**ОБРАЗЦОВА МАТЕМАТИЧЕСКА ГИМНАЗИЯ ”АКАД. КИРИЛ ПОПОВ“ - ПЛОВДИВ**

4001 Пловдив, ул.”Чемшир” № 11, тел.:032/643 157; 032/643 093; тел./факс: 032/643 192; e-mail: [omg\_plovdiv@abv.bg](mailto:omg_plovdiv@abv.bg)

НАЦИОНАЛНАТА ОЛИМПИАДА ПО ИНФОРМАЦИОННИ ТЕХНОЛОГИИ

**ПРЕЗ 2016/2017 УЧЕБНА ГОДИНА**

# Name

BG TAXI – management of taxi companies

# Authors

**AUTHOR:**

Petar Svetovlavov Bechev,

GSM: 0896 314 939, e-mail: bechev2@gmail.com

High school of mathematics “Acad. Kiril Popov”, Plovdiv

**LEADER:**

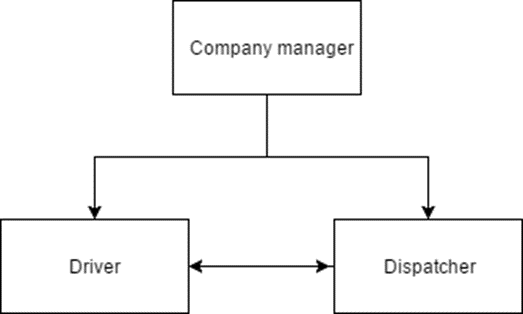
Darina Bruncheva, 0888 992 447, e-mail: dary\_brancheva@abv.bg

Senior teacher of informatics and IT in the high school

# Резюме

## 3.1. цели

BG TAXI is an Internet application aimed to make the management of taxi companies easier and also to replace the use of radio stations with a more effective way. It consists of a web portal bgtaxi.net and a mobile application for taxi drivers. The website is used for the management of all user roles – administrator (employer), dispatcher and driver. The employer possesses a company and the employees work there. As dispatchers log in their profiles they can send requests to drivers from the company. The driver app intent to visualize the request to the driver and what’s more the app sends request to the server every second with the current location and that way the server decides which request are appropriate for the car. It is given higher priority to those cars which are closer to the location of the request.



## 3.2. Main stages

During the implementing of the project I did several things

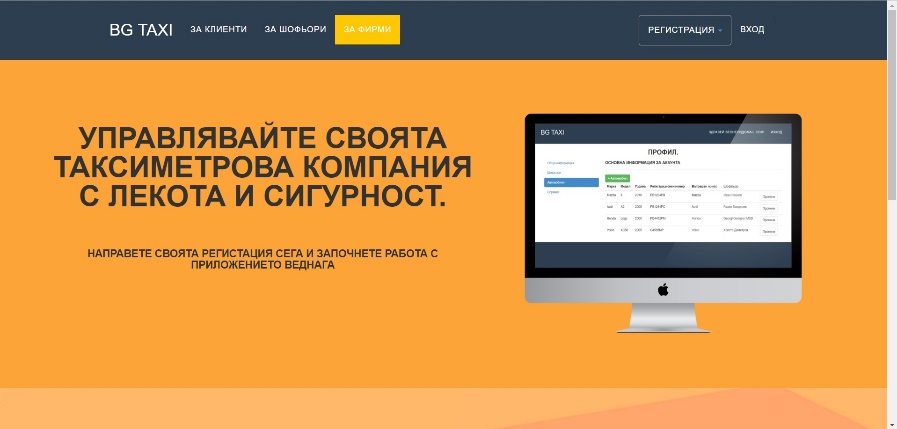
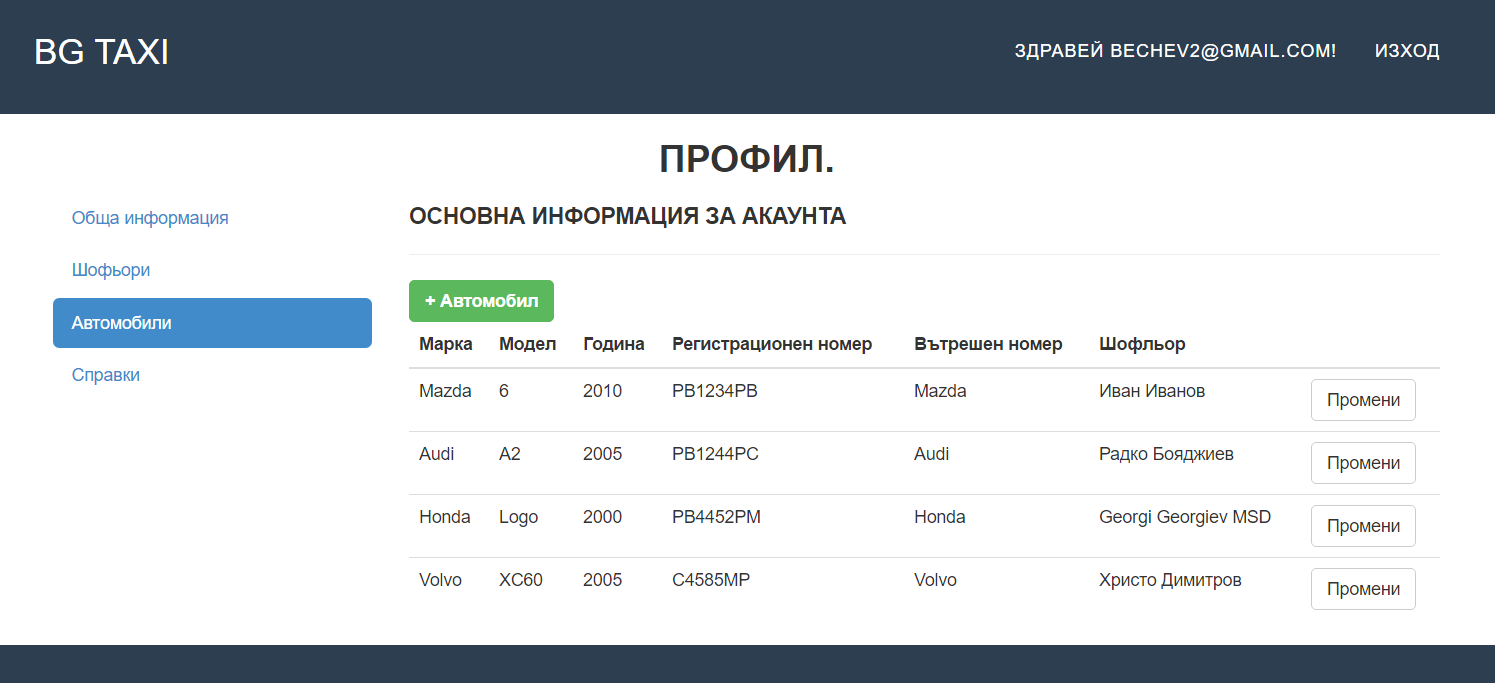
1. Considering the structure of the project
2. Building Data layer – the database
3. Designing the server side
4. Building the mobile application for drivers
5. Implementing the administration
6. Make appropriate design and navigation
7. Implementing the user interface
8. Testing and documenting

