National Research University
Higher School of Economics

Faculty of Business and Management

# Team project on course «Programming»

# «Restorizer»

**Completed by:**
*students of group BBI165*
Zakharov Ivan
Neklyudov Boris
*student of group BBI163*
Gromov Ilya

**Instructor:**
Efremov S.G.

Moscow, 2017

## CONTENTS

## GENERAL INFORMATION

**«Restorizer»** - your all-in-one tool for managing the kitchen and finances of your cafe or a restaurant business. Includes all of your recipes, ingredient purchases and estimated future expenses.

## ANNOTATION

The main aim of the application is being all-in-one helping tool for the restaurant business holders and staff. The application provides user with the basic CRUD operations over the main restaurant business entities as well as the dashboard for reviewing the statistics. The features include: registering the customers' orders, managing the dish recipes and ingredient supplies with pricing and amounts. The statistics come in the visualized or numeric way in the special section of the application called «DashBoard». The used technologies include: Entity Framework, NewtonSoft, LiveCharts.

## CENTRAL REPOSITORY ADDRESS

The link to the GitHub repository: https://github.com/ivan339339/Restorizer

## MEMBERS OF THE TEAM AND THEIR ROLES

The roles of the team remained almost the same as stated in Google Tables with minor adjustments due to improving the development process.

