
2. https://www.flaticon.com/authors/flat-icons (For all the transparent icons used in GUI)
3. https://www.jfree.org/jfreechart/ (For graph in GUI)
4. https://capricasoftware.co.uk/projects/vlcj (For video and voice over animation ins GUI)
5. https://www.eclipse.org/eclipseide/ (For Eclipse IDE 2020)