2018

Author:

Todor Vasilev

Programming languages 2



Table of Contents

[Introduction 2](#_Toc514013543)

[System Requirements 2](#_Toc514013544)

[Use Cases 2](#_Toc514013545)

[Class Diagram 2](#_Toc514013546)

[User Guide 4](#_Toc514013547)

# Introduction

AgriIPCA is a console application which main goal is provide finding and buying the needed agricultural products for people. It also has an administrative panel for managing the warehouse turnover of the company.

It is developed using the C# programming language. The application uses also SQL database to store the information about the products and users.

In this document, there will describe system requirements, the functionalities of the program and how to use it.

# System Requirements

To run the application, the user must have:

* Windows 7 or newer installed
* Visual Studio.NET 2012 or higher
* SQL Server 2014 or higher
* Minimum 50MB free space on the hard disk

# Use Cases

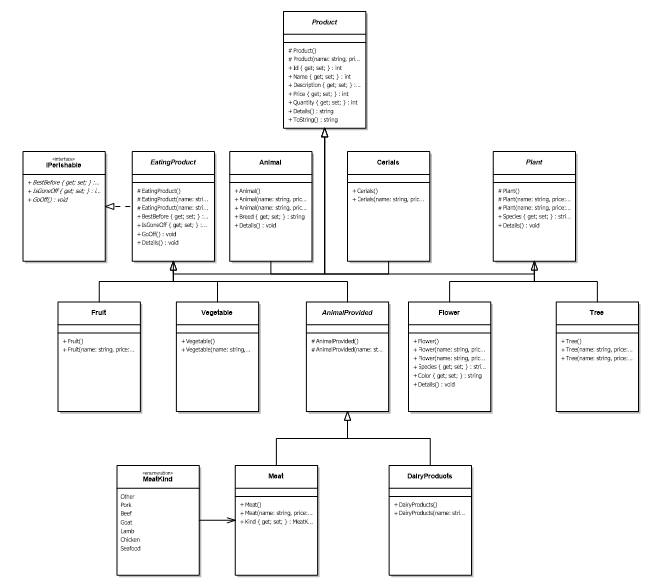
AgriIPCA is a console application for recording and storing information about users, for recoding and storing information about products and for sorting and buying products by users. In this part of the document, a list its main functionalities and explanation about them can be found. To be more understandable, the functionalities are divided in two groups: used by administrator and used by user.

* Used by users
* Create and login – not registered users can create a profile and login to the system
* Details of a profile – a register user can check their own profile and edit it
* List products – only the register users can see a list of all products in the warehouse
* Filter products – if a user wants a specific type of products, he can filter
* Sort products – this function help the registered users to find the cheapest/the most expensive products
* Buy a product – registered users can buy a product and see the total price of their purchase
* Top 10 products – users can see their most favorite products
* Product details – users can find more detailed information about a specific product
* Used by administrator
* Create new products – administrators can add new products to the data
* Edit an existing product – administrators can edit the name/price/quantity of the products
* Delete product – administrators can delete an existing product
* List all the current users – administrators can see all the registered users
* Delete a user – administrators can delete one/more registered users
* User details – administrators can view the details of the registered users

Note that administrators can also act as ordinary users.

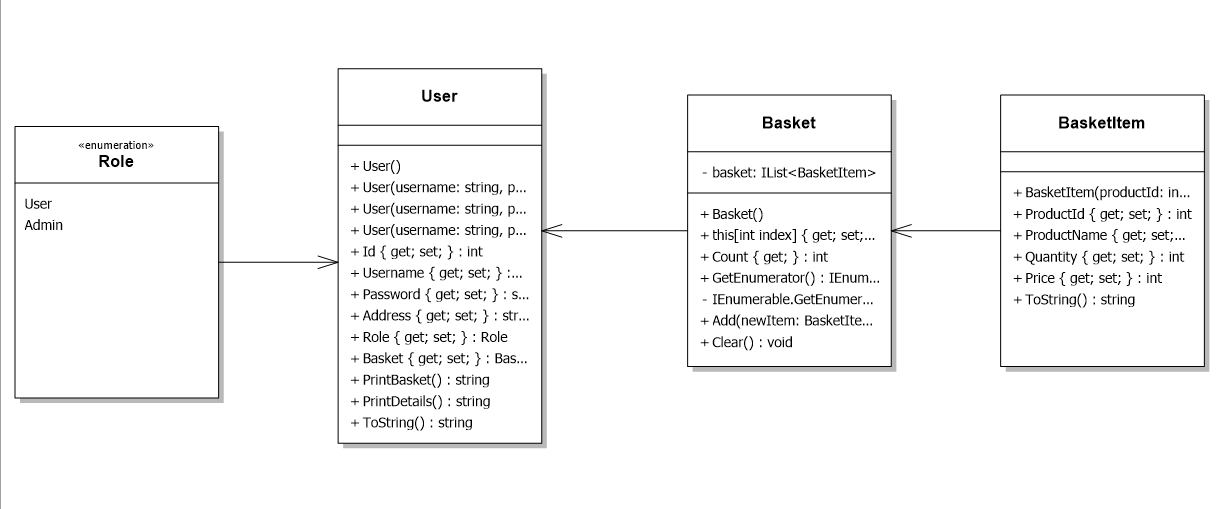
# Class Diagram

The application can be generally presented in two main diagrams, one is about the products and the other one is about the users.



