```java
package main.managers;

import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import javafx.stage.FileChooser;
import main.ClientSide;
import main.controllers.AlertHelper;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.stage.Stage;
import tale.Shorty;
import java.io.*;
import java.net.InetAddress;
import java.net.Socket;
import java.time.LocalDateTime;
import java.util.*;

public class Commander {

    private static Commander instance;
    private String email;
    private String password;
    private int ID;
    private Gson parser = new Gson();
    private List<Shorty> incomingList;
    private Stage mainWindow;

    public static Commander getInstance() {
        if (instance == null) {
            instance = new Commander();
        }
        return instance;
    }

    public boolean login(String email, String password) {
        this.ID = Integer.parseInt(requestToServer("login " + email + " " + password));
        try {
            if (this.ID != 0) {
                this.email = email;
                this.password = password;
                this.mainWindow = new Stage();
                this.mainWindow.setTitle("Main window");
                this.mainWindow.setResizable(false);
                FXMLLoader loader = new FXMLLoader();
                loader.setLocation(getClass().getResource("/main/resources/fxml/main_form
.fxml"));
                loader.setResources(ResourceBundle.getBundle("main.resources.Locale",
Locale.getDefault()));
                this.mainWindow.setScene(new Scene(loader.load()));
                this.mainWindow.show();
                return true;
            } else new AlertHelper(Alert.AlertType.ERROR, "ERROR", ClientSide.
languageResource.getString("alert.loginErrorMessage")).show();
            return false;
        } catch (IOException e) {
            e.printStackTrace(); //Никогда эта херня не выскочит, так как ресурсы FXML
внутри jar-ника, но JAVAAAAAA!!!
            return false;
        }
    }

    public void importing() {
        FileChooser fileChooser = new FileChooser();
        fileChooser.setTitle("Open json file");
        fileChooser.getExtensionFilters().add(new FileChooser.ExtensionFilter("JSON files
", "*.json"));
        File localCollectionFile = fileChooser.showOpenDialog(mainWindow);
        try (BufferedReader inputStreamReader = new BufferedReader(new FileReader(
localCollectionFile))) {
            String nextLine;
            StringBuilder result = new StringBuilder();
            while ((nextLine = inputStreamReader.readLine()) != null) result.append(
```

```
 67 nextLine);
 68             new AlertHelper(Alert.AlertType.INFORMATION, "ANSWER", requestToServer(email
     + " " + password + " " + "import " + result.toString())).show();
 69         } catch (IOException e) {
 70             new AlertHelper(Alert.AlertType.INFORMATION, "ANSWER", e.getLocalizedMessage
    ()).show();
 71         }
 72     }
 73
 74     public void removeLast() {
 75         new AlertHelper(Alert.AlertType.INFORMATION, "ANSWER", requestToServer(email +
    " " + password + " " + "remove_last")).show();
 76     }
 77
 78     public void clear() {
 79         new AlertHelper(Alert.AlertType.INFORMATION, "ANSWER", requestToServer(email +
    " " + password + " " + "clear")).show();
 80     }
 81
 82     public void add(String name, double x, double y) {
 83         Shorty s = new Shorty(name, x, y, LocalDateTime.now(), ID);
 84         new AlertHelper(Alert.AlertType.INFORMATION, "ANSWER", requestToServer(email +
    " " + password + " " + "add " + parser.toJson(s))).show();
 85     }
 86
 87     public void addIfMin(String name, double x, double y) {
 88         Shorty s = new Shorty(name, x, y, LocalDateTime.now(), ID);
 89         new AlertHelper(Alert.AlertType.INFORMATION, "ANSWER", requestToServer(email +
    " " + password + " " + "add_if_min " + parser.toJson(s))).show();
 90     }
 91
 92     public void register(String login, String password) {
 93         new AlertHelper(Alert.AlertType.INFORMATION, "ANSWER", requestToServer("register
     " + login + " " + password)).show();
 94     }
 95
 96
 97     public void deleteAccount(String login, String password) {
 98         new AlertHelper(Alert.AlertType.INFORMATION, "ANSWER", requestToServer("
    delete_account " + login + " " + password)).show();
 99     }
100
101     public List<Shorty> load() {
102         String answer = requestToServer(email + " " + password + " show");
103         incomingList = parser.fromJson(answer, new TypeToken<ArrayList<Shorty>>(){}.
    getType());
104
105         return incomingList;
106     }
107
108     public void remove(Shorty s) {
109         new AlertHelper(Alert.AlertType.INFORMATION, "ANSWER", requestToServer(email +
    " " + password + " " + "remove " + parser.toJson(s))).show();
110     }
111
112     public void save(List<Shorty> list) {
113         requestToServer(email + " " + password + " " + "update " + parser.toJson(list));
114         new AlertHelper(Alert.AlertType.INFORMATION, "ANSWER", requestToServer(email +
    " " + password + " " + "save")).show();
115     }
116
117     private String requestToServer(String request) {
118         try (Socket outcoming = new Socket(InetAddress.getLocalHost(), 8800)) {
119             outcoming.setSoTimeout(5000);
120             try (ObjectOutputStream toServer = new ObjectOutputStream(outcoming.
    getOutputStream());
121                  ObjectInputStream fromServer = new ObjectInputStream(outcoming.
    getInputStream())) {
122                 toServer.writeObject(request);
123                 return fromServer.readObject().toString();
124             }
125         } catch (IOException | ClassNotFoundException e) {
126             return e.getLocalizedMessage();
```

```
127         }
128     }
129
130     public int getID() {
131         return ID;
132     }
133
134     public String getEmail() {
135         return email;
136     }
137 }
```

File - V:\ITMO\1 course\programming\lab8-15.06.19\ClientGUI\src\main\managers\FieldsChecker.java

```
1 package main.managers;
2
3 public interface FieldsChecker {
4
5     boolean dataIsEntered();
6
7 }
8
```