File - V:\ITMO\1 course\programming\lab8-15.06.19\ClientGUI\src\client\controllers\AlertHelper.java

```java
package client.controllers;

import javafx.scene.control.Alert;

public class AlertHelper {

    private Alert alert;

    public AlertHelper(Alert.AlertType alertType, String title, String message) {
        alert = new Alert(alertType);
        alert.setTitle(title);
        alert.setHeaderText(null);
        alert.setContentText(message);
    }

    public void show() {
        alert.show();
    }
}
```

```java
package client.controllers;

import javafx.collections.FXCollections;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.Group;
import javafx.scene.control.*;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.control.cell.TextFieldTableCell;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.util.converter.DoubleStringConverter;
import client.ClientSide;
import client.managers.Commander;
import client.managers.FieldsChecker;
import tale.Shorty;
import java.net.URL;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.time.LocalDateTime;
import java.util.ArrayList;
import java.util.List;
import java.util.ResourceBundle;

public class MainFormController implements FieldsChecker, Initializable {

    @FXML
    private Label statusLabel;

    @FXML
    private TableView<Shorty> shortysTable;

    @FXML
    private TextField nameField;

    @FXML
    private TextField xCoordField;

    @FXML
    private TextField yCoordField;

    @FXML
    private Tab mapTab;

    private Group group = new Group();

    @Override
    public void initialize(URL location, ResourceBundle resources) {
        statusLabel.setText(Commander.getInstance().getEmail() + ", ID - " + Commander.
getInstance().getID());
        initTable();
        handleUpdateButtonAction();
    }

    @FXML
    private void initTable() {
        shortysTable.setEditable(true);
        TableColumn<Shorty, String> shortysNameCol = new TableColumn("Name");
        shortysNameCol.setCellValueFactory(new PropertyValueFactory<>("name"));
        shortysNameCol.setCellFactory(TextFieldTableCell.forTableColumn());
        shortysNameCol.setOnEditCommit(event -> {
            if (Commander.getInstance().getID() == event.getTableView().getItems().get(
event.getTablePosition().getRow()).getMasterID())
                event.getTableView().getItems().get(event.getTablePosition().getRow()).
setName(event.getNewValue());
            else dataEditProhibition();
        });
        TableColumn<Shorty, Double> coordXCol = new TableColumn("X");
        coordXCol.setCellValueFactory(new PropertyValueFactory<>("X"));
        coordXCol.setCellFactory(TextFieldTableCell.forTableColumn(new
DoubleStringConverter()));
```

```
70          coordXCol.setOnEditCommit(event -> {
71              if (Commander.getInstance().getID() == event.getTableView().getItems().get(
    event.getTablePosition().getRow()).getMasterID())
72                  event.getTableView().getItems().get(event.getTablePosition().getRow()).
    setX(event.getNewValue());
73              else dataEditProhibition();
74          });
75          TableColumn<Shorty, Double> coordYCol = new TableColumn<>("Y");
76          coordYCol.setCellValueFactory(new PropertyValueFactory<>("y"));
77          coordYCol.setCellFactory(TextFieldTableCell.forTableColumn(new
    DoubleStringConverter()));
78          coordYCol.setOnEditCommit(event -> {
79              if (Commander.getInstance().getID() == event.getTableView().getItems().get(
    event.getTablePosition().getRow()).getMasterID())
80                  event.getTableView().getItems().get(event.getTablePosition().getRow()).
    setY(event.getNewValue());
81              else dataEditProhibition();
82          });
83          TableColumn<Shorty, LocalDateTime> birthdayCol = new TableColumn("Birthday");
84          birthdayCol.setCellValueFactory(new PropertyValueFactory<>("birthday"));
85          TableColumn masterIDCol = new TableColumn("Master ID");
86          masterIDCol.setCellValueFactory(new PropertyValueFactory<>("masterID"));
87          shortysTable.getColumns().addAll(shortysNameCol, coordXCol, coordYCol,
    birthdayCol, masterIDCol);
88          fillTable();
89      }
90
91      @FXML
92      private void fillMap() {
93          group.getChildren().clear();
94          List<Shorty> shorties = Commander.getInstance().load();
95          for (Shorty s: shorties) {
96              Circle circle = new Circle(s.getX(), s.getY(), 20, Color.valueOf(colorByID(s
    .getMasterID())));
97              circle.setOnMouseClicked(event -> new AlertHelper(Alert.AlertType.