**Ivan Zakharov (BBI-165):**

WPF design and template prototyping, API implementation and logic

**Boris Neklyudov (BBI-165):**

Data visualization, LINQ queries, statistics logic, app testing

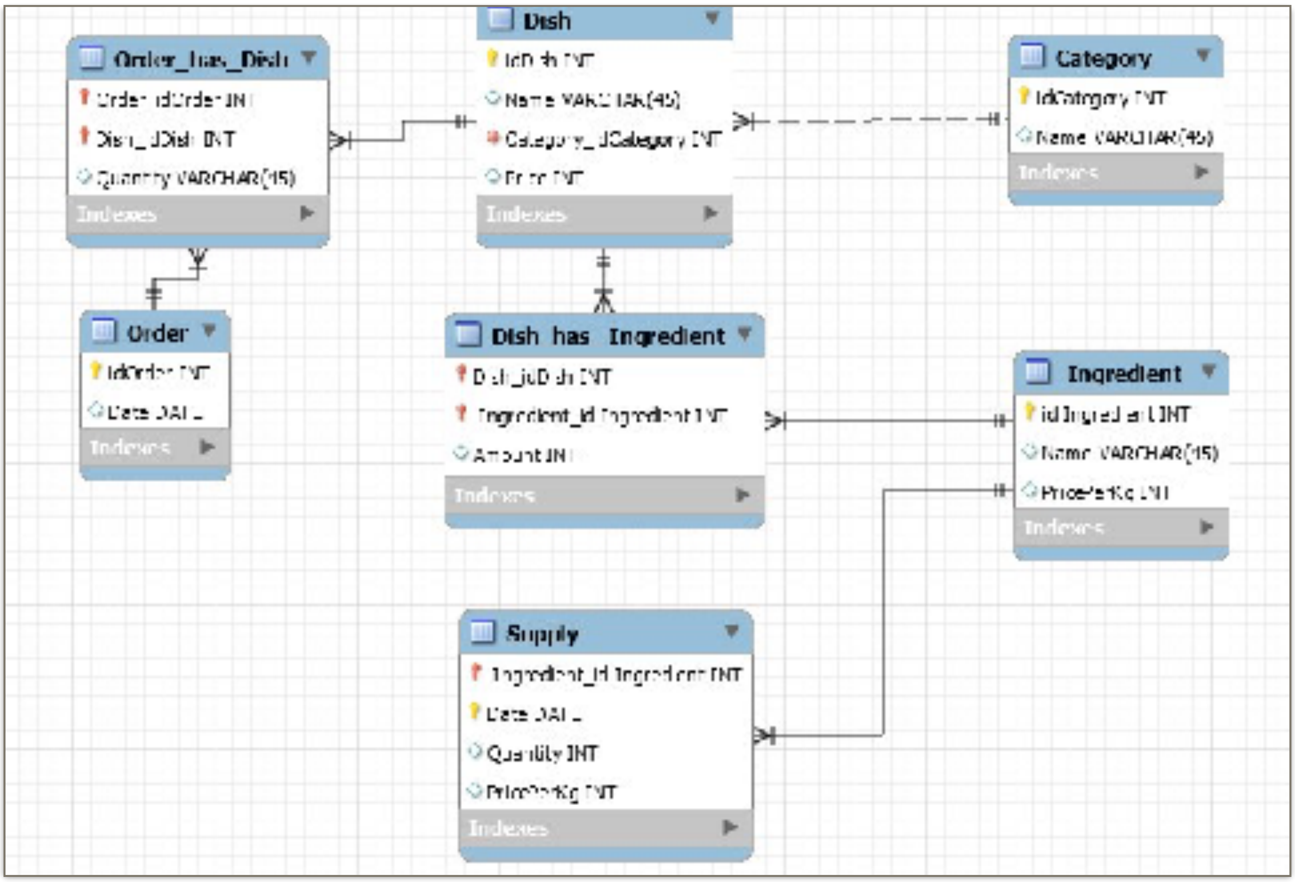
**Ilya Gromov (BBI-163):**

Database design and app architecture, CRUD/backend logic

## DESCRIPTION OF CLASSES

### MODEL CLASSES

First of all, we developed EER database model to rely on while developing the EF database using code-first approach.

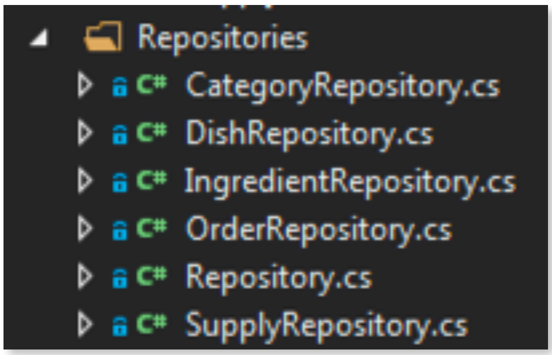


**Class Order** is used for registering the fact of ordering the dish(-es).
**Class Dish** is used to store the information about the dish category and the ingredients and the amounts of them needed to prepare the dish.
**Class Category** is used for dividing dishes into logical segments.
**Class Ingredient** is used to describe the ingredient by its name and storing the current price of the ingredient for buying.
**Class Supply** is used for storing the history of ingredient supplies.

Associative tables **OrderHasDish** and **DishHasIngredient** are used for implementing many-to-many connections.

