Explanatory note for **ERPForYou** project (Just enjoy it!)

# 1. Topic of the project

ERPForYou

# 2. Annotation

## 2.1. Program description

The program is called **ERPForYou**. It is an mini ERP-system for the manufacturing company of advertising samples (small business). In this program a user can manage process of income and expenditure of materials, get the general price of the inserting order and see the quantity of each material in a warehouse at present time.

## 2.2. Server logic and API

MySQL Database is located on a remote server (kornilova.styleru.net), it was created using phpmyadmin. It provides the possibility of online remote work with the database from multiple devices. The API which provides all types of work with DB like editing, adding and deleting items is written on PHP (it is the "web" folder in the repository). We also used fat-free framework to shorten the PHP code and protect the database from SQL injections.
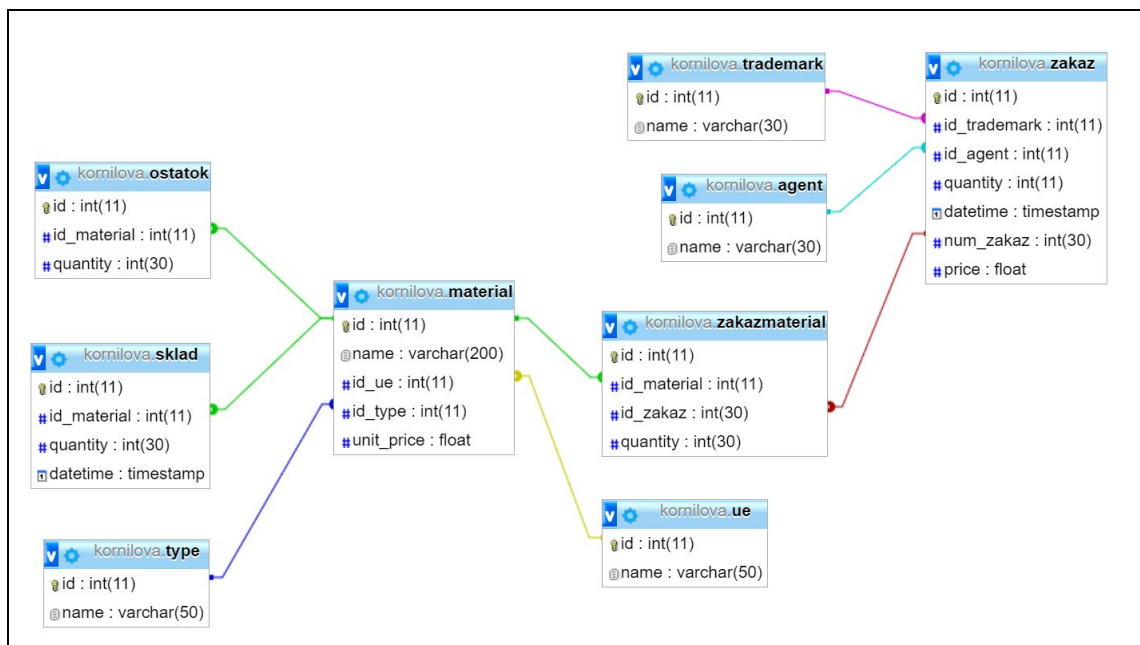


Database diagram

## 2.3. Program

**MVVM (Model-View-ViewModel) pattern** is implemented in our program. This pattern enables us to separate data/logic and UI.

Also <u>additional libraries</u> are used in the program:

**Newtonsoft.Json**
**MVVMLight**
**MVVMLightLibs**
**CommonServiceLocator**

The solution consists of only one project.
In the project there are 3 folders:

**Model folder** - classes for database entities (models) are located there,
**ViewModel folder** - ViewModels classes with almost all logic are located there,
**View folder** - UI is located there, but it doe not know anything about logic and data.
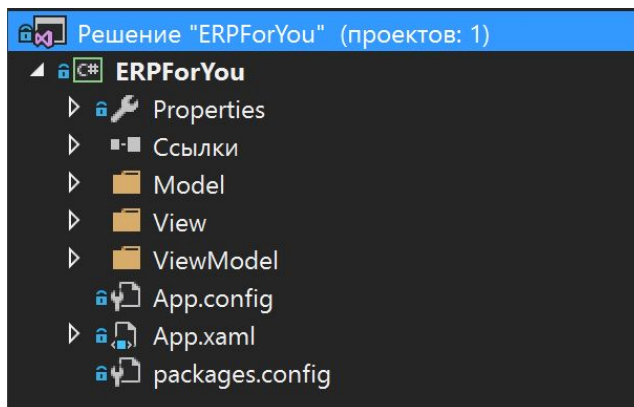


Project structure

## 3.    Central repository address

https://github.com/skor98/ERPForYou

## 4.    Members of the team

| Name and group | Role |
|---|---|
| Natalia Pestova 154(1) | server logic, API |
| Svetlana Kornilova 154(2) | UI/UX, abstractions, database interconnection |
| Parvin Tagieva 153 | UI, content, client logic |
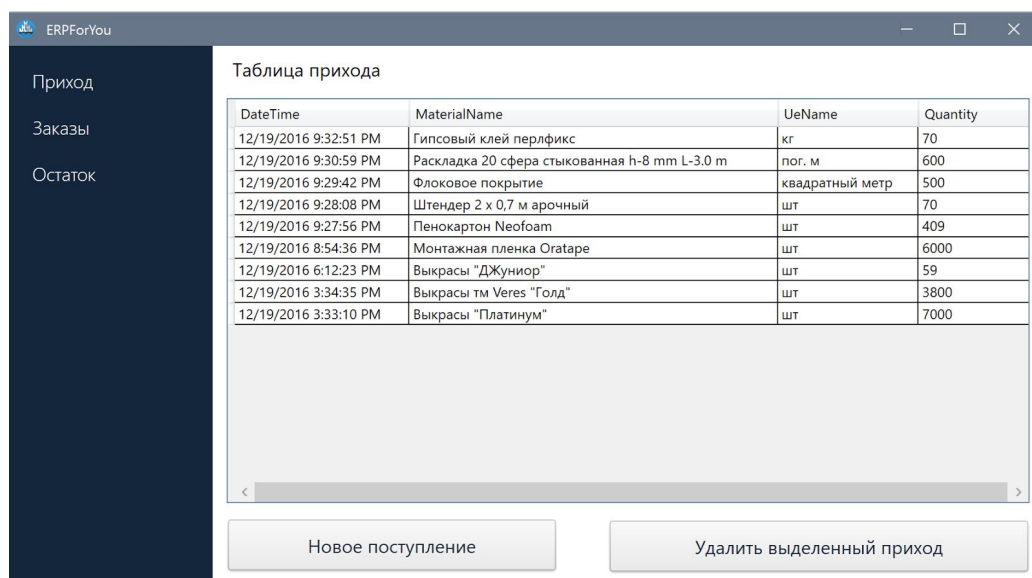
## 5. List of classes

| Class | Annotation |
|---|---|
| **Model folder** | |
| Agent.cs | Basic classes which are created for database integration. Contain the same editable fields as their DB prototypes. Do not have any declared methods. |
| Material.cs | |
| Ostatok.cs | |
| Sklad.cs | |
| Trademark.cs | |
| Type.cs | |
| UE.cs | |
| Zakaz.cs | |
| Zakaz_material.cs | |
| **ViewModel folder** | |
| Repository.cs | Main repository of the data from the DB. Contains lists of items of classes from the Model folder. Also contains update method for each list which gets the information from the DB using API. |
| ViewModelLocator.cs | Automatically created class by MVVMLight library. |
| DelegateCommand.cs | Basic class for implementing commands in ViewModel classes. |
| **ViewModelPattern folder** | Classes in this folder are for accurate representing data in the tables. |
| AddingNewMaterialViewModel.cs and other classes | These classes contain all logic for each View page except navigation. |
| **View folder** | |
| **Icon folder** | Only one icon is located here. |
| MainWndow.xaml | Main window of the application which consists a frame, where other pages appear. |
| AddingNewMaterialViewModel.xaml and other pages | Application UI and makeup are located here. |

## 6.    Program interface

**ERPForYou** is an multiple window WPF application. The MainWindow consists of a fixed dark navigation sidebar and one main frame control where other pages of the application appear.

