

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж

Звіт  
до лабораторної роботи №4  
з дисципліни «Екстримальне програмування»  
на тему «Створення UI для розроблюваної системи»

Виконала  
студентка групи КН-311  
Мельничук М.П.

Прийняв  
Щербак С. С.

Львів-2020

**Мета:** створення UI для обраної системи.

## **Теоретичні відомості**

Інтерфейс користувача (ІК), (англ. user interface, UI) — засіб зручної взаємодії користувача з інформаційною системою. Сукупність засобів для обробки та відбиття інформації, якнайбільше пристосованих для зручності користувача; у графічних системах інтерфейс користувача, втілюється багатовіконним режимом, змінами кольору, розміру, видимості (прозорість, напівпрозорість, невидимість) вікон, їхнім розташуванням, сортуванням елементів вікон, гнучкими налаштуваннями як самих вікон, так і окремих їх елементів (файли, теки, ярлики, шрифти тощо), доступністю багатокористувацьких налаштувань.

Інтерфейс користувача, в галузі промислового дизайну, взаємодії між людиною та комп'ютером — це простір, де відбувається співпраця між людьми та машинами. Мета цієї співдії, полягає у тому, щоби забезпечити досконалу роботу та керування машиною з боку людини, а машина одночасно, надає інформацію, яка допомагає прийняттю рішень операторами. Прикладами цієї широкої концепції інтерфейсів користувача, є: інтерактивні аспекти комп'ютерних операційних систем, ручні інструменти, операторські засоби керування важким обладнанням та елементи контролю над технологічними процесами. Конструктивні міркування, що застосовуються під час створення інтерфейсів користувача, пов'язано з такими галузями, як ергономіка та психологія.

Як правило, мета створення інтерфейсу користувача, полягає у тому, щоби зробити інтерфейс користувача, який спрощує (самозрозумілий), ефективний і приємний (зручний для користувача) для керування машиною таким чином, щоби забезпечити бажаний результат. Це, зазвичай, означає, що оператор повинен застосовувати щонайменші зусилля для досягнення очікуваного підсумку, а також, щоби машина зменшувала небажані результати для людини. З посиленням використання персональних комп'ютерів та відносним зниженням обізнаності (інтересу) суспільства про важкі машини, термін «інтерфейс користувача», як правило, передбачає графічний інтерфейс користувача, тоді як промислові панелі керування та механізми контролю за обладнанням, частіше стосуються (НМІ) людино-машинної взаємодії.

## **Хід роботи**

Для побудови UI для системи підтримки прийняття рішень щодо розміщення товарів відповідно споживчого кошика було обрано платформу і набір інструментів JavaFX версії 11.0. JavaFX включає в себе набір утиліт, за допомогою яких веб-розробники та дизайнери можуть швидко створювати та

надавати розвинуті інтернет-застосунки для десктопів, мобільних пристроїв, телебачення та інших платформ.

Розробимо всі необхідні форми для функціонування нашого програмного продукту.

*Лістинг коду вікна StartForm:*

```
public class StartForm extends JFrame{

    private JButton login;

    private JButton register;

    public JPanel mainPanel;

    private static JFrame jRegisterFrame = new JFrame("RegisterForm");

    private static JFrame jLoginFrame = new JFrame("LoginForm");

    public StartForm() {

        login.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                jLoginFrame.setContentPane(new LoginForm().mainLoginPanel);

                jLoginFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

                jLoginFrame.pack();

                jLoginFrame.setVisible(true);

            }

        });

        register.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                jRegisterFrame.setContentPane(new RegisterForm().mainRegisterPanel);

                jRegisterFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

                jRegisterFrame.pack();

                jRegisterFrame.setVisible(true);

            }

        });

    }

    public static JFrame getjRegisterFrame() {

        return jRegisterFrame;

    }

    public static JFrame getjLoginFrame() {

        return jLoginFrame;

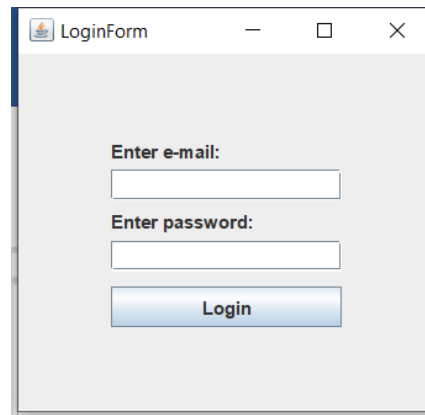
    }

}
```

```

    }
}

