# Person Information System

## 1.Tool Description:

1. This tool is used to store the data about person (first name, last name, title, phone, age, email).
2. Main Functions are :
   a. Add a new person's information.
   b. Update person's information.
   c. List person information inside a JSON file.
   d. Filter based on any field selected.
   e. Sort based on any field selected.
   f. Delete any person's information.

## 2.Setting up the environment:

1. Any "Eclipse" version is needed and click on this link to download it: link.
2. After you've downloaded it you need to install some packages to it:
   - Press "Help" on the tool bar.
   - Install new software.
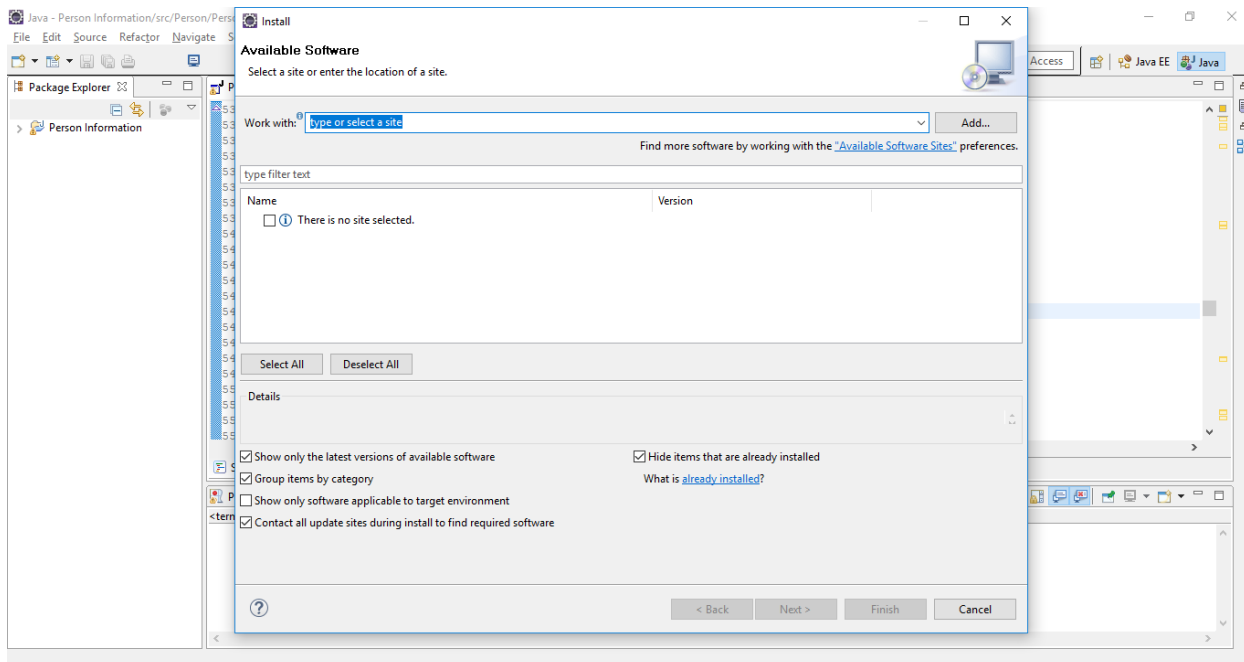
- Work with: Choose all available sites--.
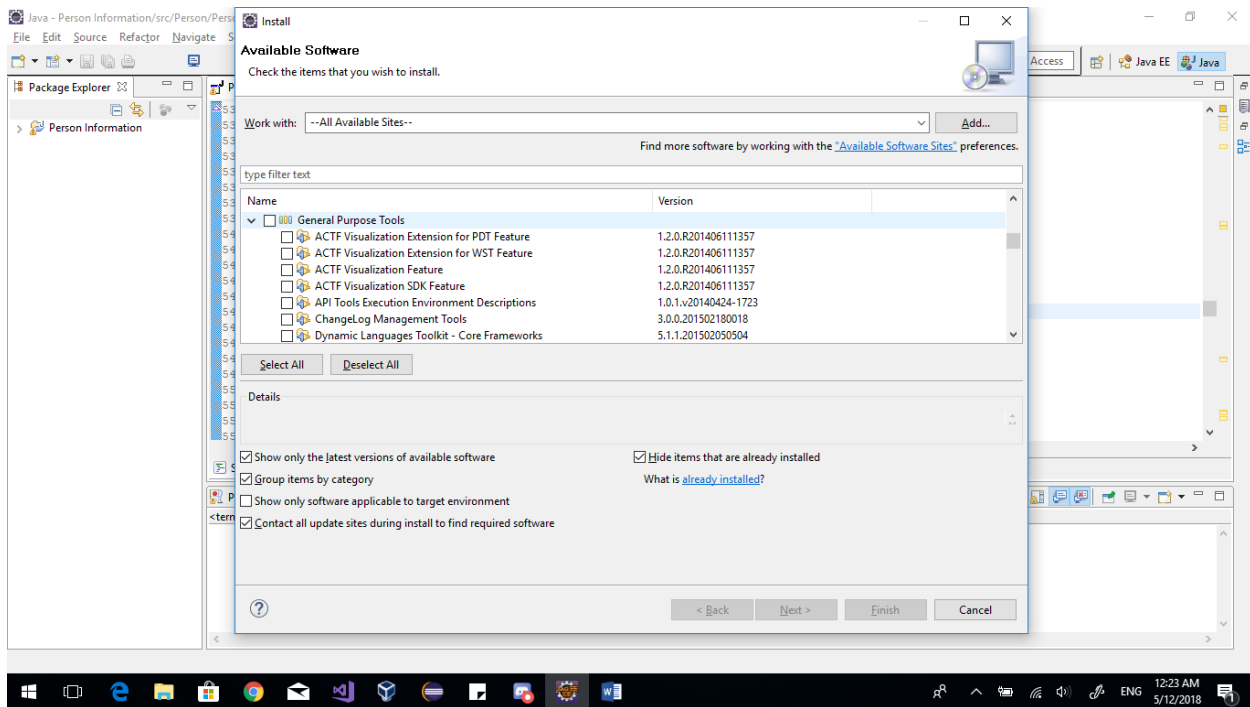