First of all you can see the receipt table screen (Picture 1). It is possible to delete row from this table and database by clicking the right button if this receipt row is mistaken.
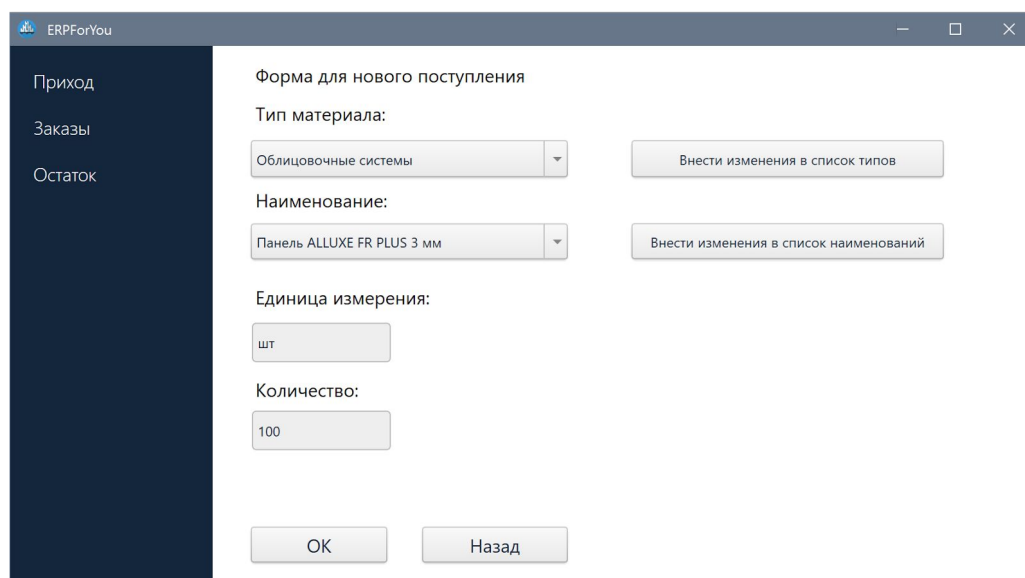


Picture 1

After clicking the left button a form for a new receipt will appear (Picture 2). Due to the fact that our application is for small business which does not have so many materials in the warehouse, we decided to add comboboxes filled with data from the database in order to make forms filling more comfortable and faster. The second combobox will represent only materials of the type selected above. The first textbox fills out automatically for the material selected above. Back button at the bottom relocates you to the previous page.



Picture 2

After clicking one of the buttons on the previous page you will navigate to editing page of type list or name list respectively. The picture 3 represents the editing page for type list. Typing in the first textbox here makes the add button enabled. If there is mistake in the type name in the list, you can change this name by selecting it in the list, inserting the new name in the second textbox and clicking the change button (Picture 4). All the changes will appear in the listbox and database immediately.



Picture 3



Picture 4

The picture 5 represents the first editing page for name list. Then you can navigate to picture 6 and add new material of type selected on the receipt form page (picture 2) setting its price and measure.



