ДОДАТОК Б КОД КОНТРОЛЕРІВ, ТОЧКУ ВХОДУ КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ ТА MODUL-INFO

Controller/OrderManagementController.java

package org.vitapasser.photocopypoint.Controller;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.scene.Node;

import javafx.scene.Scene;

import javafx.stage.Stage;

import org.vitapasser.photocopypoint.Controller.CreatingOrder.Controller;

import org.vitapasser.photocopypoint.MainApplication;

import org.vitapasser.photocopypoint.Model.Register;

import java.io.IOException;

import java.util.Objects;

public class OrderManagementController {

private Register register;

@FXML

protected void onStartCreateOrderButtonClick(ActionEvent event) throws IOException {

register.newOrder();

FXMLLoader FXMLLoader = new FXMLLoader(Objects.requireNonNull(

MainApplication.class.getResource("creating-order.fxml")));

Scene scene = new Scene(FXMLLoader.load());

Controller controller = FXMLLoader.getController();

controller.putData(register);

Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();

stage.setScene(scene);

stage.show();

}

public void initialize() {}

public void putData(Register register) {

this.register = register;

}

}

Controller/PaymentController.java

package org.vitapasser.photocopypoint.Controller;

import javafx.application.Platform;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.scene.Node;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

import org.vitapasser.photocopypoint.Controller.CreatingOrder.Controller;

import org.vitapasser.photocopypoint.MainApplication;

import org.vitapasser.photocopypoint.Model.Money;

import org.vitapasser.photocopypoint.Model.Register;

import org.vitapasser.photocopypoint.Model.Ticket;

import java.io.IOException;

import java.util.List;

import java.util.Objects;

public class PaymentController {

private Register register;

String nameClient;

String phoneNumberClient;

ObservableList<Ticket> ticketsList = FXCollections.observableArrayList();

@FXML

TableView<Ticket> listOfTicketsTableView;

@FXML

private TableColumn<Ticket, String> fullNameColumn;

@FXML

private TableColumn<Ticket, Long> idTicketColumn;

@FXML

private TableColumn<Ticket, Boolean> isReadyColumn;

@FXML

private TableColumn<Ticket, String> namesTypeServiceColumn;

@FXML

private TableColumn<Ticket, String> phoneNumberColumn;

@FXML

Label resultOddMoney;

@FXML

Label resultPrice;

@FXML

Label payLabel;

@FXML

TextField countClientPayTextField;

@FXML

VBox oddPayVBox;

@FXML

Button acceptButton;

@FXML

Button cancelButton;

@FXML

private void onPaymentButtonClick(ActionEvent event) {

if (countClientPayTextField.getText().isEmpty()) {

payLabel.setText("До сплати| Введіть суму оплаченою клієнтом");

return;

}

try {

if (Double.parseDouble(countClientPayTextField.getText()) < register.getPrice().value()){

payLabel.setText("До сплати| Не вистачає\n грошей для оплати!");

return;

} else {

payLabel.setText("До сплати");

}

oddPayVBox.setDisable(false);

resultOddMoney.setText(register.makePayment(

new Money(Double.parseDouble(countClientPayTextField.getText()), register.getPrice().unit()),

nameClient,

phoneNumberClient).toString());

cancelButton.setDisable(true);

acceptButton.setDisable(true);

changeTable();

} catch (NumberFormatException e) {

payLabel.setText("До сплати| Використовуйте\n тільки числа та точку!");

}

}

@FXML

private void onOrderManagementButtonClick(ActionEvent event) throws IOException {

FXMLLoader FXMLLoader = new FXMLLoader(Objects.requireNonNull(

MainApplication.class.getResource("order-management.fxml")));

Scene scene = new Scene(FXMLLoader.load());

OrderManagementController controller = FXMLLoader.getController();

controller.putData(register);

Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();

stage.setScene(scene);

stage.show();

}

@FXML

protected void onCancelButtonClick(ActionEvent event) throws IOException {

FXMLLoader FXMLLoader = new FXMLLoader(Objects.requireNonNull(

MainApplication.class.getResource("creating-order.fxml")));

Scene scene = new Scene(FXMLLoader.load());

Controller controller = FXMLLoader.getController();

controller.putData(register);

Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();

stage.setScene(scene);

stage.show();

}

private void changeTable() {

List<Ticket> tickets = register.getAllTickets();

ticketsList.clear();

ticketsList.addAll(tickets);

}

@FXML

private void initialize() {

Platform.runLater(() -> {

resultPrice.setText(register.getPrice().toString());

listOfTicketsTableView.setItems(ticketsList);

ticketsList.clear();

idTicketColumn.setCellValueFactory(new PropertyValueFactory<>("id"));

fullNameColumn.setCellValueFactory(new PropertyValueFactory<>("fullName"));

phoneNumberColumn.setCellValueFactory(new PropertyValueFactory<>("phoneNumber"));

isReadyColumn.setCellValueFactory(new PropertyValueFactory<>("isUsed"));

namesTypeServiceColumn.setCellValueFactory(new PropertyValueFactory<>("name"));

changeTable();