- After the link has finished pending you will get all packages you only need to choose from "General Purpose Tools" the following packages.
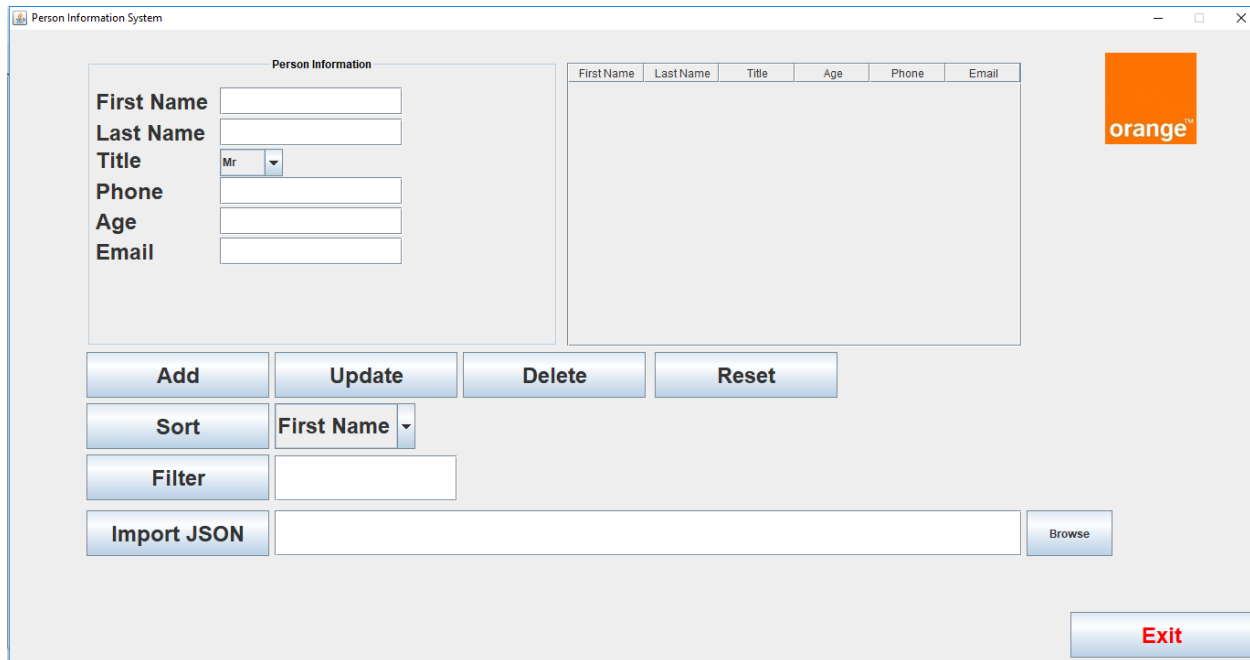
- Now you can open the project from eclipse from File -> Import -> General ->Existing projects to workspace.
- Finally you will have 1 error you need to change a path of JSON simple jar file to the new path you've downloaded the project to by going to the project properties -> Java Build Path.
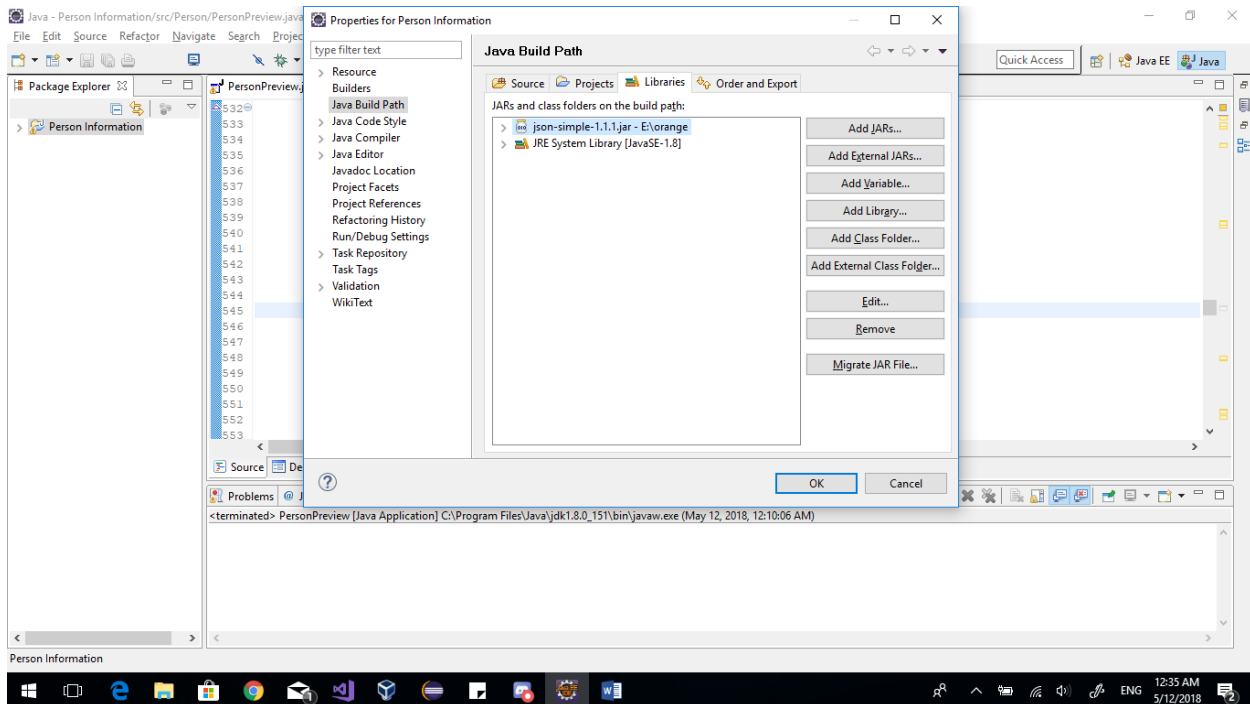