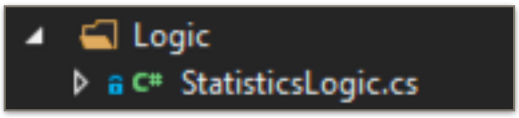
## REPOSITORY CLASSES

There is a number of **Repository classes**, each corresponding to its model entity. There is also a **Repository** parent **class**, which contains repeating pieces of code.



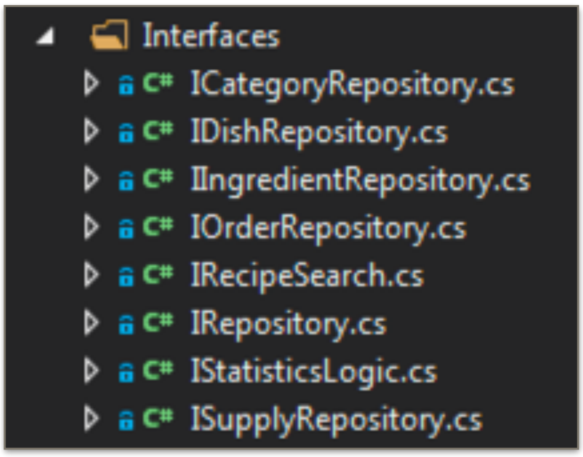
## DEDICATED LOGIC CLASS

The **StatisticsLogic** class is used for extracting the logic into the separate class (accord. to single responsibility principle)



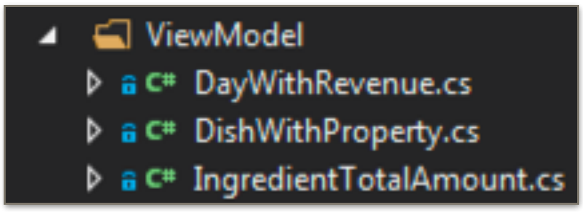
## INTERFACES

The **Interfaces** are implemented for abstract realization of UnitOfWork pattern. There are **I[Entity]Repositories** responsible for Repository abstractions, they all have the parent class **IRepository**. There is an **IStatisticsLogic** for the StatisticLogic and **IRecipeSearch** for loose coupling with the preferred search service.