});

}

public void putData(Register register, String nameClient, String phoneNumberClient) {

this.register = register;

this.nameClient = nameClient;

this.phoneNumberClient = phoneNumberClient;

}

}

Controller/CreatingOrder/TypeView.java

package org.vitapasser.photocopypoint.Controller.CreatingOrder;

import org.vitapasser.photocopypoint.Model.Type;

import org.vitapasser.photocopypoint.Model.TypeItem;

import java.time.LocalTime;

public class TypeView {

long id;

String name;

String info;

LocalTime term;

Double countMoney;

String currencyMoney;

int count;

public int getCount() {

return count;

}

public void setCount(int count) {

this.count = count;

}

public TypeView (Type type){

this.id = type.id();

this.name = type.name();

this.info = type.info();

this.term = type.term().toLocalTime();

this.countMoney = type.money().value();

this.currencyMoney = type.money().unit();

this.count = 1;

}

public TypeView (TypeItem type){

this.id = type.type().id();

this.name = type.type().name();

this.info = type.type().info();

this.term = type.getTerm().toLocalTime();

this.countMoney = type.getMoney().value();

this.currencyMoney = type.getMoney().unit();

this.count = type.count();

}

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getInfo() {

return info;

}

public void setInfo(String info) {

this.info = info;

}

public LocalTime getTerm() {

return term;

}

public void setTerm(LocalTime term) {

this.term = term;

}

public Double getCountMoney() {

return countMoney;

}

public void setCountMoney(Double countMoney) {

this.countMoney = countMoney;

}

public String getCurrencyMoney() {

return currencyMoney;

}

public void setCurrencyMoney(String currencyMoney) {

this.currencyMoney = currencyMoney;

}

}

Controller/CreatingOrder/Controller.java

package org.vitapasser.photocopypoint.Controller.CreatingOrder;

import com.mysql.cj.util.StringUtils;

import javafx.application.Platform;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.scene.Node;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.input.KeyEvent;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

import org.vitapasser.photocopypoint.Controller.OrderManagementController;

import org.vitapasser.photocopypoint.Controller.PaymentController;

import org.vitapasser.photocopypoint.Exception.NotExistTypeException;

import org.vitapasser.photocopypoint.MainApplication;

import org.vitapasser.photocopypoint.Model.Register;

import org.vitapasser.photocopypoint.Model.Type;

import org.vitapasser.photocopypoint.Model.TypeItem;

import java.io.IOException;

import java.time.LocalTime;

import java.util.List;

import java.util.Objects;

public class Controller {

ObservableList<TypeView> typeViewsListOfSelectableServices = FXCollections.observableArrayList();

ObservableList<TypeView> typeViewsListSelectedServices = FXCollections.observableArrayList();

Register register;

@FXML

private TableColumn<TypeView, Double> countMoneyTypeColumn;

@FXML

private TableColumn<TypeView, Double> countMoneyTypeColumn1;

@FXML

private TableColumn<TypeView, String> currencyMoneyTypeColumn;

@FXML

private TableColumn<TypeView, String> currencyMoneyTypeColumn1;

@FXML

private TableColumn<TypeView, Long> idTypeColumn;

@FXML

private TableColumn<TypeView, Long> idTypeColumn1;

@FXML

private TableColumn<TypeView, String> infoTypeColumn;

@FXML

private TableColumn<TypeView, String> infoTypeColumn1;

@FXML

private TableColumn<TypeView, String> nameTypeColumn;

@FXML

private TableColumn<TypeView, String> nameTypeColumn1;

@FXML

private TableColumn<TypeView, LocalTime> termTypeColumn;

@FXML

private TableColumn<TypeView, String> termTypeColumn1;

@FXML

protected TableColumn<TypeView, Integer> countTypeColumn1;

@FXML

protected TextField nameTypeService;

@FXML

protected TextField countTypeService;

@FXML

protected VBox clientVBox;

@FXML

protected TextField nameClient;

@FXML

protected TextField phoneNumberClient;

@FXML

protected Label resultPrice;

@FXML

protected Label resultTerm;

@FXML

protected Label nameNameTypeService;

@FXML

protected TableView<TypeView> listSelectedServices;

@FXML

protected TableView<TypeView> listOfSelectableServices;

@FXML

protected Button paymentButton;

@FXML

protected Button addTypeServiceButton;

@FXML

protected Label countTypeServiceLabel;

@FXML

protected Label phoneNumberLabel;

@FXML

protected void onChangeNameTypeServiceTextField(KeyEvent event) {

changeTable(typeViewsListOfSelectableServices);