- A bit long but now you can finally run the project and go to the next phase.
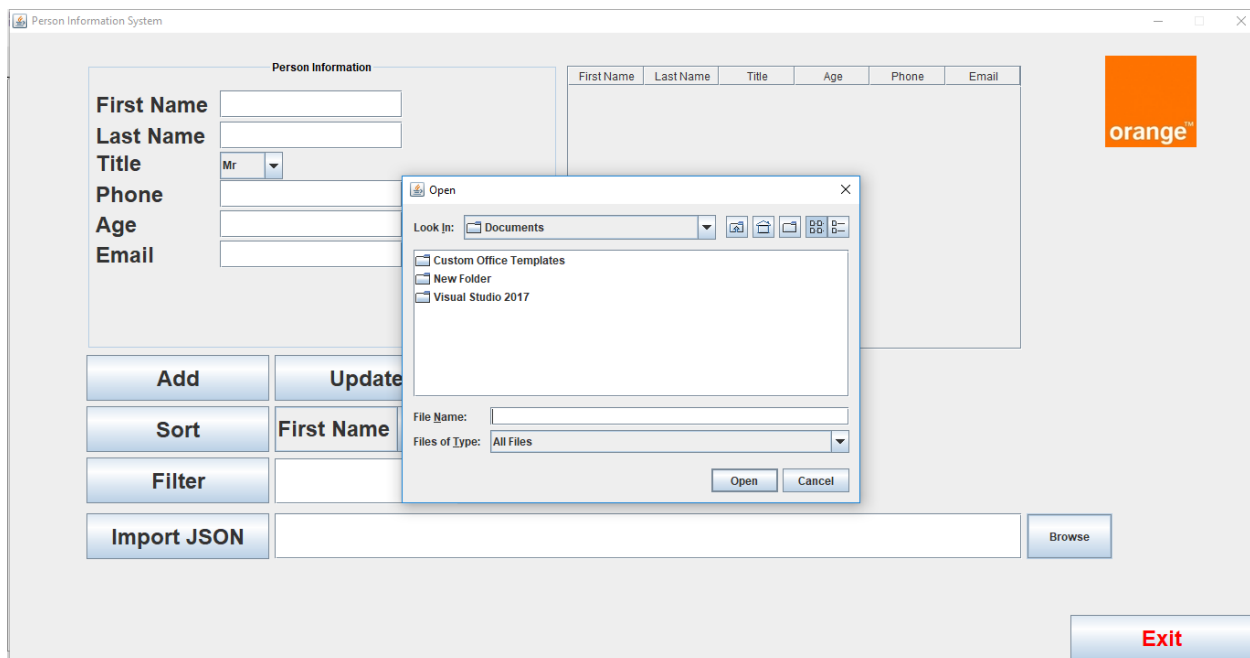
## 3.How to use the tool:

1. Tool interface:



2. Import the JSON file by two options:
   a. Simply by clicking the "Browse" button. (I've included myJSON.json file I've been testing on it you can browse for it in the project folder or any other type of JSON files as long as data is saved each record per a line.

b. Or by copying down the path of the file to the text box next to "Import JSON" button.



3. After you have imported the JSON file you will be able to see all the persons' data.

4. How to add:

5. How to update:
   a. Choose a row to update it and type in the new info then click update.

6. How to delete:
   b. Choose the row to be deleted then click delete button.
7. How to sort:
   a. First option to sort is throught clicking on the column header for ascending order, the next click will make it descending order and so on.



b. Second option is through choosing the column you want to sort from the combo box .

8. How to filter:
   c. Type in the text field next to filter the value and click "Filter" button.





9. "Reset" button is to clear all the data you've entered.

# 4.If we decided to change from JSON file to CSV format:

## a. Edit WriteToFile:
Instead of putting the records as (key,value) will be stored coma separated FirstName,LastName,Title,Phone,Age,Email.

```java
 92    private void WriteToFile(String Path)
 93    {
 94        BufferedReader br = null;
 95        JSONObject obj = new JSONObject ();
 96        try
 97        {
 98            obj.put("FirstName", FirstName);
 99            obj.put("LastName", LastName);
100            obj.put("Title", Title);
101            obj.put("Phone", Phone);
102            obj.put("Age", Age);
103            obj.put("Email", Email);
104        }
105        catch(Exception el)
106        {
107            el.printStackTrace();
108        }
109        try (FileWriter filewriter = new FileWriter(Path,true))
110        {
111            br = new BufferedReader(new FileReader(Path));
112            if (br.readLine() == null) {
113                filewriter.write(obj.toString());
```

## b. Edit how to Import the file:
Read the file and parse each line with the delimeter ','.

```java
542    public void actionPerformed(ActionEvent e) {
543        //C:/Users/badom/workspace/Person Information/myJSON.json
544        PersonList.clear();
545        List<String> RepeatedPersonList = new ArrayList<String>();
546        String JSONPathfile;
547        int RepeatedRows=0;
548        JSONPathfile= JSONPath.getText();
549        DefaultTableModel model = (DefaultTableModel)table.getModel();
550        model.setRowCount(0);
551        String sCurrentLine;
552        JSONParser parser = new JSONParser();
553        BufferedReader br = null;
554        try {
555            br = new BufferedReader(new FileReader(JSONPathfile));
556            while ((sCurrentLine = br.readLine()) != null)
557            {
558                Object obj = parser.parse(sCurrentLine);
559                JSONObject jsonobj = (JSONObject) obj;
560                FirstName= (String) jsonobj.get("FirstName");
561                LastName= (String) jsonobj.get("LastName");
562                Title =(String) jsonobj.get("Title");
563                Age=(String) jsonobj.get("Age");
564                Phone= (String) jsonobj.get("Phone");
565                Email=(String) jsonobj.get("Email");
566                if(!PersonList.contains(FirstName+LastName+Title+Age+Phone+Email))
567                {
568                    PersonList.add(FirstName+LastName+Title+Age+Phone+Email);
569                    model.addRow(new Object[]
570
```

## c. Edit Delete:

First search for the record containing the values to delete then erase
the line from the file.

```java
210        ;
211⊖    private void Delete(String FName,String LName,String TitleArg,String AgeArg,String PhoneArg,String EmailArg,String PathToFile)
212    {
213        JSONParser parser = new JSONParser();
214        BufferedReader br = null;
215        try {
216            int i=1;
217            String sCurrentLine;
218            br = new BufferedReader(new FileReader(PathToFile));
219            while ((sCurrentLine = br.readLine()) != null)
220            {
221                Object obj = parser.parse(sCurrentLine);
222                JSONObject jsonobj = (JSONObject) obj;
223                if(FName.contains((String)jsonobj.get("FirstName"))&&LName.contains((String)jsonobj.get("LastName"))
224                        &&TitleArg.contains((String)jsonobj.get("Title"))&&AgeArg.contains((String)jsonobj.get("Age"))
225                        &&PhoneArg.contains((String)jsonobj.get("Phone"))&&EmailArg.contains((String)jsonobj.get("Email")))
226                {
227                    jsonobj.remove("FirstName", (String)jsonobj.get("FirstName"));
228                    jsonobj.remove("LastName",(String)jsonobj.get("LastName"));
229                    jsonobj.remove("Title",(String)jsonobj.get("Title"));
230                    jsonobj.remove("Age",(String)jsonobj.get("Age"));
231                    jsonobj.remove("Phone",(String)jsonobj.get("Phone"));
232                    jsonobj.remove("Email",(String)jsonobj.get("Email"));
233                }
234                else
235                {
236                    WriteToNewFile(jsonobj,PathToFile,i);
237                    i++;
```