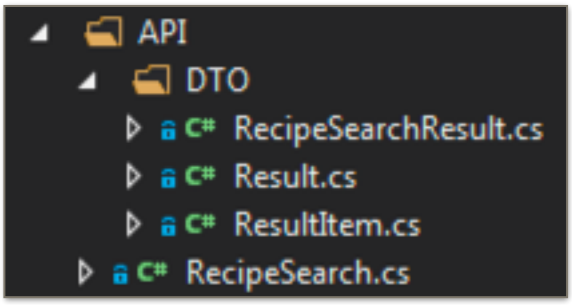
## VIEW MODEL CLASSES

**ViewModel** classes are the special classes used for convenient representation of data in the interface.



## API RELATED CLASSES

**RecipeSeacrh** class realizes the API service and implements the IRecipeSearch interface. **DTO classes** are used for translation the JSON responses from API service into program entities.



## GUI CLASSES

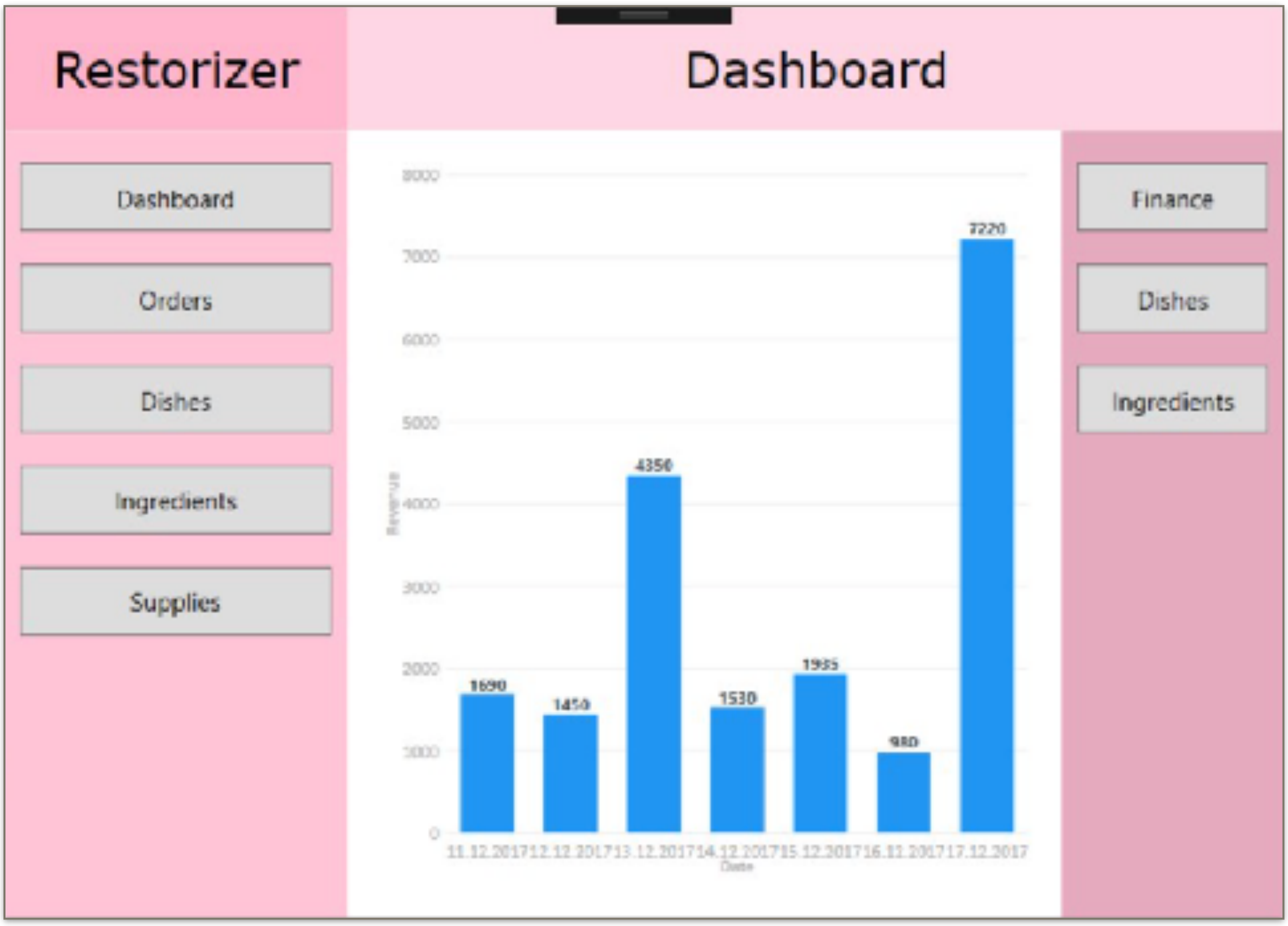
The **WPF page and window classes** are stored in the corresponding folders. The **PagesFactory** class is used for storing pages, which should stay as single instance in the program. **ISectionPage** interface is used to require the section pages to have Heading property.