```



*Рис. 4.1 Вікно входу у систему*

*Лістинг коду вікна MainForm:*

```

public class MainForm {
    private JLabel wellcomeText;
    private JLabel userName;
    private JLabel userSurname;
    private JLabel userEmail;
    private JButton shops;
    public JPanel mainMainPanel;
    private JTable shopsTable;
    private static JFrame jShopFrame = new JFrame("ShopForm");
    private List<Shop> shopsData;
    public MainForm(Integer userId) {
        User currentUser = UserRepository.getUser(userId);
        userName.setText(currentUser.getName());
        userSurname.setText(currentUser.getSurname());
        userEmail.setText(currentUser.getEmail());
        shops.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                shopsData = ShopRepository.getUsersShop(userId);
                DefaultTableModel model = new DefaultTableModel() {
                    @Override
                    public boolean isCellEditable(int row, int column) {
                        return false;
                    }
                };
                DefaultTableCellRenderer color = new DefaultTableCellRenderer();
                color.setForeground(Color.red);
                String [] columns = new String[] { "Id",
                    "Name",

```

```

        "Country",
        "City",
        "Owner"
    };

    for(String columnName : columns) {
        model.addColumn(columnName);
    }

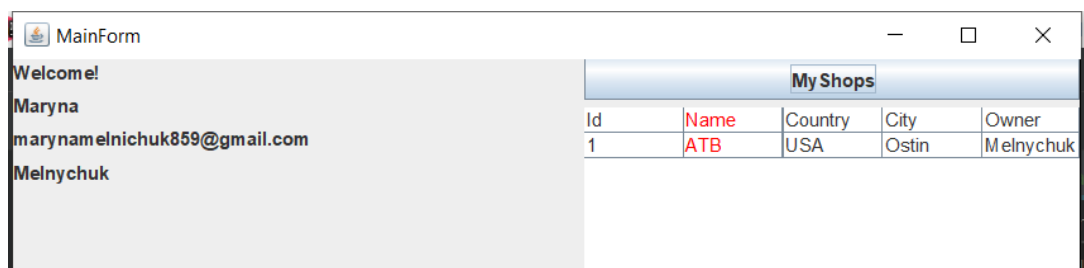
    Object [][] data = new Object[shopsData.size()+1][5];
    model.addRow(columns);
    for(int i = 1; i < data.length; i++) {
        data[i][0] = shopsData.get(i-1).getId();
        data[i][1] = shopsData.get(i-1).getName();
        data[i][2] = shopsData.get(i-1).getCountry();
        data[i][3] = shopsData.get(i-1).getCity();
        data[i][4] = shopsData.get(i-1).getUser().getSurname();
        model.addRow(data[i]);    }

    shopsTable.setModel(model);
    shopsTable.getColumnModel().getColumn(1).setCellRenderer(color);
    }
});

shopsTable.addComponentListener(new ComponentAdapter() {
});

shopsTable.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        super.mouseClicked(e);
        int row = shopsTable.rowAtPoint(e.getPoint());
        int col = shopsTable.columnAtPoint(e.getPoint());
        if (row > 0 && col == 1) {
            jShopFrame.setContentPane(new ShopForm(shopsData.get(row-1).getId()).mainShopPanel);
            jShopFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            jShopFrame.pack();
            jShopFrame.setVisible(true);
        } }
});
}}