}

@FXML

protected void onGetNameTypeServiceTableView(MouseEvent event) {

TypeView typeView = listOfSelectableServices.getSelectionModel().getSelectedItem();

if (typeView == null) return;

nameTypeService.setText(typeView.name);

}

@FXML

protected void onWriteCheckTextField(KeyEvent event) {

countTypeServiceLabel.setText("Кількість послуги");

if (!StringUtils.isStrictlyNumeric(countTypeService.getText())) {

addTypeServiceButton.setDisable(true);

countTypeServiceLabel.setText("Кількість послуги| Введіть тільки числове значення!");

} else {

addTypeServiceButton.setDisable(false);

}

}

private void changeTable(ObservableList<TypeView> list) {

List<Type> types;

nameNameTypeService.setText("Назва послуги");

addTypeServiceButton.setDisable(false);

try {

types = register.getTypes(nameTypeService.getText());

list.clear();

types.forEach(type -> list.add(new TypeView(type)));

} catch (NotExistTypeException e) {

String string = "Назва послуги| " + e.getMessage();

nameNameTypeService.setText(string);

addTypeServiceButton.setDisable(true);

}

}

private void initLoadTackedTable() {

List<TypeItem> types = register.getOrderTypes();

typeViewsListSelectedServices.clear();

types.forEach(type -> typeViewsListSelectedServices.add(new TypeView(type)));

}

@FXML

protected void onAddTypeServiceButtonClick(ActionEvent event) {

if (nameTypeService.getText().isEmpty() || countTypeService.getText().isEmpty()) {

return;

}

List<TypeItem> typeItem = register.addType(nameTypeService.getText(), Integer.parseInt(countTypeService.getText()));

typeViewsListSelectedServices.clear();

typeItem.forEach(typeItem1 -> typeViewsListSelectedServices.add(new TypeView(typeItem1)));

clientVBox.setDisable(false);

resultPrice.setText(register.getPrice().toString());

resultTerm.setText(register.getTerm().toString());

}

@FXML

protected void onCancelTypeServiceButtonClick(ActionEvent event) {

}

@FXML

protected void onWriteNamePhoneNumber(KeyEvent event) {

if (!phoneNumberClient.getText().matches("^(\\+\\d{1,3}\\s?)?1?\\-?\\.?\\s?\\(?\\d{3}\\)?[\\s.-]?\\d{3}[\\s.-]?\\d{4}$")){

phoneNumberLabel.setText("Номер телефону клієнта| Помилковий вираз!");

paymentButton.setDisable(true);

return;

}

phoneNumberLabel.setText("Номер телефону клієнта");

paymentButton.setDisable(false);

}

private boolean isEmptyNamePhoneNumberClient() {

return nameClient.getText().isEmpty() || phoneNumberClient.getText().isEmpty();

}

@FXML

protected void onPaymentButtonClick(ActionEvent event) throws IOException {

if (isEmptyNamePhoneNumberClient()) {

return;

}

FXMLLoader FXMLLoader = new FXMLLoader(Objects.requireNonNull(

MainApplication.class.getResource("payment.fxml")));

Scene scene = new Scene(FXMLLoader.load());

PaymentController controller = FXMLLoader.getController();

controller.putData(register, nameClient.getText(), phoneNumberClient.getText());

Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();

stage.setScene(scene);

stage.show();

}

@FXML

protected void onCancelButtonClick(ActionEvent event) throws IOException {

FXMLLoader FXMLLoader = new FXMLLoader(Objects.requireNonNull(

MainApplication.class.getResource("order-management.fxml")));

Scene scene = new Scene(FXMLLoader.load());

OrderManagementController controller = FXMLLoader.getController();

controller.putData(register);

Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();

stage.setScene(scene);

stage.show();

}

public void initialize() {

Platform.runLater(() -> {

listOfSelectableServices.setItems(typeViewsListOfSelectableServices);

listSelectedServices.setItems(typeViewsListSelectedServices);

typeViewsListOfSelectableServices.clear();

typeViewsListSelectedServices.clear();

idTypeColumn.setCellValueFactory(new PropertyValueFactory<>("id"));

idTypeColumn1.setCellValueFactory(new PropertyValueFactory<>("id"));

nameTypeColumn.setCellValueFactory(new PropertyValueFactory<>("name"));

nameTypeColumn1.setCellValueFactory(new PropertyValueFactory<>("name"));

infoTypeColumn.setCellValueFactory(new PropertyValueFactory<>("info"));

infoTypeColumn1.setCellValueFactory(new PropertyValueFactory<>("info"));

termTypeColumn.setCellValueFactory(new PropertyValueFactory<>("term"));

termTypeColumn1.setCellValueFactory(new PropertyValueFactory<>("term"));

countMoneyTypeColumn.setCellValueFactory(new PropertyValueFactory<>("countMoney"));

countMoneyTypeColumn1.setCellValueFactory(new PropertyValueFactory<>("countMoney"));

currencyMoneyTypeColumn.setCellValueFactory(new PropertyValueFactory<>("currencyMoney"));

currencyMoneyTypeColumn1.setCellValueFactory(new PropertyValueFactory<>("currencyMoney"));

countTypeColumn1.setCellValueFactory(new PropertyValueFactory<>("count"));

changeTable(typeViewsListOfSelectableServices);

initLoadTackedTable();