## 3.3. Ниво на сложност

During the implementation of the project I dealt with the following problems

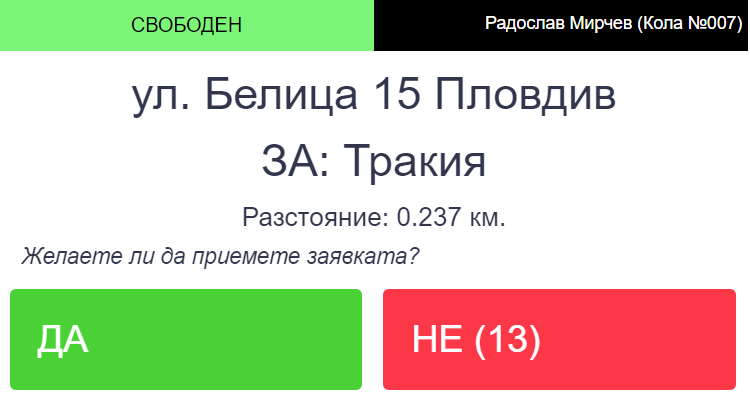
1. Connecting all different parts into one finished project
2. Choosing best technology for developing the project
3. Creating a mobile app that can be run on the most popular mobile platforms: Android, IOS и Windows Phone
4. Following best practices during the development