The abstract class Product to determine the characteristics of all other classes related to the products. The products (class Product) are divided in 4 main groups:

1. EatingProduct – all types of eating products have a characteristic of perishability (IPerishable). The eating products are divided in 3 main categories: fruits, vegetables and animal provided (meat and dairy products). The separation is done in this way because the fruits and vegetables are the most popular products in our warehouse. MeatKind enumerations is responsible for variability of meat
2. Animal
3. Cereals
4. Plat – the property Species is shared between all the plants which are flowers and trees



The class User presents the main user of the application. There are 2 types of users: ordinary user and an administrator, this characteristic come from the enumerable Role.

Every user has a basket of products he wants to buy. To present an item from the basket, the class BasketItem is used.

Basket implements the interface IEnumberable, which give it the ability to behave as a list.

# User Guide

When the user starts the application, he goes to the menu for not registered users. There he has only 3 options: to create a new account, to login with existing account and to exit the applications.

To create a new account, the user need to fill in a form with his username, password and address. To login, he needs to provide correct username and password.

After successful login, user goes to the main menu of the application:

1. List Products
2. Buy Products
3. Profile Details
4. Admin
5. Help
6. Log out
7. Exit

*List products* – this only provides a list of products in format “[id];[product name];[quantity];[price]”

*Buy products* – this again prints all the available products for the user and then asks for the code(id) of the wanted product, the wanted quantity. Then, it creates a new item, add it to the basket and print the product from the basket. The user has the option to add another product to his basket. If he is ready with his purchase, the details of the purchase are printed, and the program asks for the user’s confirmation. If he enters “N” or “n”, he can add another product to his basket. If enters “Y” or “y”, the purchase is successful, and the items are removed from the database.

*Profile details* – the user can see its profile and edit it from here. All the information about the user is overwritten in the database

*Admin* – this function is available only for the administration users. If a user is not admin, he cannot access the function. When an admin user access it, an admin menu is presented. For it, the user can see all the products in the database or see all the registered users.

Note that there are some hidden commands. To see some help about these commands, go to the help or write “help” in the console. The help looks like this:

* + **'list products'** or **'1'** to see a list of all products
  + **'list users'** or **'2'** to see a list of all users
  + **'create [type of product]'** to create a product
  + **'edit product [product id]'** to edit a product
  + **'delete product [product id]'** to delete a product
  + **‘details product [product id]’** to see a detailed information about a product
  + **'create user'** to create a new user
  + **'edit user [username]**' to edit a user
  + **'delete user [username]**' to delete a user
  + **‘details user [username]’** to view user’s profile

For example, if the user wants to see *all the products*. He can do it from the admin menu or just enter “list products”, respectively, it is valid for *the users*.

To *create a new product*, he needs to enter “create” plus the type of product. The currently available types of products are:

* Animal
* Cereals
* Dairy
* Flower
* Fruit
* Meat
* Tree
* Vegetable

To *edit a product*, the administrator should enter “edit product” plus its id.

To *delete a product*, the administrator should enter “delete product” plus its id.

To see *detailed information* about a product, the administrator should enter “details product” plus its id.

To *create a user,* the administrator should enter “create user”.

To *edit a user,* the administrator should enter “edit user” plus its username.

To *delete a user,* the administrator should enter “delete user” plus its username.

To see *users’ profile,* the administrator should enter “details user” plus its username.

*Help* – provides a similar menu to the administrator’s help menu. It contains:

* **'product [product id]'** to see its details
* **'sort [asc or desc]'** to show products in ascending or descending order, depending on the price
* **'top10'** to show top buying products
* **'empty basket'** to empty the current user’s basket

*Log out* – this function gives the ability to user to logout of the system. If he closes the applications the same is valid.

*Exit* - closes the application

# Database

The project uses a SQL database. The approach that is implemented is Code First. Firstly, which means that from the written code the database is generated. To make this possible Entity Framework 6 is used.

The code first approach that is used is called Table Per Type (TPT). Compared to the other approaches, this is more object-oriented. It provides more flexibility because new subtypes can easily be added by creating new tables for those subtypes.

This is the scheme of the database:

