

# CafeGuide App

## Annotation

*CafeGuide* is an application that helps user to find a cafe or a restaurant that suits his or her request. Search parameters are: Location, Type (bar, restaurant, etc.), Types of cuisine provided by a restaurant and so on. The information about cafes is partly stored in a database (e.g. Name, Address, Average Check) and partly received from the Web with the help of Google Directions and Google Places API. Database is created via MSSQL Server. A café may be also found by its name, if information about it is stored in the DB. UI - WPF.

## Repository

<https://github.com/sabitova/CafeGuide/tree/test>

## Team Members

Sabitova Svetlana 143 - DB, app logic

Mikhailova Valeriya 143 - API, UI, app logic

## Classes description

*Café.cs* – a class that contains the information (fields: Name, Address, Type, Cuisine Type, etc.) about objects that are considered to be cafes.

*Address.cs* – a class that contains the information about addresses (fields: Latitude, Longitude, Place\_Id that is required by Google API)

*Type.cs* – a class that contains the information about existing types of cafes (e.g. Restaurant, Bar)

*Cuisine.cs* – a class that contains the information about existing types of cuisines (e.g. Italian, Japanese)

*APIDirection.cs* – a class that contains logic for working with Google Directions API

Folder *DTO* contains classes: *Distance.cs*, *Duration.cs*, *Geocoded\_waypoints.cs*, *Legs.cs*, *Response.cs*, *Routes.cs*, which are needed for deserializing Json object returned by Directions API

*APIPlaces.cs* – a class that contains logic for working with Google Places API

Folder *DTO2* contains classes: *Response.cs*, *Result.cs*, *Reviews.cs*, which are needed for deserializing Json object returned by Places API

*IProcessing* is an interface that contains methods that have to be implemented by classes that inherit from it. These classes are:

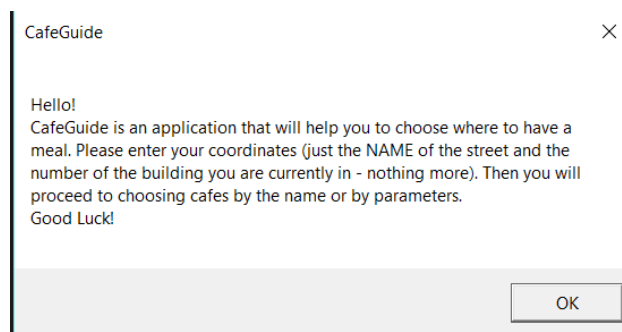
*RepoProcessing.cs* – a class that contains methods for processing data stored in a repository.

*DBProcessing.cs* – a class that contains methods for processing data stored in a DB

The information about classes that contain logic for working with windows will be provided below.

## Using the application

The execution process starts with appearing of a MessageBox that provides some information about the app.



When the Box is closed, the StartWindow appears.



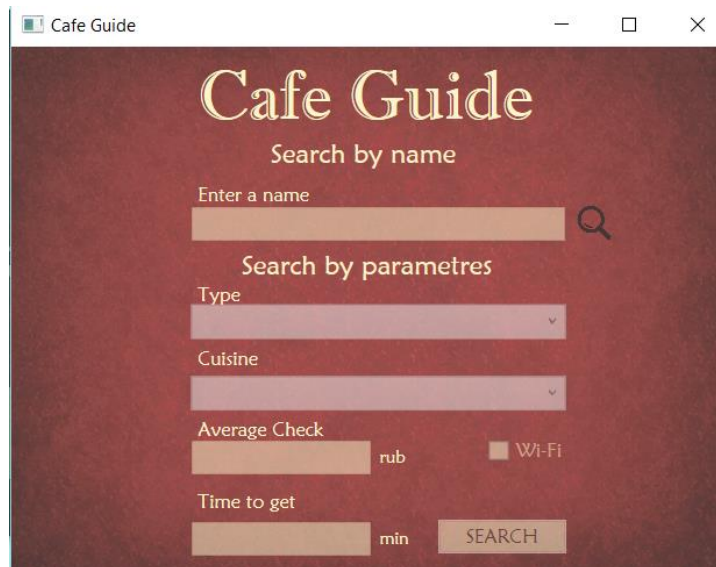
The image shows a window titled "Cafe Guide" with a dark red background. At the top, the title "Cafe Guide" is written in a large, gold, serif font. Below the title, the text "Enter your location" is centered in a smaller, gold font. Underneath, there are two input fields: "Street" on the left and "House" on the right, both with gold text and empty text boxes. Below these fields, the text "Choose a means of transport" is centered in a gold font. At the bottom, there are three buttons: "By car", "On foot", and "By public transport", each in a gold font on a dark red background.