});

}

public void putData(Register register) {

this.register = register;

}

}

Creating-order.fxml

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.control.ScrollPane?>

<?import javafx.scene.control.Separator?>

<?import javafx.scene.control.TableColumn?>

<?import javafx.scene.control.TableView?>

<?import javafx.scene.control.TextField?>

<?import javafx.scene.layout.BorderPane?>

<?import javafx.scene.layout.ColumnConstraints?>

<?import javafx.scene.layout.GridPane?>

<?import javafx.scene.layout.HBox?>

<?import javafx.scene.layout.RowConstraints?>

<?import javafx.scene.layout.VBox?>

<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="800.0" prefWidth="1200.0" xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1" fx:controller="org.vitapasser.photocopypoint.Controller.CreatingOrder.Controller">

<right>

<BorderPane prefWidth="339.0" BorderPane.alignment="CENTER">

<center>

<BorderPane BorderPane.alignment="CENTER">

<center>

<VBox alignment="CENTER" prefHeight="323.0" prefWidth="200.0" spacing="10.0">

<children>

<VBox alignment="CENTER">

<children>

<VBox prefHeight="31.0" prefWidth="200.0">

<children>

<Label fx:id="nameNameTypeService" text="Назва послуги" />

<TextField fx:id="nameTypeService" onKeyTyped="#onChangeNameTypeServiceTextField" />

</children>

</VBox>

<VBox alignment="CENTER\_LEFT" prefHeight="39.0" prefWidth="200.0">

<children>

<Label fx:id="countTypeServiceLabel" text="Кількість послуги" />

<TextField fx:id="countTypeService" onKeyTyped="#onWriteCheckTextField" />

</children>

</VBox>

<HBox alignment="CENTER">

<VBox.margin>

<Insets top="10.0" />

</VBox.margin>

<children>

<Button fx:id="addTypeServiceButton" contentDisplay="BOTTOM" mnemonicParsing="false" onAction="#onAddTypeServiceButtonClick" text="Додати послугу" />

<Button contentDisplay="BOTTOM" disable="true" mnemonicParsing="false" onAction="#onCancelTypeServiceButtonClick" prefWidth="98.0" text="Відмінити" />

</children>

<opaqueInsets>

<Insets />

</opaqueInsets>

</HBox>

</children>

</VBox>

<VBox fx:id="clientVBox" disable="true" layoutX="10.0" layoutY="52.0" prefHeight="58.0" prefWidth="200.0">

<children>

<VBox layoutX="10.0" layoutY="10.0">

<children>

<Label text="Ім'я клієнта" />

<TextField fx:id="nameClient" onKeyTyped="#onWriteNamePhoneNumber" />

</children>

</VBox>

<VBox layoutX="10.0" layoutY="10.0">

<children>

<Label fx:id="phoneNumberLabel" text="Номер телефону клієнта" />

<TextField fx:id="phoneNumberClient" onKeyTyped="#onWriteNamePhoneNumber" />

</children>

</VBox>

</children>

</VBox>

<VBox layoutX="10.0" layoutY="459.0">

<children>

<Label text="Приблизний час вироблювання">

<font>

<Font size="21.0" />

</font>

</Label>

<Label fx:id="resultTerm" text="00:00">

<font>

<Font size="24.0" />

</font>

</Label>

</children>

</VBox>

<VBox>

<children>

<Label text="Ціна замовлення">

<font>

<Font size="21.0" />

</font>

</Label>

<Label fx:id="resultPrice" text="0.00 грн">

<font>

<Font size="24.0" />

</font>

</Label>

</children>

</VBox>

</children>

<padding>

<Insets top="10.0" />

</padding>

</VBox>

</center>

<padding>

<Insets left="10.0" right="10.0" />

</padding>

<bottom>

<VBox alignment="CENTER" spacing="10.0" BorderPane.alignment="CENTER">

<children>

<Button fx:id="paymentButton" alignment="CENTER" disable="true" mnemonicParsing="false" onAction="#onPaymentButtonClick" prefHeight="25.0" prefWidth="200.0" text="Оплата" />

<Button mnemonicParsing="false" onAction="#onCancelButtonClick" prefHeight="25.0" prefWidth="200.0" text="Відмінити створення замовлення" />

</children>

<BorderPane.margin>

<Insets bottom="10.0" />

</BorderPane.margin>

</VBox>

</bottom>

</BorderPane>

</center>

<left>

<Separator orientation="VERTICAL" BorderPane.alignment="CENTER" />

</left>

<BorderPane.margin>

<Insets />

</BorderPane.margin></BorderPane>

</right>

<center>

<GridPane BorderPane.alignment="CENTER">

<columnConstraints>

<ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />

</columnConstraints>

<rowConstraints>

<RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />

<RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />

</rowConstraints>

<children>

<BorderPane prefHeight="200.0" prefWidth="200.0">

<top>

<Label text="Список обираємих послуг" BorderPane.alignment="CENTER" />

</top>

<center>

<ScrollPane fitToHeight="true" fitToWidth="true" prefHeight="200.0" prefWidth="200.0" BorderPane.alignment="CENTER">

<content>

<TableView fx:id="listOfSelectableServices" onMouseClicked="#onGetNameTypeServiceTableView" prefHeight="400.0" prefWidth="736.0">

<columnResizePolicy>

<TableView fx:constant="CONSTRAINED\_RESIZE\_POLICY" />

</columnResizePolicy>

<columns>

<TableColumn fx:id="idTypeColumn" prefWidth="75.0" text="Ід" />

<TableColumn fx:id="nameTypeColumn" prefWidth="75.0" text="Назва" />

<TableColumn fx:id="infoTypeColumn" prefWidth="75.0" text="Додаткова інформація" />

<TableColumn fx:id="termTypeColumn" prefWidth="75.0" text="Термін" />

<TableColumn fx:id="countMoneyTypeColumn" prefWidth="75.0" text="Ціна" />

<TableColumn fx:id="currencyMoneyTypeColumn" prefWidth="75.0" text="Валюта" />

</columns>

</TableView>

</content>

</ScrollPane>

</center></BorderPane>

<BorderPane layoutX="10.0" layoutY="10.0" prefHeight="200.0" prefWidth="200.0" GridPane.rowIndex="1">

<top>

<Label text="Список обраних послуг" BorderPane.alignment="CENTER" />

</top>

<center>

<ScrollPane fitToHeight="true" fitToWidth="true" prefHeight="200.0" prefWidth="200.0" BorderPane.alignment="CENTER">

<content>

<TableView fx:id="listSelectedServices" prefHeight="200.0" prefWidth="200.0">

<columnResizePolicy>

<TableView fx:constant="CONSTRAINED\_RESIZE\_POLICY" />

</columnResizePolicy>

<columns>

<TableColumn fx:id="idTypeColumn1" prefWidth="75.0" text="Ід" />

<TableColumn fx:id="nameTypeColumn1" prefWidth="75.0" text="Назва" />

<TableColumn fx:id="infoTypeColumn1" prefWidth="75.0" text="Додаткова інформація" />

<TableColumn fx:id="termTypeColumn1" prefWidth="75.0" text="Термін" />

<TableColumn fx:id="countTypeColumn1" prefWidth="75.0" text="Кількість" />

<TableColumn fx:id="countMoneyTypeColumn1" prefWidth="75.0" text="Ціна" />

<TableColumn fx:id="currencyMoneyTypeColumn1" prefWidth="75.0" text="Валюта" />

</columns>

</TableView>

</content>

</ScrollPane>

</center>

</BorderPane>

</children>

</GridPane>

</center>

</BorderPane>

Order-management.fxml

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.Separator?>

<?import javafx.scene.layout.BorderPane?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="800.0" prefWidth="1200.0" xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1" fx:controller="org.vitapasser.photocopypoint.Controller.OrderManagementController">

<right>

<BorderPane prefWidth="339.0" BorderPane.alignment="CENTER">

<left>

<Separator orientation="VERTICAL" BorderPane.alignment="CENTER" />

</left>

<BorderPane.margin>

<Insets />

</BorderPane.margin>

<center>

<Button alignment="CENTER" mnemonicParsing="false" onAction="#onStartCreateOrderButtonClick" prefHeight="25.0" prefWidth="200.0" text="Створити нове замовлення" BorderPane.alignment="CENTER" />

</center></BorderPane>

</right>

</BorderPane>

Payment.fxml

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.control.ScrollPane?>

<?import javafx.scene.control.Separator?>

<?import javafx.scene.control.TableColumn?>

<?import javafx.scene.control.TableView?>

<?import javafx.scene.control.TextField?>

<?import javafx.scene.layout.BorderPane?>

<?import javafx.scene.layout.VBox?>

<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="800.0" prefWidth="1200.0" xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1" fx:controller="org.vitapasser.photocopypoint.Controller.PaymentController">

<right>

<BorderPane prefWidth="339.0" BorderPane.alignment="CENTER">

<center>

<VBox alignment="CENTER" spacing="30.0">

<children>

<VBox spacing="10.0">

<children>

<VBox alignment="CENTER\_LEFT">

<children>

<Label fx:id="payLabel" text="До сплати">

<font>

<Font size="21.0" />

</font>

</Label>

<Label fx:id="resultPrice" text="0.00 грн">

<font>

<Font size="24.0" />

</font>

</Label>

</children>

</VBox>

<VBox alignment="CENTER\_LEFT" prefHeight="39.0" prefWidth="200.0" spacing="5.0">