```



*Рис. 4.2 Головне вікно користувача зі списком магазинів*

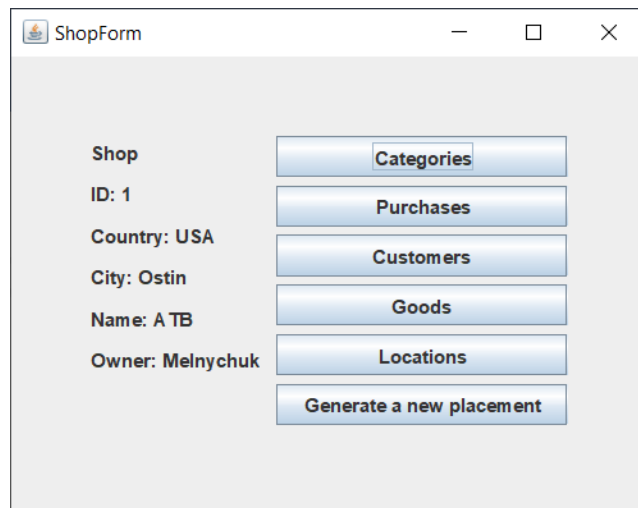
### *Лістинг коду вікна ShopForm:*

```
public class ShopForm {
    public JPanel mainShopPanel;
    private JLabel shopLabel;
    private JLabel shopName;
    private JLabel shopCountry;
    private JLabel shopId;
    private JLabel shopCity;
    private JLabel shopOwner;
    private JButton shopCategory;
    private JButton shopLocation;
    private JButton shopGoods;
    private JButton shopPurchase;
    private JButton shopCustomers;
    private JButton generateANewPlacement;
    private static JFrame jCategoriesFrame = new JFrame("CategoriesForm");
    private static JFrame jPurchaseFrame = new JFrame("PurchaseForm");
    private static JFrame jCustomerFrame = new JFrame("CustomerForm");
    private static JFrame jGoodsFrame = new JFrame("GoodsForm");
    private static JFrame jLocationsFrame = new JFrame("LocationsForm");
    private static JFrame jCategoryLocationFrame = new JFrame("CategoryLocationForm");
    public static boolean isPurchaseCalling = true;
    public static boolean isCustomerCalling = true;
    public ShopForm(Integer currentShopId) {
        Shop shop = ShopRepository.getShop(currentShopId);
        shopName.setText("Name: " + shop.getName());
        shopCountry.setText("Country: " + shop.getCountry());
        shopCity.setText("City: " + shop.getCity());
        shopOwner.setText("Owner: " + shop.getUser().getSurname());
        shopId.setText("ID: "+currentShopId);
        shopCategory.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                jCategoriesFrame.setContentPane(new CategoriesForm(currentShopId).mainCategoriesPanel);
                jCategoriesFrame.pack();
                jCategoriesFrame.setVisible(true);
            }
        });
        shopPurchase.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                isPurchaseCalling = false;
                jPurchaseFrame.setContentPane(new PurchaseForm(currentShopId).mainPurchasePanel);
                jPurchaseFrame.pack();
            }
        });
    }
}
```

```

        jPurchaseFrame.setVisible(true);
    }
});
shopCustomers.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        isCustomerCalling = false;
        jCustomerFrame.setContentPane(new CustomersForm(currentShopId).mainCustomerPanel);
        jCustomerFrame.pack();
        jCustomerFrame.setVisible(true);
    }
});
shopGoods.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        jGoodsFrame.setContentPane(new GoodsForm(currentShopId).mainGoodsPanel);
        jGoodsFrame.pack();
        jGoodsFrame.setVisible(true);
    }
});
shopLocation.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        jLocationsFrame.setContentPane(new LocationsForm(currentShopId).mainLocationsPanel);
        jLocationsFrame.pack();
        jLocationsFrame.setVisible(true);
    }
});
generateANewPlacement.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        jCategoryLocationFrame.setContentPane(new CategoryLocationForm(currentShopId).mainCategoryLocationPanel);
        jCategoryLocationFrame.pack();
        jCategoryLocationFrame.setVisible(true);
    }
});
}
public static JFrame getjCategoryLocationFrame() {
    return jCategoryLocationFrame;
}
}