Here user can choose his current location and the means of transport he wants to use. *StartWindow.cs* contains methods that read the information that was input by user and use it to calculate time which is needed to get to known cafes.

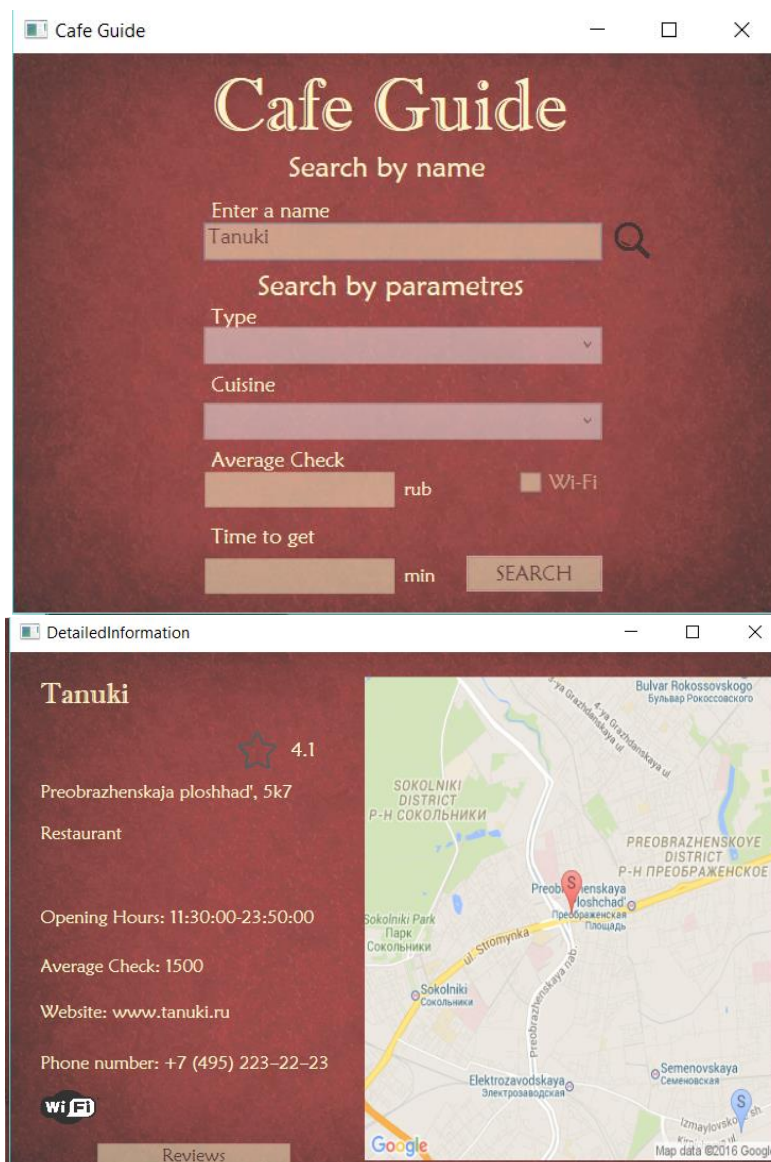


The image shows the same "Cafe Guide" window, but now the input fields are filled with example data. The "Street" field contains the text "Kírpíchnaya" and the "House" field contains the number "33". The "By car" button is highlighted with a lighter, purpleish-red background, while the other two buttons remain dark red. The rest of the interface, including the title and transport options, is identical to the previous image.

After the moment when one of the three buttons below was pressed, appears the *MainWindow*. Here user can input information to find suitable cafes either by name or by other parameters.

