```java
package lockedmeapp;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.PrintWriter;
import java.nio.file.Paths;
import java.util.Arrays;
import java.util.Scanner;


//
https://github.com/ProgrammedPeinado/PracticeJava/blob/cd8e6f74265e1b614b1c580f7a1f8c478b07f711/Simplilearn/Final%20Projects/Implement%20OOPS%20using%20JAVA%20with%20Data%20Structures%20and%20Beyond/src/lockedmeapp/LockedMe.java
// By Hector Alarcon
public class LockedMe
{
    //Class variable declaration; most of these are resources we reuse multiple times throughout the program.
    private static Scanner input = new Scanner(System.in);
    private static String sel;
    private static String path = Paths.get("").toAbsolutePath().toString();
    private static final String USER_STORAGE = "\\Implement OOPS using JAVA with Data Structures and Beyond\\src\\ApplicationStorage";
    private static File storage = new File(path+USER_STORAGE);
    private static File[] dir = storage.listFiles();

    public static void main(String[] args)
    {
        //Kept main method clean and only handeling the exceptions form the screen methods.
        try
        {
            mainScreen();
        }
        catch (FileNotFoundException e)
        {
            System.out.println("File was not found in directory.");
        }
        catch (IOException e)
        {
            System.out.println("Something went wrong on the file directory.");
            System.out.println(e.getCause());
        }
        catch (Exception e)
        {
            System.out.println("Something went wrong.");
            System.out.println(e.getCause());
        }
        finally
        {
            System.out.println("Thank you for using the LockedMe App desgined by Hector Alarcon.");
            //Closing the scanner object and exiting the application.
            input.close();
            System.exit(0);
        }
    }

    private static void mainScreen() throws Exception
    {
        //Mainscreen user interaction information and welcome message.
        System.out.print("Welcome to the LockedMe app\ndesigned by Hector Alarcon\n");

        System.out.println("============================================================");
        System.out.println("Please select among the following options by typing the corresponding number:");
```

```
60            System.out.println("1. Inspect current directory.");
61            System.out.println("2. File handeling.");
62            System.out.println("3. Exit application.");
63
              System.out.print("============================================================
              ==============\n");
64            sel = input.nextLine();
65            System.out.println();
66            updateList();
67            Arrays.parallelSort(dir); //Updating the stored files if any had been modified.
68
69            //Sentinel value logic, making sure user inputs a number corresponding to the
              options.
70            while(!sel.startsWith("1") && !sel.startsWith("2") && !sel.startsWith("3"))
71            {
72                System.out.println("INVALID INPUT");
73
                  System.out.println("============================================================
                  ====================");
74                System.out.println("Please select among the following options by typing the
                  corresponding number:");
75                System.out.println("1. Inspect current directory.");
76                System.out.println("2. File handeling.");
77                System.out.println("3. Exit application.");
78
                  System.out.print("============================================================
                  ==================\n");
79                sel = input.nextLine();
80                System.out.println();
81            }
82
83            //Mainscreen options
84            if(sel.startsWith("1"))
85            {
86                firstOption();
87            }
88            else if(sel.startsWith("2"))
89            {
90                secondOption();
91            }
92            else if(sel.startsWith("3"))
93            {
94                System.out.println("Exiting..");
95            }
96        }
97
98        /**
99         * First option for the locker app, displays a list of the current stored files in
           Application Storage.
100        * @throws Exception
101        */
102       private static void firstOption() throws Exception
103       {
104           //If directory is not empty, display the file names, otherwise let the user
              known its empty.
105           if(dir != null)
106           {
107               System.out.println("You have the following files stored:");
108               for(File f : dir)
109                   System.out.println(f.getName());
110           }
111           else
112           {
113               System.out.println("The current repository is empty.\n\n");
114           }
115           System.out.print("\n");
116           mainScreen();
117       }
118
```

```
119         /**
120          * Second option for the locker app, adds a file to the directory as well as adding
             whatever content the user wishes to add.
121          * @throws IOException
122          * @throws Exception
123          */
124         private static void secondOption() throws IOException, Exception
125         {
126             //Second option user interaction information.