## OTHER CLASSES

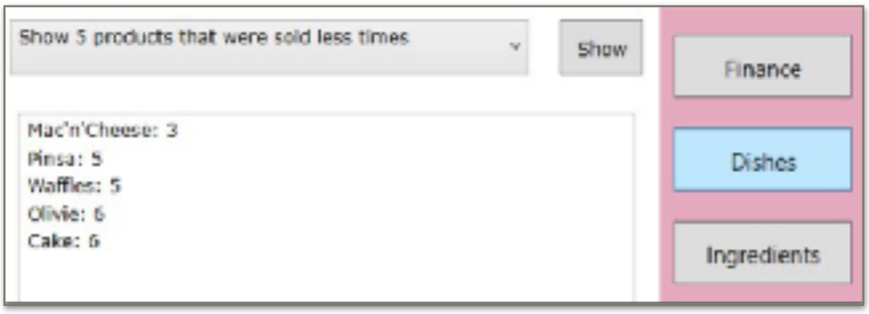
**Context class** is used for the EF implementation and **UnitOfWork class** is used for the implementation of Unit Of Work pattern in the program.

## PROGRAM INTERFACE

### THE DASHBOARD PAGES



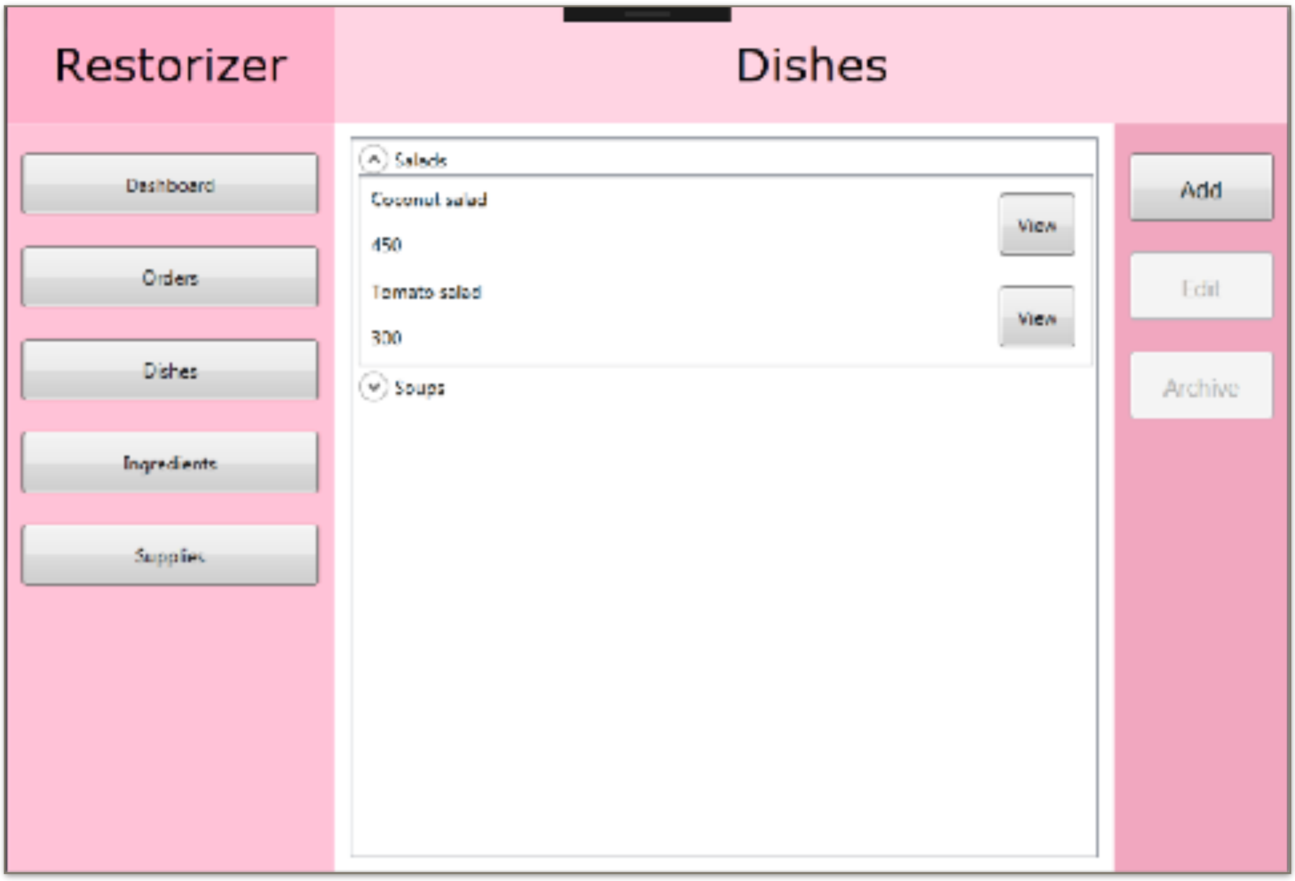
On the screenshot above is the visualized data report completed with LiveCharts WPF library. In the example, the visualized data is the amount of revenue (y-axis) by days (x-axis).