```



*Рис. 4.3 Вікно користувача для конкретного магазину*

*Лістинг коду вікна CategoriesForm:*

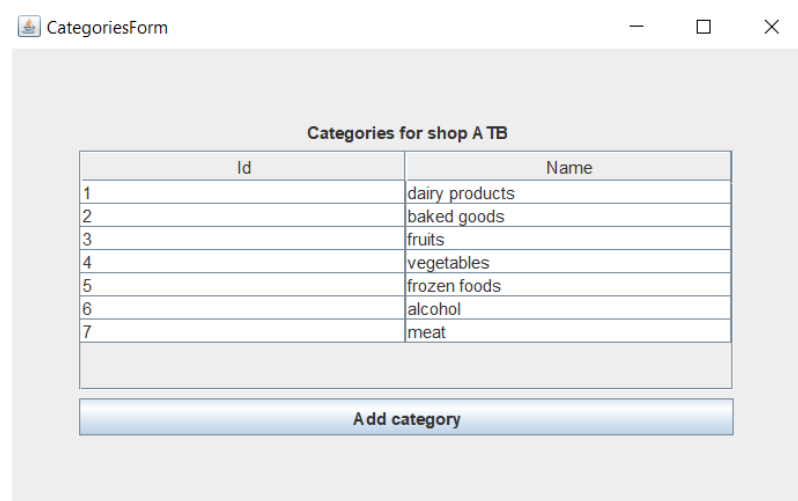
```
public class CategoriesForm {
    private JTable categoriesTable;
    private JLabel shopName;
    private JButton addCategory;
    public JPanel mainCategoriesPanel;
    private static JFrame jAddCategoryFrame = new JFrame("AddCategoryForm");
    public CategoriesForm(Integer currentShopId){
        Shop shop = ShopRepository.getShop(currentShopId);
        shopName.setText("Categories for shop " + shop.getName());
        List<Category> categoriesData = CategoryRepository.getCategoriesShop(currentShopId);
        DefaultTableModel model = new DefaultTableModel() {
            @Override
            public boolean isCellEditable(int row, int column) {
                return false;
            }
        };
        String [] columns = new String[] { "Id",
            "Name"
        };
        for(String columnName : columns) {
            model.addColumn(columnName);
        }
        Object [][] data = new Object[categoriesData.size()][2];
        //com.kursova.model.addRow(columns);
        for(int i = 0; i < data.length; i++) {
            data[i][0] = categoriesData.get(i).getId();
            data[i][1] = categoriesData.get(i).getName();
            model.addRow(data[i]);
        }
    }
}
```



```

categoriesTable.setModel(model);
addCategory.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        jAddCategoryFrame.setContentPane(new AddCategoryForm(currentShopId).mainAddCategoryPanel);
        jAddCategoryFrame.pack();
        jAddCategoryFrame.setVisible(true);
    }
});
}
public static JFrame getjAddCategoryFrame() {
    return jAddCategoryFrame;
}
}

```

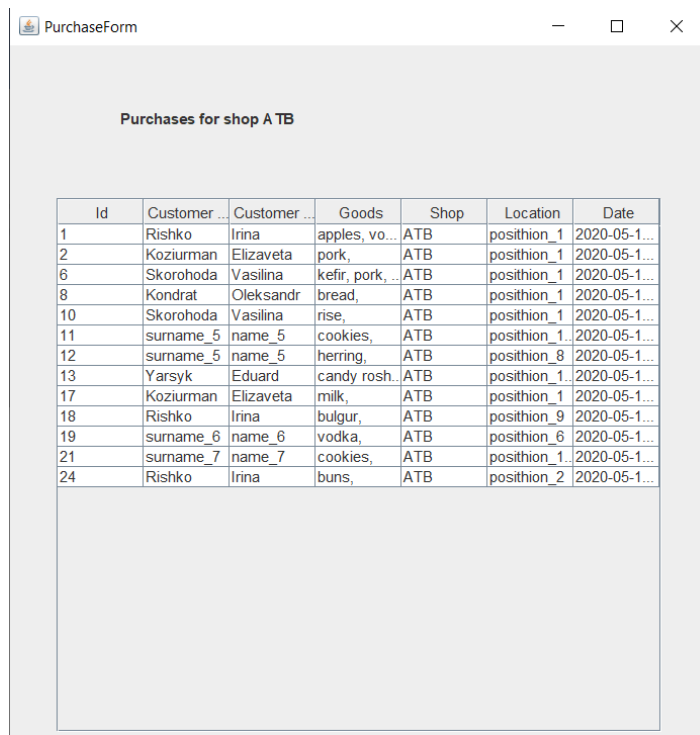


*Рис. 4.4 Вікно списку категорій для конкретного магазину*

Для подальших розроблених вікон буде прикріплене посилання з відповідним ресурсним кодом у створеному репозиторії:

*Лістинг коду вікна PurchaseForm:*

<https://github.com/maryamelnichuk/EP--PROJECT/blob/master/src/main/java/com/kursova/ui/PurchaseForm.java>



**Purchases for shop ATB**

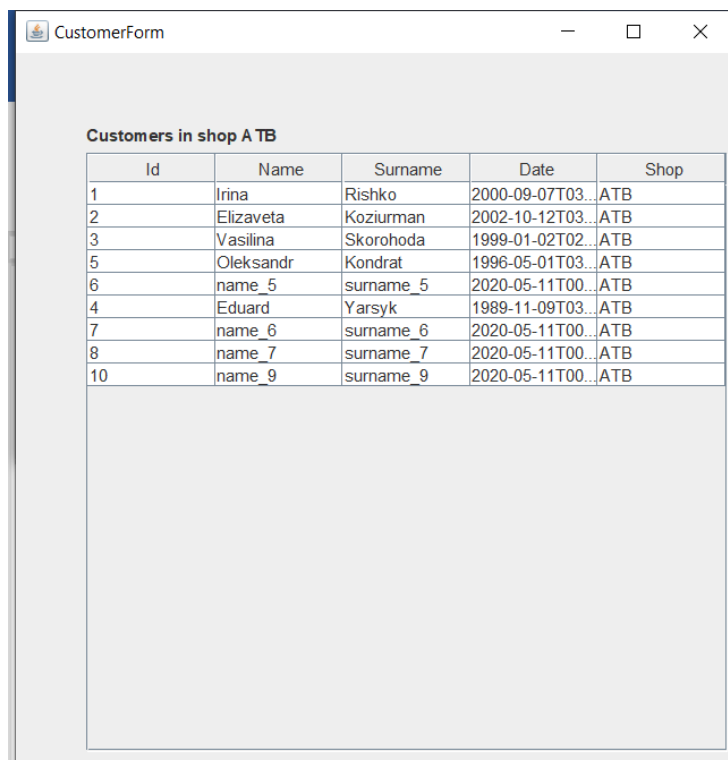
Id	Customer ...	Customer ...	Goods	Shop	Location	Date
1	Rishko	Irina	apples, vo...	ATB	posithion_1	2020-05-1...
2	Koziurman	Elizaveta	pork,	ATB	posithion_1	2020-05-1...
6	Skorohoda	Vasilina	kefir, pork, ..	ATB	posithion_1	2020-05-1...
8	Kondrat	Oleksandr	bread,	ATB	posithion_1	2020-05-1...
10	Skorohoda	Vasilina	rise,	ATB	posithion_1	2020-05-1...
11	surname_5	name_5	cookies,	ATB	posithion_1	2020-05-1...
12	surname_5	name_5	herring,	ATB	posithion_8	2020-05-1...
13	Yarsyk	Eduard	candy rosh.	ATB	posithion_1	2020-05-1...
17	Koziurman	Elizaveta	milk,	ATB	posithion_1	2020-05-1...
18	Rishko	Irina	bulgur,	ATB	posithion_9	2020-05-1...
19	surname_6	name_6	vodka,	ATB	posithion_6	2020-05-1...
21	surname_7	name_7	cookies,	ATB	posithion_1	2020-05-1...
24	Rishko	Irina	buns,	ATB	posithion_2	2020-05-1...

Рис. 4.5 Вікно списку покупок в конкретному магазині

Лістинг коду вікна CustomersForm:

[https://github.com/marynamelnichuk/EP--](https://github.com/marynamelnichuk/EP--PROJECT/blob/master/src/main/java/com/kursova/ui/CustomersForm.java)

[PROJECT/blob/master/src/main/java/com/kursova/ui/CustomersForm.java](https://github.com/marynamelnichuk/EP--PROJECT/blob/master/src/main/java/com/kursova/ui/CustomersForm.java)