127 
                System.out.println("================================================================
                ================");
128             System.out.println("Please select among the following options by typing the
                corresponding number:");
129             System.out.println("1. Add a file to existing directory.");
130             System.out.println("2. Delete a file from existing directory.");
131             System.out.println("3. Search for a file from existing directory.");
132             System.out.println("4. Return to main menu.");
133 
                System.out.println("================================================================
                ================\n");
134             sel = input.nextLine();
135             System.out.println();
136 
137             //Sentinel value logic, making sure user inputs a number corresponding to the
                options.
138             while(sel.charAt(0) != '1' && sel.charAt(0) != '2' && sel.charAt(0) != '3' &&
                sel.charAt(0) != '4')
139             {
140                 System.out.println("INVALID INPUT");
141 
                    System.out.println("================================================================
                    ====================");
142                 System.out.println("Please select among the following options by typing the
                    corresponding number:");
143                 System.out.println("1. Add a file to existing directory.");
144                 System.out.println("2. Delete a file from existing directory.");
145                 System.out.println("3. Search for a file from existing directory.");
146                 System.out.println("4. Return to main menu.");
147 
                    System.out.println("================================================================
                    ====================\n");
148                 sel = input.nextLine();
149                 System.out.println();
150             }
151 
152             switch(sel.charAt(0))
153             {
154                 case('1'):
155                 {
156                     boolean finished = false;
157 
158                     System.out.println("Please type the name of the new file:\n");
159                     sel = input.nextLine();
160                     System.out.println();
161                     File upload = new File(storage.getAbsoluteFile()+"\\"+sel.toLowerCase());
162                     PrintWriter pw = new PrintWriter(upload.getAbsoluteFile());
163 
164                     System.out.println("Proceeds to type in the content of the file:");
165 
                        System.out.println("================================================
                        ========================");
166                     while(!finished)
167                     {
168                         pw.println(input.nextLine());
169 
170                         System.out.println("If you are done, please type "+"Exit"+"
                        otherwise, press Enter.");
171                         if(input.nextLine().contentEquals("Exit"))
```

```
172                     {
173                         finished = true;
174                     }
175                 }
176 
                    System.out.println("================================================
                    ========================\n");
177 
178 
179 
180                 //Giving confirmation to the user about whether or not the operation
                    was successful.
181                 System.out.println("File created successfully.\n");
182 
183                 //Updating the list, re-sorting, and closing the writer.Back to menu.
184                 pw.close();
185                 updateList();
186                 Arrays.parallelSort(dir);
187                 secondOption();
188                 break;
189             }
190             case('2'):
191             {
192                 System.out.println("Please type the name of the file you want to
                    delete:");
193 
                    System.out.println("================================================
                    ========================\n");
194                 sel = input.nextLine();
195                 System.out.println();
196                 boolean deleted = false;
197                 int i=1;
198 
199 
200                 for(File f: dir)
201                 {
202                     if(f.getName().equals(sel))
203                     {
204                         f.delete();
205                         deleted = true;
206                         System.out.println("File deleted successfully.\n");
207                     }
208                     else if(i == dir.length-1 && deleted == false)
209                     {
210                         System.out.println("File not found.\n");
211                     }
212                     i++;
213                 }
214                 //No need to sort after elimination since the rest of the files will be
                    sorted already
215                 //Updating the list. Back to menu.
216                 updateList();
217                 secondOption();
218                 break;
219             }
220             case('3'):
221             {
222                 System.out.println("Please type the name of the file you are looking
                    for:");
223 
                    System.out.println("================================================
                    ========================\n");
224                 sel = input.nextLine();
225                 System.out.println();
226                 boolean found = false;
227                 int i=0;
228 
229                 for(File f: dir)
230                 {
```

```
231                     if(f.getName().equals(sel))
232                     {
233                         found = true;
234                         System.out.println(dir[i].getName()+" has been found in
                            position "+i+" of the directory.\n");
235                     }
236                     else if(i == dir.length-1 && found == false)
237                     {
238                         System.out.println("File not found.\n");
239                     }
240                     i++;
241                 }
242 
243                 //Back to menu
244                 secondOption();
245                 break;
246             }
247             case('4'):
248             {
249                 //Back to mainscreen
250                 mainScreen();
251                 break;
252             }
253         }
254     }
255 
256     private static void updateList()
257     {
258         //Making sure the directory file list is always updated
259         dir = storage.listFiles();
260     }
261 }
262 
```