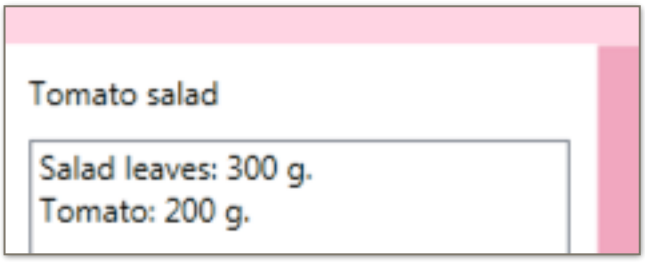
Mac'n'Cheese: 3
Pinsa: 5
Waffles: 5
Olivie: 6
Cake: 6

Another form of dashboard data presentation is to the right. The desired report is chosen in ComboBox and showed with the press of the Show button.

## THE EXAMPLE OF SECTION PAGE

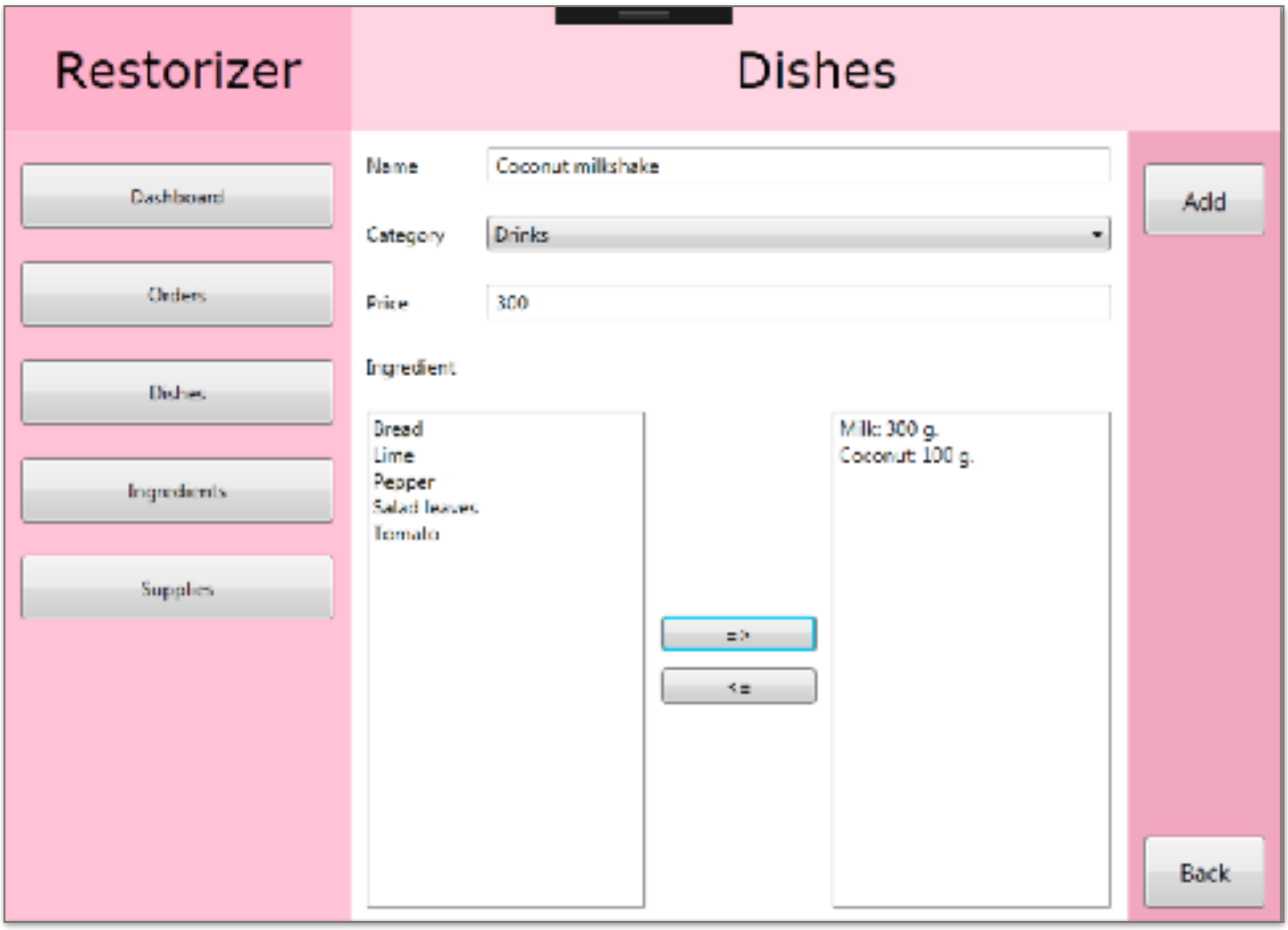


The section pages provide users with the convenient view of data and intuitive controls of CRUD actions. The recipe for the ingredient can be viewed with the press of a button, opening in the new page. The recipe ListBox example is depicted to the right.


Tomato salad

Salad leaves: 300 g.
Tomato: 200 g.

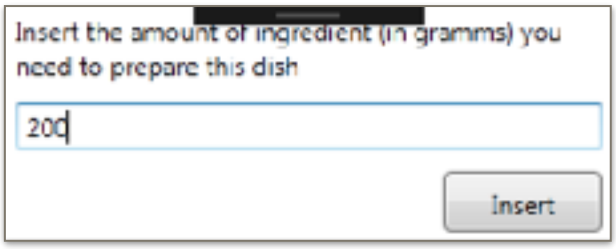
## THE EXAMPLES OF ADD PAGES



Name: Coconut milkshake
Category: Drinks
Price: 300
Ingredient

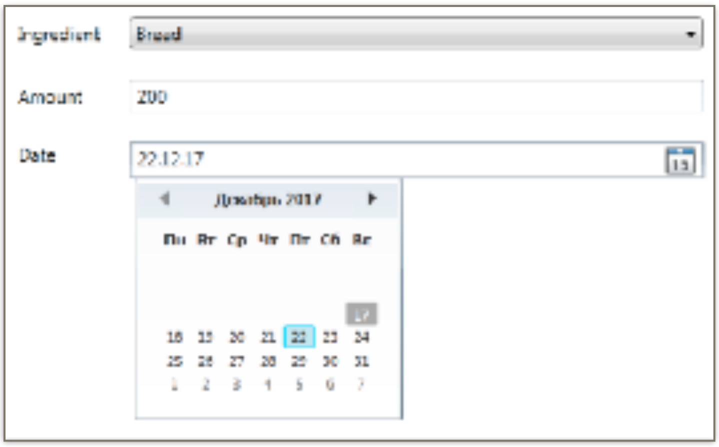
Bread
Lime
Pepper
Salad leaves
Tomato

Milk: 300 g.
Coconut: 100 g.