Picture 5



Picture 6

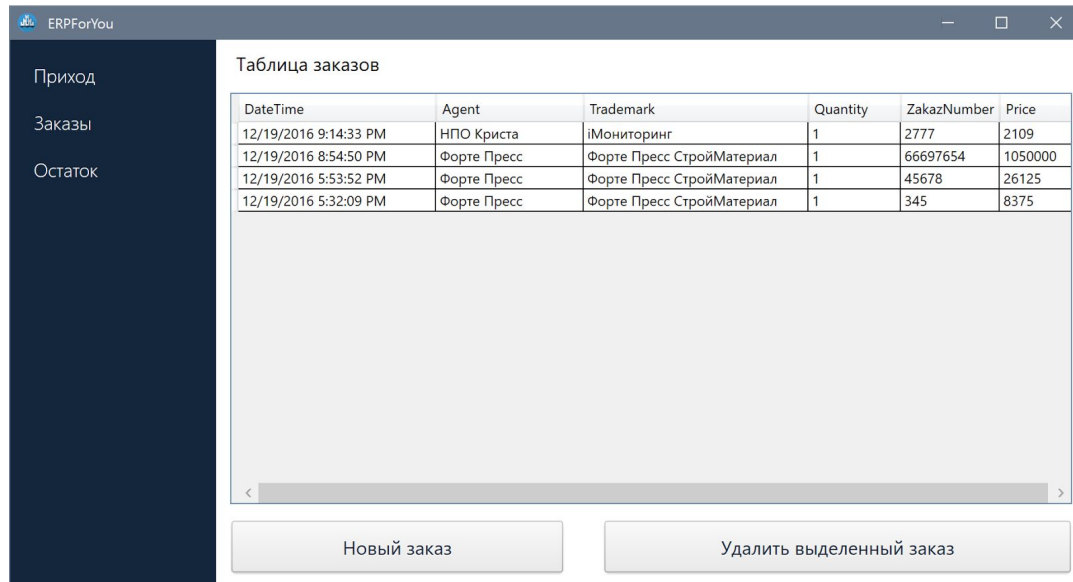The expenditure table page is just the same as the receipt one (Picture 7).
Table includes:

**agent** - client company,

**trademark** - agent's product which is going to be advertised by using materials,

**quantity** - quantity of different materials in the order,
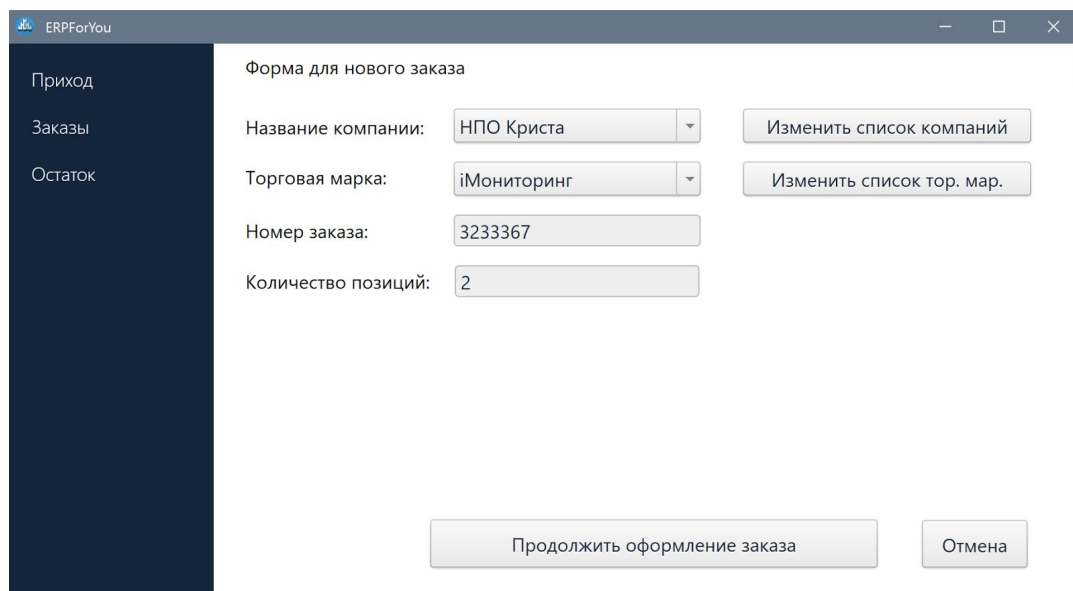
**price** - calculated by the program

and other obvious columns.