<children>

<Label text="Внесена оплата клієнтом" />

<TextField fx:id="countClientPayTextField" />

</children>

</VBox>

<Button fx:id="acceptButton" mnemonicParsing="false" onAction="#onPaymentButtonClick" text="Пітвердити оплату" />

<Button fx:id="cancelButton" mnemonicParsing="false" onAction="#onCancelButtonClick" text="Відменити оплату" />

</children>

</VBox>

<VBox fx:id="oddPayVBox" disable="true" layoutX="10.0" layoutY="92.0" spacing="10.0">

<children>

<VBox alignment="CENTER\_LEFT" prefHeight="10.0" prefWidth="193.0">

<children>

<Label text="Решта">

<font>

<Font size="21.0" />

</font>

</Label>

<Label fx:id="resultOddMoney" text="0.00 грн">

<font>

<Font size="24.0" />

</font>

</Label>

</children>

</VBox>

<Button mnemonicParsing="false" onAction="#onOrderManagementButtonClick" text="Повернутися до меню" />

</children>

</VBox>

</children>

<padding>

<Insets left="10.0" right="10.0" />

</padding>

</VBox>

</center>

<left>

<Separator orientation="VERTICAL" BorderPane.alignment="CENTER" />

</left>

</BorderPane>

</right>

<center>

<BorderPane BorderPane.alignment="CENTER">

<center>

<ScrollPane fitToHeight="true" fitToWidth="true">

<content>

<TableView fx:id="listOfTicketsTableView" prefHeight="200.0" prefWidth="200.0">

<columns>

<TableColumn fx:id="idTicketColumn" prefWidth="75.0" text="Ід квитка" />

<TableColumn fx:id="fullNameColumn" prefWidth="75.0" text="ПІБ" />

<TableColumn fx:id="phoneNumberColumn" prefWidth="75.0" text="Телефон" />

<TableColumn fx:id="namesTypeServiceColumn" prefWidth="75.0" text="Назви послуг" />

<TableColumn fx:id="isReadyColumn" prefWidth="75.0" text="Готово" />

</columns>

<columnResizePolicy>

<TableView fx:constant="CONSTRAINED\_RESIZE\_POLICY" />

</columnResizePolicy>

</TableView>

</content>

</ScrollPane>

</center>

<top>

<Label text="Послуги у замовлені" BorderPane.alignment="CENTER" />

</top>

</BorderPane>

</center>

</BorderPane>

MainApplication.java

package org.vitapasser.photocopypoint;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Scene;

import javafx.stage.Stage;

import org.vitapasser.photocopypoint.Controller.OrderManagementController;

import org.vitapasser.photocopypoint.Model.PickUpStation;

import org.vitapasser.photocopypoint.Model.Register;

import org.vitapasser.photocopypoint.Model.TicketList;

import org.vitapasser.photocopypoint.Model.TypeList;

import org.vitapasser.photocopypoint.Util.Mysql;

import java.sql.Connection;

import java.sql.SQLException;

import java.util.Objects;

public class MainApplication extends Application {

Connection connection;

@Override

public void start(Stage stage){

try{

Class.forName("com.mysql.cj.jdbc.Driver").getDeclaredConstructor().newInstance();

System.out.println("Connection driver included!");

try {

connection = Mysql.getConnection();

TypeList typeList = new TypeList(connection);

PickUpStation pickUpStation = new PickUpStation(connection,

"Київ, вул. Хрещатик, 1");

TicketList ticketList = new TicketList(connection);

Register register = new Register(typeList, pickUpStation, ticketList);

FXMLLoader FXMLLoader = new FXMLLoader(Objects.requireNonNull(

MainApplication.class.getResource("order-management.fxml")));

Scene scene = new Scene(FXMLLoader.load());

OrderManagementController controller = FXMLLoader.getController();

controller.putData(register);

stage.setTitle("Hello!");

stage.setScene(scene);

stage.show();

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

catch(Exception ex){

System.out.println("Connection failed...");

System.out.println(ex.getMessage());

}

}

@Override

public void stop() throws SQLException {

connection.close();

}

public static void main(String[] args) {

launch();

}

}

PickUpStationTest.java

package org.vitapasser.photocopypoint.Model;

import org.junit.jupiter.api.Assertions;

import org.junit.jupiter.api.Test;

import org.vitapasser.photocopypoint.Util.Mysql;

import java.io.IOException;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Objects;

import static org.junit.jupiter.api.Assertions.\*;

class PickUpStationTest {

Connection connection = Mysql.getConnection();