Insert the amount of ingredient (in gramms) you need to prepare this dish

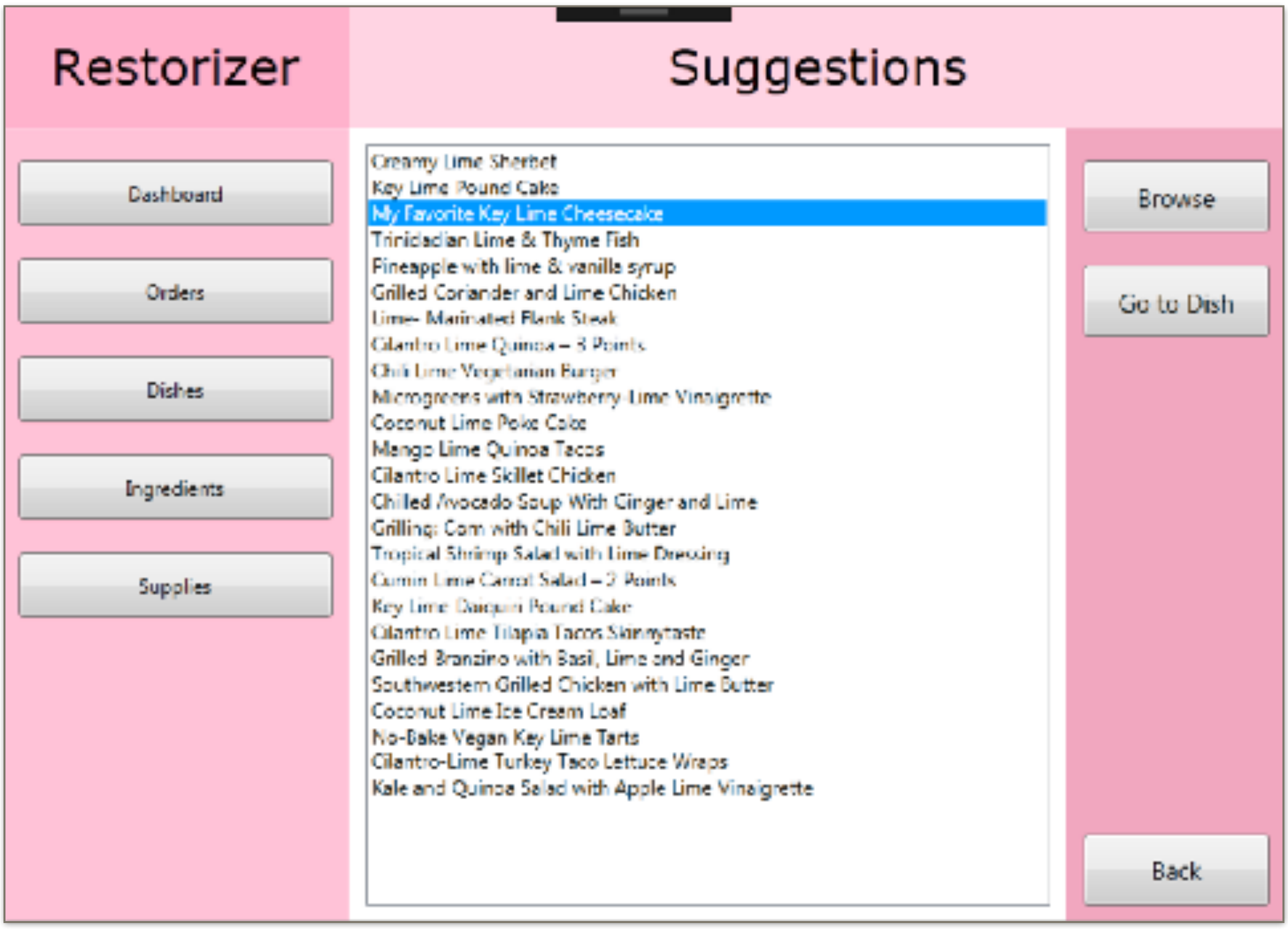
200

The convenient GUI for adding the entity, which combines ComboBox, ListBox, TextBox controls. The categories are considered static and are hardcoded. The additional information to many-to-many relationships is entered through supporting windows (depicted above).


Ingredient: Bread
Amount: 200
Date: 22.12.17

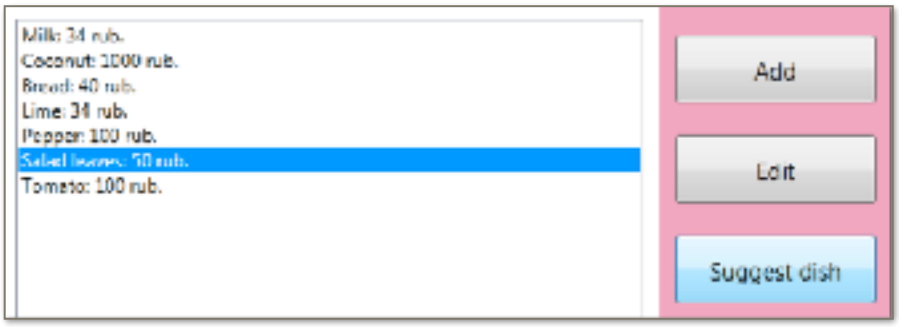
Some additional controls are used where necessary, e.g. DatePicker in the supply addition page.