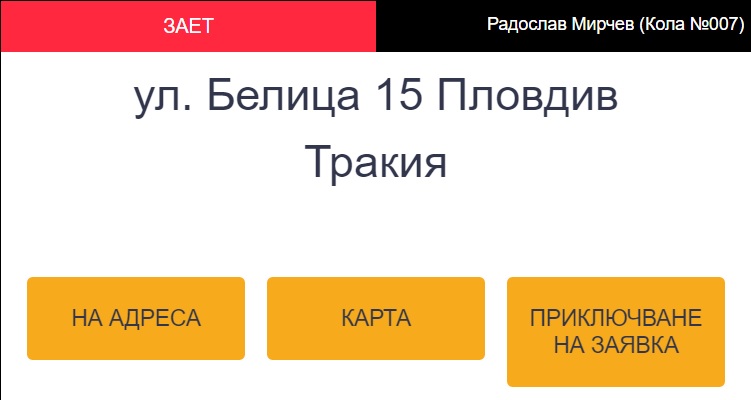
## 3.4. Description of the project

The website bgtaxi.net is the administration side. That is the place where every employer of each company can manage their businesses. After they make a registration as a company they receive an email with a link for activation and a unique code for the company. Every employee need to enter this code when they register. That is the way every user is identified as an employee of the particular company. In profile of the employer there are four fields: "General Information", "Drivers", "Dispatcher", "Cars." In the "General Information" section manager can change information for the current company. In "Drivers" section is shown data about all drivers entered the unique code of the company. The manager also removes drivers with "Remove" button positioned in the right side of each row of the table. In "Dispatchers" section is shown data about all drivers entered the unique code of the company. The manager also removes drivers with "Remove" button positioned in the right side of each row of the table. In the “Cars” section manager can give each driver a car which is already registered or can create a new one from “+ Car” button. Each already created car can be modified and viewed by pressing “Change” button. Driver who has a car cannot be chosen for a driver of another car. Another important thing is that driver without a car cannot receive requests from the dispatchers. That is why the administration side is so important and significant for the driver mobile application. Every dispatcher in their profile has basic setting in the main page. Also “Dashboard” section is the place where the requests to drivers can be sent. There is a map on which are placed colourful markers updated real time. Colours shows the current status of the cars, for example the green means the car is free, the red shows it busy, etc. There is moreover a table displaying currently sent requests and also their status and all information needed. Above the table there is a form for sending new requests where it is obligatory to write the “Starting address” and the “Finish address”. The whole page updates itself and no refreshing is needed. That what makes it so fast and convenient as was planned in the beginning of the project. Let’s explain vary briefly how the server works out with the requests. Ones the requests is sent the server chooses the closest car in the area as it takes into account the current status of it and obviously it should be free. The request is sent to the driver and they have 15 seconds to answer with “Yes” or “No” and after the time is up the request is resent to the second closest car. If it answers “Yes” on the dispatcher’s dashboard appears the number of the car and the approximate time calculated by Google Maps Distance Matrix API.

**Company:** [**bechev2@gmail.com**](mailto:bechev2@gmail.com) **pass: 123456**

**Unique code of the company: IOE0ZV**

For using the driver’s app registration is also obligatory however it can only be made from bgtaxi.net. There are a few things that need to be set up before the drivers can start using the application. First the email must be confirmed, they need to be employees of a company and they must have set a car that they will be using. Unless all of these three things are made, the driver will not be able to use the app. When they log in successfully in the app the current application starts to be sent every few seconds to the server. In the menu there is a button “Map” or “Logout” which makes the car status “Offduty”. As long as they receive a request if is opened on a new screen and there is piece of information about the addresses and also the options to accept it or deny it. There is a time limit of 15 seconds and after the time is up the request is sent to another driver. Accepted request are visualized on a third screed where the driver has access to the map and can signalize that they are on the address (most of the cases that happens automatically when the server calculates that the car is located on 20 m radius of the address). As soon as the car status is busy, other requests cannot be sent to this car. When the request is marked as finished the car status changes into free and now other request can be sent and accepted. MISSING status is added as a feature and that is how driver can stop receiving requests for some minutes if they need a break for example.



**Driver:** [**pbechev@schoolmath.eu**](mailto:pbechev@schoolmath.eu)  **pass: 123456**

**Dispatcher:** [**sv\_bechev@abv.bg**](mailto:sv_bechev@abv.bg) **pass: 123456**

## 3.5. Реализация

The project is successfully implemented thanks to the following technology:

* ASP.NET MVC – platform used for developing web application using the Model-View-Controller (MVC). It allows implementing the administration side as well as the web service. On sending a HTTP request the server returns s response in a JSON format which is presented in an appropriate way to the user
* Telerik Platform – web-based platform for creating a hybrid mobile apps using HTML, CSS, JAVASCRIPT, supporting the three most popular mobile OS .
* Google Console API – web services which returns specific type of data in a JSON for example the distance between two points.
* SQL Database – for data layer is used SQL database and the Entity Framework Code First

Used libraries (all open source) :

jQuery – DOM манипулиция

Bootstrap – front-end framework за дизaйн

Ninject- open source dependency injector

Knockout.js – library for creating single-page apps

The system is secured from the most popular attacks:

SQL Injection, Parameter tampering, XSS attack (cross site scripting), CSRF (Cross site request forgery), etc

Driver’s mobile application :

Android – <http://bgtaxi.net/download/driver/apk>

Windows Phone - <http://bgtaxi.net/download/driver/xap>

## 3.6. references

I have used these sources of information:

Introduction to programming with C# - <http://www.introprogramming.info/> - Nakov

<http://telerikacademy.com/> - online courses of C#, ASP.NET MVC;

<http://www.w3schools.com/> - HTML, JAVASCRIPT, CSS, BOOTSTRAP