    INFORMATION, "INFO", s.toString()).show());
98             group.getChildren().add(circle);
99          }
100         mapTab.setContent(group);
101     }
102
103     @FXML
104     private void fillTable() {
105         shortysTable.setItems(FXCollections.observableArrayList(Commander.getInstance().
    load()));
106         shortysTable.refresh();
107     }
108
109     @FXML
110     private void handleAddButtonAction() {
111         try {
112             if (dataIsEntered()) {
113                 Commander.getInstance().add(nameField.getText(), Double.parseDouble(
    xCoordField.getText()),
114                         Double.parseDouble(yCoordField.getText()));
115                 handleUpdateButtonAction();
116             }
117         } catch (NumberFormatException e) {
118             notNumberAlert(e);
119         }
120     }
121
122     @FXML
123     private void handleAddIfMinButtonAction() {
124         try {
125             if (dataIsEntered()) {
126                 Commander.getInstance().addIfMin(nameField.getText(), Double.parseDouble
    (xCoordField.getText()),
127                         Double.parseDouble(yCoordField.getText()));
128                 handleUpdateButtonAction();
129             }
130         } catch (NumberFormatException e) {
131             notNumberAlert(e);
```

```
132         }
133     }
134 
135     @FXML
136     private void notNumberAlert(Exception e) {
137         new AlertHelper(Alert.AlertType.ERROR, e.getLocalizedMessage(), ClientSide.
    languageResource.
138                 getString("alert.CoordsIllegalMessage")).show();
139     }
140 
141     @FXML
142     private void handleSaveButtonAction() {
143         Commander.getInstance().save(new ArrayList<>(shortysTable.getItems()));
144         handleUpdateButtonAction();
145     }
146 
147     @FXML
148     private void handleRemoveLastButtonAction() {
149         Commander.getInstance().removeLast();
150         handleUpdateButtonAction();
151     }
152 
153     @FXML
154     private void handleUpdateButtonAction() {
155         fillTable();
156         fillMap();
157     }
158 
159     @FXML
160     private void handleDeleteButtonAction() {
161         Commander.getInstance().remove(shortysTable.getSelectionModel().getSelectedItem(
    ));
162         handleUpdateButtonAction();
163     }
164 
165     @FXML
166     private void handleClearButtonAction() {
167         Commander.getInstance().clear();
168         handleUpdateButtonAction();
169     }
170 
171     @FXML
172     private void handleImportButtonAction() {
173         Commander.getInstance().importing();
174         handleUpdateButtonAction();
175     }
176 
177     public void dataEditProhibition() {
178         new AlertHelper(Alert.AlertType.ERROR, "ACCESS_DENIED", ClientSide.
    languageResource.
179                 getString("alert.dataEditErrorMessage")).show();
180         shortysTable.refresh();
181     }
182 
183     private String colorByID(int ID) {
184         try {
185             MessageDigest mDigest = MessageDigest.getInstance("MD5");
186             byte[] hash = mDigest.digest(Integer.toString(ID).getBytes());
187             StringBuilder hex = new StringBuilder();
188             for (byte b : hash) hex.append(Integer.toString((b & 0xff) + 0x100, 16).
    substring(1));
189             return hex.toString().substring(0, 6);
190         } catch (NoSuchAlgorithmException e) {
191             e.printStackTrace();
192             return "#babbbc";
193         }
194     }
195 
196     @FXML
197     @Override
198     public boolean dataIsEntered() {
199         if (nameField.getText().isEmpty() || xCoordField.getText().isEmpty() ||
    yCoordField.getText().isEmpty()) {
```

```
200             new AlertHelper(Alert.AlertType.ERROR, "ERROR", ClientSide.languageResource.
    getString("alert.dataInputErrorMessage")).show();
201             return false;
202         } else return true;
203     }
204 }
```

```java
package client.controllers;

import javafx.fxml.FXML;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.stage.Stage;
import client.ClientSide;
import client.managers.Commander;
import client.managers.FieldsChecker;

public class LoginFormController implements FieldsChecker {

    @FXML
    private TextField loginField;

    @FXML
    private PasswordField passwordField;

    @FXML
    private Button loginButton;

    @FXML
    protected void handleLoginButtonAction() {
        if (dataIsEntered() && Commander.getInstance().login(loginField.getText(), passwordField.getText())) {
            Stage currentWindow = (Stage) loginButton.getScene().getWindow();
            currentWindow.close();
        }
    }

    @FXML
    protected void handleRegisterButtonAction() {
        if (dataIsEntered()) Commander.getInstance().register(loginField.getText(), passwordField.getText());
    }

    @FXML
    protected void handleDeleteButtonAction() {
        if (dataIsEntered()) Commander.getInstance().deleteAccount(loginField.getText(), passwordField.getText());
    }

    @Override
    public boolean dataIsEntered() {
        if (loginField.getText().isEmpty() || passwordField.getText().isEmpty()) {
            new AlertHelper(Alert.AlertType.ERROR, ClientSide.languageResource.getString("alert.emptyField"),
                    ClientSide.languageResource.getString("alert.emptyFieldMessage")).show();
            return false;
        }
        return true;
    }
}
```