## THE API IMPLEMENTATION PAGE

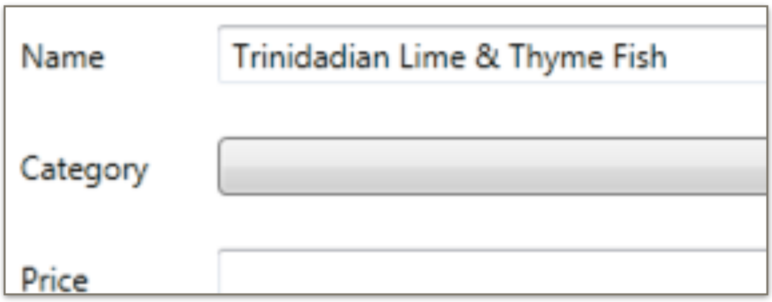


Creamy Lime Sherbet
Key Lime Pound Cake
My Favorite Key Lime Cheesecake
Trinidadian Lime & Thyme Fish
Pineapple with lime & vanille syrup
Grilled Coriander and Lime Chicken
Lime- Marinated Flank Steak
Cilantro Lime Quinoa – 3 Points
Chili Lime Vegetarian Burger
Microgreens with Strawberry-Lime Vinaigrette
Coconut Lime Poke Cake
Mango Lime Quinoa Tacos
Cilantro Lime Skillet Chicken
Chilled Avocado Soup With Ginger and Lime
Grilling: Corn with Chili Lime Butter
Tropical Shrimp Salad with Lime Dressing
Cumin Lime Carrot Salad – 2 Points
Key Lime Daiquiri Pound Cake
Cilantro Lime Tilapia Tacos Skinnytaste
Grilled Branzino with Basil, Lime and Ginger
Southwestern Grilled Chicken with Lime Butter
Coconut Lime Ice Cream Loaf
No-Bake Vegan Key Lime Tarts
Cilantro-Lime Turkey Taco Lettuce Wraps
Kale and Quinoa Salad with Apple Lime Vinaigrette

On the screen above there is page with a ListBox showing results of API recipes' search, which can be entered from Ingredients page.



Milk: 34 rub.
Coconut: 1000 rub.
Bread: 40 rub.
Lime: 34 rub.
Pepper: 100 rub.
Salad leaves: 50 rub.
Tomato: 100 rub.



| | |
|---|---|
| Name | Trinidadian Lime & Thyme Fish |
| Category | |
| Price | |

The user then can either examine the recipe for the suggested dish online or instantly move on to creating the chosen dish.

The name of suggested dish is preloaded for the user.

# TEST CASES

Here are several cases that required thorough examination and testing/debugging:

**Deletion of dishes:** the connection between the Dish and the Order entities is of the many-to-many type, therefore come certain restrictions on deleting the dishes. The reason for this is, that if the dish was deleted from the database and, consequently from the connection table, the calculations of statistics connected to ordered dishes would be misleading and wrong. The solution was to keep all the dishes in the database and to introduce the flag «IsArchived» to restrict the usage of certain dish in program, but to allow the statistics logic to receive data about it.

**Displaying the dishes' ingredients:** one of key functions of the application is providing user with dishes' recipes, so the malfunctioning of this aspect is a big problem in our application. While implementing the feature in the Dishes section, no bugs were found, but there was a problem with it in the Orders section - no ingredients were present in the recipe. The solution was to tweak the corresponding LINQ query and to extend it with the missing Include( ) methods.

**Getting suggestion for ingredients:** the user is provided with the ability to find dish ideas for their restaurant in the web via the recipe search API, hence there are a lot of probable pitfalls linked to web services' usage. The test case was to check the feature with no internet connection, which resulted in the unhandled exception. It was a vital program malfunction, therefore we came up with solution of wrapping the insecure code in the 'try' block and later catching the exception.