**Customers in shop ATB**

Id	Name	Surname	Date	Shop
1	Irina	Rishko	2000-09-07T03...	ATB
2	Elizaveta	Koziurman	2002-10-12T03...	ATB
3	Vasilina	Skorohoda	1999-01-02T02...	ATB
5	Oleksandr	Kondrat	1996-05-01T03...	ATB
6	name_5	surname_5	2020-05-11T00...	ATB
4	Eduard	Yarsyk	1989-11-09T03...	ATB
7	name_6	surname_6	2020-05-11T00...	ATB
8	name_7	surname_7	2020-05-11T00...	ATB
10	name_9	surname_9	2020-05-11T00...	ATB

Рис. 4.6 Вікно списку покупців в конкретному магазині

Лістинг коду вікна GoodsForm:

[https://github.com/marynamelnichuk/EP--](https://github.com/marynamelnichuk/EP--PROJECT/blob/master/src/main/java/com/kursova/ui/GoodsForm.java)

[PROJECT/blob/master/src/main/java/com/kursova/ui/GoodsForm.java](https://github.com/marynamelnichuk/EP--PROJECT/blob/master/src/main/java/com/kursova/ui/GoodsForm.java)

**Goods for shop A TB**

Id	Name	Category
4	milk	dairy products
5	yogurt	dairy products
6	kefir	dairy products
7	cheese	dairy products
3	bread	baked goods
22	buns	baked goods
1	apples	fruits
17	oranges	fruits
24	bananas	fruits
2	cucumber	vegetables
18	potato	vegetables
25	tomatoes	vegetables
11	dumplings	frozen foods
19	pancakes	frozen foods
8	vodka	alcohol
20	beer	alcohol
9	pork	meat
10	beef	meat

**Add good**

*Рис. 4.7 Вікно списку товарів в конкретному магазині*

*Лістинг коду вікна LocationsForm:*

[https://github.com/marynamelnichuk/EP--](https://github.com/marynamelnichuk/EP--PROJECT/blob/master/src/main/java/com/kursova/ui/LocationsForm.java)

[PROJECT/blob/master/src/main/java/com/kursova/ui/LocationsForm.java](https://github.com/marynamelnichuk/EP--PROJECT/blob/master/src/main/java/com/kursova/ui/LocationsForm.java)

**Locations for shop A TB**

Id	Name	Distance	Side	Shop
1	posithion_1	1.5	left	ATB
2	posithion_2	2.0	right	ATB
3	posithion_3	2.5	center	ATB
4	posithion_4	1.7	left	ATB
5	posithion_5	2.0	right	ATB
11	posithion_11	1.5	right	ATB

**Add Location**

*Рис. 4.8 Вікно списку місць розташувань в конкретному магазині*

*Лістинг коду вікна CategoryLocationForm:*

[https://github.com/marynamelnichuk/EP--](https://github.com/marynamelnichuk/EP--PROJECT/blob/master/src/main/java/com/kursova/ui/CategoryLocationForm.java)

[PROJECT/blob/master/src/main/java/com/kursova/ui/CategoryLocationForm.java](https://github.com/marynamelnichuk/EP--PROJECT/blob/master/src/main/java/com/kursova/ui/CategoryLocationForm.java)

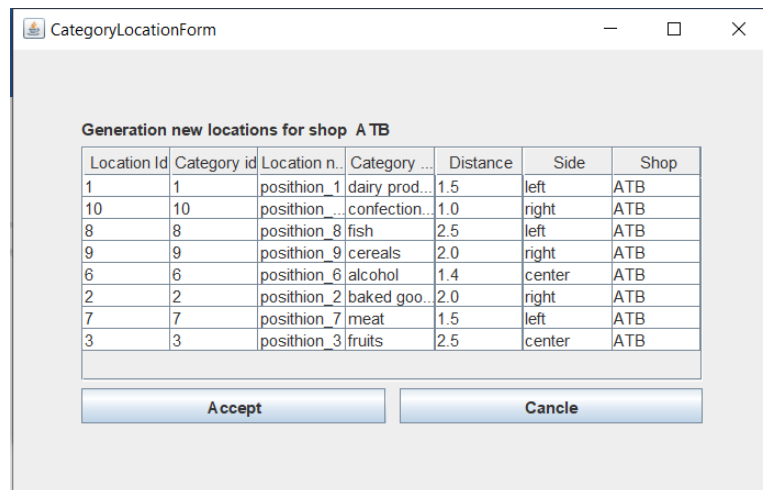


Рис. 4.9 Вікно згенерованого місця розташування категорій

Лістинг коду вікна AddCategoryForm:

[https://github.com/marynamelnichuk/EP--](https://github.com/marynamelnichuk/EP--PROJECT/blob/master/src/main/java/com/kursova/ui/AddCategoryForm.java)

[PROJECT/blob/master/src/main/java/com/kursova/ui/AddCategoryForm.java](https://github.com/marynamelnichuk/EP--PROJECT/blob/master/src/main/java/com/kursova/ui/AddCategoryForm.java)

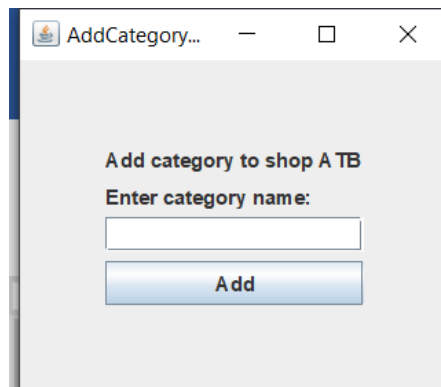


Рис. 4.10 Вікно додавання нової категорії для конкретного магазину

Лістинг коду вікна AddGoodForm:

[https://github.com/marynamelnichuk/EP--](https://github.com/marynamelnichuk/EP--PROJECT/blob/master/src/main/java/com/kursova/ui/AddGoodForm.java)

[PROJECT/blob/master/src/main/java/com/kursova/ui/AddGoodForm.java](https://github.com/marynamelnichuk/EP--PROJECT/blob/master/src/main/java/com/kursova/ui/AddGoodForm.java)

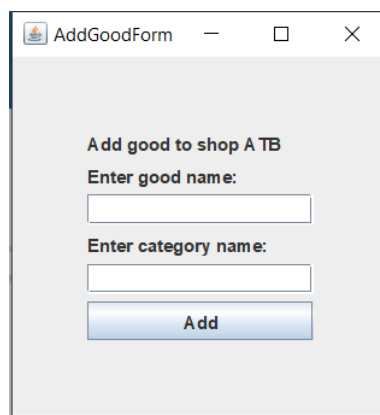
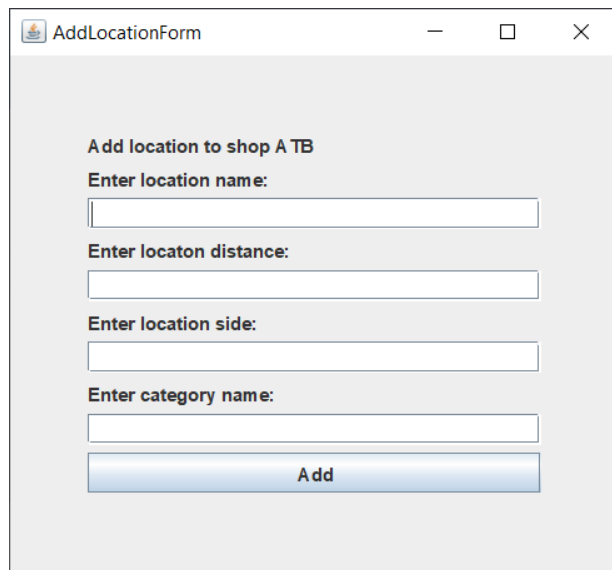


Рис. 4.11 Вікно додавання нового товару для конкретного магазину

*Лістинг коду вікна AddLocationForm:*

[https://github.com/marynamelnichuk/EP--](https://github.com/marynamelnichuk/EP--PROJECT/blob/master/src/main/java/com/kursova/ui/AddLocationForm.java)

[PROJECT/blob/master/src/main/java/com/kursova/ui/AddLocationForm.java](https://github.com/marynamelnichuk/EP--PROJECT/blob/master/src/main/java/com/kursova/ui/AddLocationForm.java)



Add location to shop A TB

Enter location name:

Enter locaton distance:

Enter location side:

Enter category name:

Add

*Рис. 4.12 Вікно додавання нового місця розташування для конкретного магазину*

**Висновок:** в результаті лабораторної роботи було створено UI для системи підтримки прийняття рішень щодо розміщення товарів відповідно споживчого кошику.