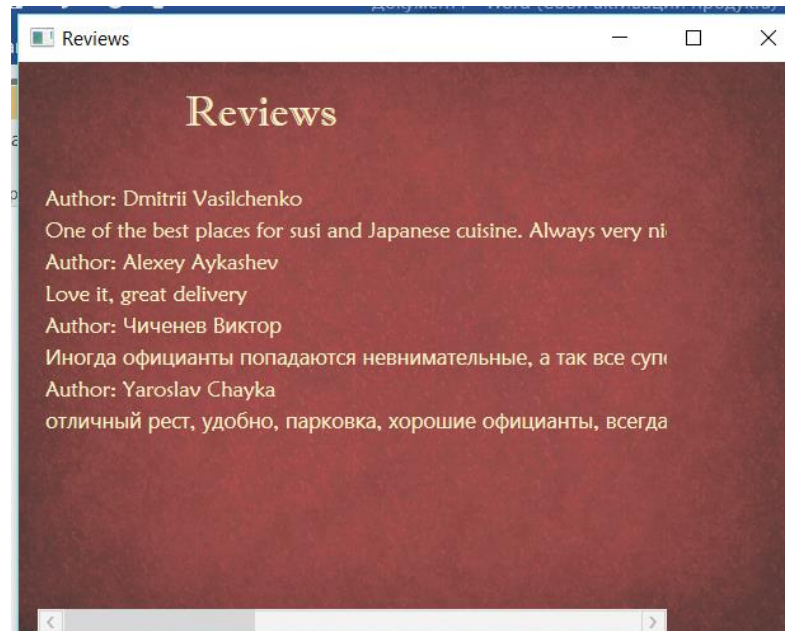


*MainWindow.cs* contains methods that read the information from WPF elements and passes to further methods.

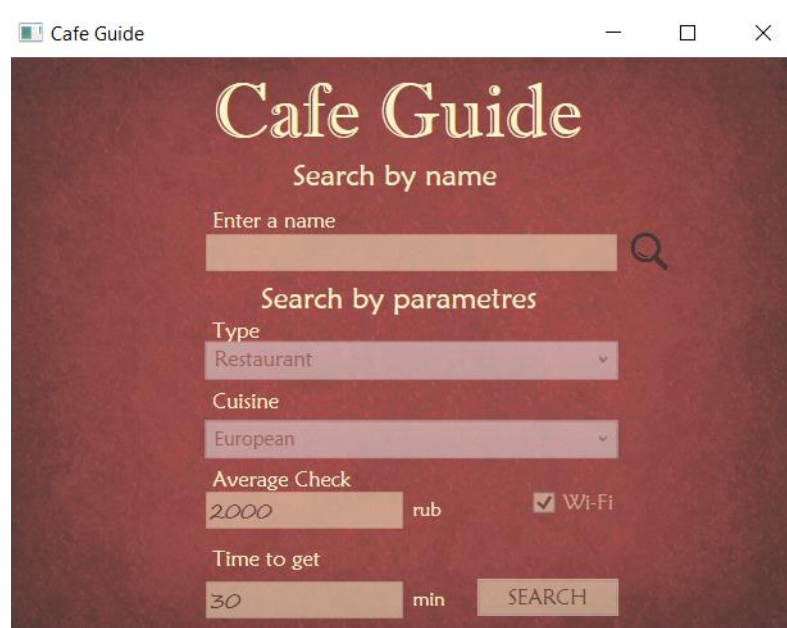


That is how search by name works.

A window with detailed information appears. *DetailedInformation.cs* contains logic mostly for filling it with content. User may press the Reviews button to see reviews that are downloaded from Web.



If user chooses to search by parameters, the interface looks like following.






Name	Time (min)	Average Check (rub)
Il Patio	30	1000
Temple Bar	26	1500
TGI Fridays	30	1000

A window with detailed information may be opened by clicking on a row corresponding to an item.

### Il Patio

 3.9

Zemlyanoy Val ul., 33


Restaurant

Opening Hours: 12:00:00-23:50:00

Average Check: 1000

Website: [il-patio.rosinter.ru](http://il-patio.rosinter.ru)

Phone number: +7 (495) 788-44-88



Reviews