PickUpStationTest() throws SQLException, IOException {

}

@Test

void getOperator() {

PickUpStation pickUpStation = new PickUpStation(connection,

"Київ, вул. Хрещатик, 1");

Operator operatorGet = pickUpStation.getOperator();

Operator operatorReference = new Operator(2L, "Петро Петренко");

Assertions.assertEquals(operatorReference, operatorGet);

}

@Test

void fixSale() {

PickUpStation pickUpStation = new PickUpStation(connection,

"Київ, вул. Хрещатик, 1");

Order order = Order.create();

TypeList typeList = new TypeList(connection);

order.addType("Друк", typeList, 3);

order.getTerm();

order.getPrice();

String fullName = "ТЕСТДорно ДованоТЕСТ";

order.setFullName(fullName);

order.setPhoneNumber("+3809998748904");

order.setOperator(new Operator(1L, "Петро Петренко"));

Money pay = new Money(500.00, "UAH");

OrderIDAndOddMany orderIDOddMoney = pickUpStation.fixSale(order, pay);

assert orderIDOddMoney != null;

Money odd\_money = orderIDOddMoney.oddMoney();

Money odd\_money\_check = new Money(pay.value() - order.getPrice().value(), pay.unit());

Boolean oddMoneyTest = Objects.equals(odd\_money, odd\_money\_check);

String sqlCheck = "SELECT `PhotocopyPoint`.`Contact\_info`.`full\_name`, `PhotocopyPoint`.`Order`.id FROM `PhotocopyPoint`.`Contact\_info`\n" +

"INNER JOIN `PhotocopyPoint`.Client on `PhotocopyPoint`.Client.contact\_info\_id = `PhotocopyPoint`.`Contact\_info`.`id`\n" +

"INNER JOIN `PhotocopyPoint`.`Order` on `PhotocopyPoint`.`Order`.`client\_id` = `PhotocopyPoint`.Client.id\n" +

"WHERE `PhotocopyPoint`.`Contact\_info`.`full\_name` = '"+fullName+"'\n" +

"GROUP BY `PhotocopyPoint`.`Order`.id ORDER BY `PhotocopyPoint`.`Order`.id DESC LIMIT 1;";

String fullNameCreated = null;

long orderId = -1;

try {

Statement statement = connection.createStatement();

ResultSet sqlResult = statement.executeQuery(sqlCheck);

sqlResult.next();

fullNameCreated = sqlResult.getString("full\_name");

orderId = sqlResult.getLong("id");

} catch (Exception e)

{

System.out.println("Error on test 'FixSale': " +e.getMessage());

e.printStackTrace();

}

Assertions.assertEquals(fullNameCreated, fullName);

Assertions.assertEquals(orderId, orderIDOddMoney.orderID());

Assertions.assertEquals(odd\_money, odd\_money\_check);

}

}

RegisterTest.java

package org.vitapasser.photocopypoint.Model;

import org.junit.jupiter.api.Assertions;

import org.junit.jupiter.api.Test;

import org.vitapasser.photocopypoint.Util.Mysql;

import java.io.IOException;

import java.sql.Connection;

import java.sql.SQLException;

import java.util.Objects;

import static org.junit.jupiter.api.Assertions.\*;

class RegisterTest {

Connection connection = Mysql.getConnection();

RegisterTest() throws SQLException, IOException {

}

@Test

void getTerm() {

TypeList typeList = new TypeList(connection);

PickUpStation pickUpStation = new PickUpStation(connection,

"Київ, вул. Хрещатик, 1");

TicketList ticketList = new TicketList(connection);

Register register = new Register(typeList, pickUpStation, ticketList);

register.newOrder();

register.addType("Друк", 57);

Term term = register.getTerm();

Term testTerm = new Term(57 \* 60);

Assertions.assertEquals(testTerm, term);

}

@Test

void getPrice() {

TypeList typeList = new TypeList(connection);

PickUpStation pickUpStation = new PickUpStation(connection,

"Київ, вул. Хрещатик, 1");

TicketList ticketList = new TicketList(connection);

Register register = new Register(typeList, pickUpStation, ticketList);

register.newOrder();

register.addType("Друк", 57);

Money price = register.getPrice();

Money testPrice = new Money(57 \* 100.0000, "UAH");

Assertions.assertEquals(testPrice, price);

}

@Test

void makePayment() {

TypeList typeList = new TypeList(connection);

PickUpStation pickUpStation = new PickUpStation(connection,

"Київ, вул. Хрещатик, 1");

TicketList ticketList = new TicketList(connection);

Register register = new Register(typeList, pickUpStation, ticketList);

register.newOrder();

register.addType("Друк", 57);

register.getTerm();

register.getPrice();

Money oddMoney = register.makePayment(new Money(60 \* 100.0000, "UAH"),

"Сергій Сергійович Сержі",

"+3809844678983");

Money testOddMoney = new Money(3 \* 100.0000, "UAH");