Picture 7

Form for the expenditure is depicted on picture 8. In the last textbox quantity of different materials should be inserted. Buttons on the right side will navigate you to editing pages (just the same as Picture 4).
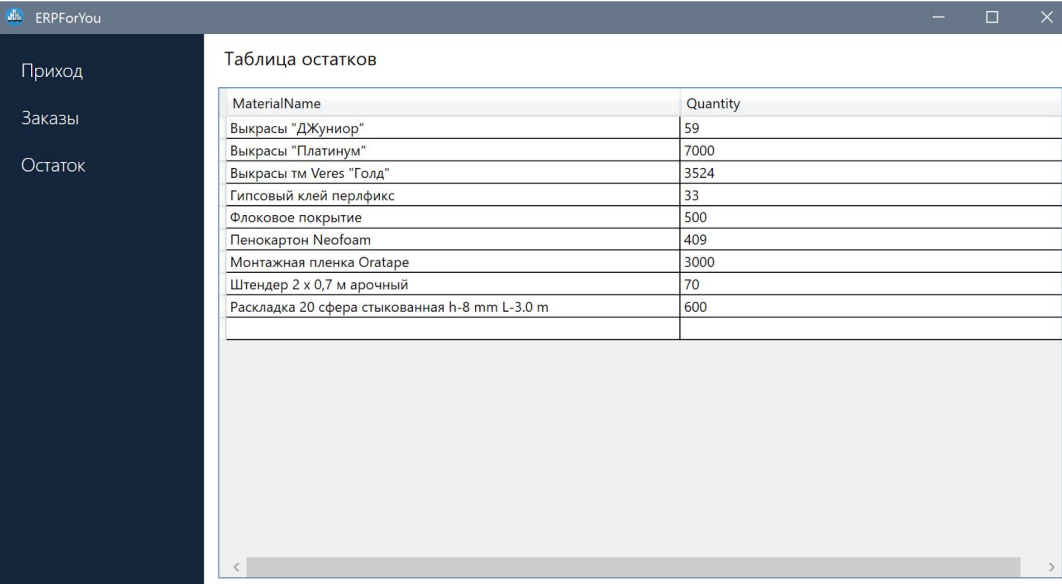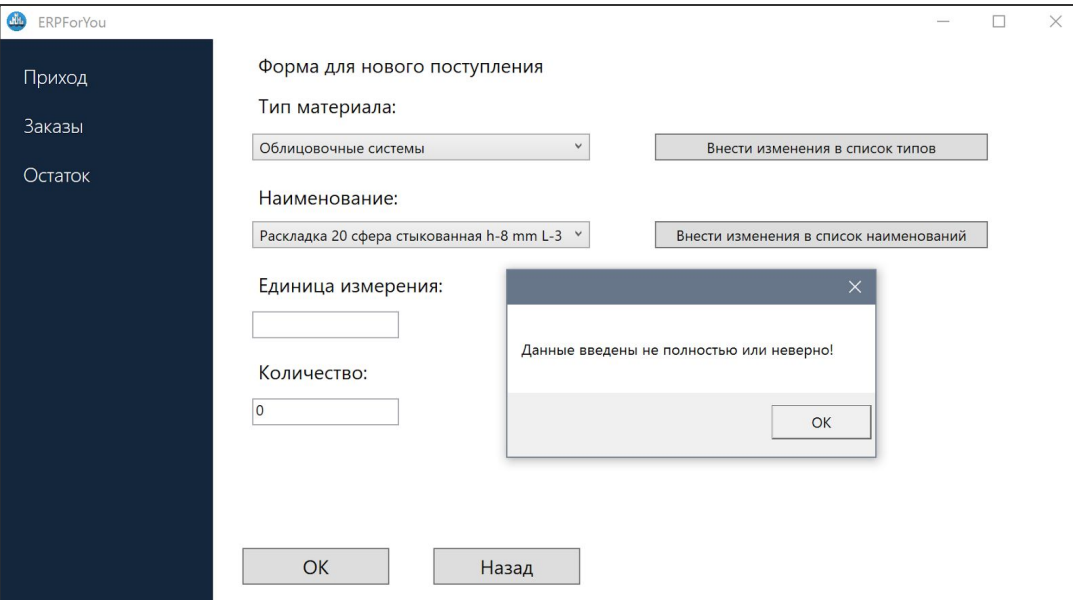


Picture 8

Finally, it is possible to see remainder of the materials in the warehouse (Picture 9). The quantity is also calculated by the program which takes into account orders and incomes both.



Picture 9

In application exception handling is also implemented. The MessageBox with exception message will appear immediately after an exception (Picture 10).



Picture 10