Assertions.assertEquals(testOddMoney, oddMoney);

}

}

TicketList.java

package org.vitapasser.photocopypoint.Model;

import org.junit.jupiter.api.Assertions;

import org.junit.jupiter.api.Test;

import org.vitapasser.photocopypoint.Util.Mysql;

import java.io.IOException;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

class TicketListTest {

Connection connection = Mysql.getConnection();

TicketListTest() throws SQLException, IOException {

}

@Test

void getAllTickets() {

TypeList typeList = new TypeList(connection);

PickUpStation pickUpStation = new PickUpStation(connection,"Київ, вул. Хрещатик, 1");

TicketList ticketList = new TicketList(connection);

Register register = new Register(typeList, pickUpStation, ticketList);

List<Ticket> tickets = register.getAllTickets();

List<Ticket> ticketReference = new ArrayList<>();

ticketReference.add(new Ticket(1, 1, "Друк", false, "Іван Іванов", "+380501234567"));

Assertions.assertEquals(ticketReference.getFirst(), tickets.getFirst());

}

@Test

void createTicket() {

TicketList ticketList = new TicketList(connection);

long idOrder = 1;

long idTicket = ticketList.createTicket(idOrder);

String sqlCheck = "SELECT `PhotocopyPoint`.`Order`.id as OrderId, `PhotocopyPoint`.`Ticket`.id as TicketId\n" +

"FROM `PhotocopyPoint`.`Ticket`\n" +

"INNER JOIN `PhotocopyPoint`.`Order` " +

"on `PhotocopyPoint`.`Order`.`id` = `PhotocopyPoint`.Ticket.order\_id\n" +

"WHERE `PhotocopyPoint`.`Order`.`id` = "+idOrder+"\n" +

"GROUP BY OrderId, TicketId " +

"ORDER BY OrderId, TicketId DESC LIMIT 1;";

long testOrderId = -1, testTicketId = -1;

try {

Statement statement = connection.createStatement();

ResultSet sqlResult = statement.executeQuery(sqlCheck);

sqlResult.next();

testOrderId = sqlResult.getLong("OrderId");

testTicketId = sqlResult.getLong("TicketId");

} catch (Exception e)

{

System.out.println("Error on test 'FixSale': " +e.getMessage());

e.printStackTrace();

}

Assertions.assertEquals(idOrder, testOrderId);

Assertions.assertEquals(idTicket, testTicketId);

}

}

TypeListTest.java

package org.vitapasser.photocopypoint.Model;

import org.junit.jupiter.api.Assertions;

import org.junit.jupiter.api.Test;

import org.vitapasser.photocopypoint.Exception.NotExistTypeException;

import org.vitapasser.photocopypoint.Util.Mysql;

import java.io.IOException;

import java.sql.Connection;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

class TypeListTest {

Connection connection = Mysql.getConnection();

TypeListTest() throws SQLException, IOException {

}

@Test

void getTypes(){

TypeList typeList = new TypeList(connection);

List<Type> typesGet;

try {

typesGet = typeList.getTypes("д");

} catch (NotExistTypeException e) {

throw new RuntimeException(e);

}

List<Type> typesReference = new ArrayList<>();

typesReference.add(new Type(5L, "Друк", "Чорно-білий друк",

new Term(60), new Money(100.0000, "UAH"), Mysql.dbDateTimeToLocalDateTime("2024-05-27 11:48:27")));

Assertions.assertEquals(typesReference, typesGet);

}

@Test

void addType(){

TypeList typeList = new TypeList(connection);

Type typeGot = typeList.getType("Копіювання");

Type typeReference = new Type(2L, "Копіювання", "Кольорове копіювання",

new Term(150), new Money(200.0000, "UAH"), Mysql.dbDateTimeToLocalDateTime("2024-05-27 11:48:24"));

Assertions.assertEquals(typeGot, typeReference);

}

}

Modul-info.java

module org.vitapasser.photocopypoint {

requires javafx.controls;

requires javafx.fxml;

requires javafx.web;

requires org.controlsfx.controls;

requires com.dlsc.formsfx;

requires net.synedra.validatorfx;

requires org.kordamp.ikonli.javafx;

requires org.kordamp.bootstrapfx.core;

requires eu.hansolo.tilesfx;

requires com.almasb.fxgl.all;

requires mysql.connector.j;

requires java.sql;

opens org.vitapasser.photocopypoint to javafx.fxml;

exports org.vitapasser.photocopypoint;

exports org.vitapasser.photocopypoint.Controller;

opens org.vitapasser.photocopypoint.Controller to javafx.fxml;

exports org.vitapasser.photocopypoint.Model;

opens org.vitapasser.photocopypoint.Model to javafx.fxml;

exports org.vitapasser.photocopypoint.Controller.CreatingOrder;

opens org.vitapasser.photocopypoint.Controller.CreatingOrder to javafx.fxml;

}