

Explanatory note

for the team project

«Telegram Bot: HSE Route Bot»

Team members:

*Bulgakov Dmitry, 145(1)*

*Rogova Daria, 145(2)*

*Trofimenko Anastasia, 145(1)*

Moscow, 2016

1. *Topic of the project*

Our project is Telegram Bot: HSERouteBot

1. *Annotation*

Our Telegram Bot finds all possible routes from one of HSE dormitories to one of HSE buildings and calculates required time. It creates a step-by-step instruction, which includes departure time and points of surface and subway transport that should be taken.

1. *Central repository address*

[*https://github.com/dbulgakov/HSE\_transport\_manager*](https://github.com/dbulgakov/HSE_transport_manager)

1. *Members of the team and their roles*

Bulgakov Dmitry – API, application logic

Rogova Daria – database, application logic

Trofimenko Anastasia – UI, unit tests

1. *List of classes with short description for each class*

**GoogleGeoMatrixService**

DTO:

* EstimateTimeResponse
* ResponseElement
* ResponseRow
* TripDuration
* GoogleGeoMatrixService (sends a request and receives a JSON response)
* RequestBuilder (creates API request)

**HSE\_transport\_manager**

* MainWindow.xaml.cs (contains 3 tabs: Settings, Status, About)
* AboutView.xaml.cs (shows information about the bot, its authors and used services)
* SettingsView.xaml.cs (receives API keys, updates or removes train schedule)
* StatusView.xaml.cs (shows connection status, starts or cancels the program)
* AboutViewModel.cs
* MainViewModel.cs (contains commands for changing the current view)
* SettingsViewModel.cs (contains methods, which allow to save or reset API keys, to update or remove local train schedule)
* StatusViewModel.cs (contains methods, which allow to start the bot or
* ViewModelLocator.cs
* PluginManager.cs (loads .dll files)

**-**WpfMessageProvider.cs (manage WPF notifications)

**HSE\_transport\_manager.Common**

* TaxiTripData.cs (describes an object of a taxi ride by duration, price, departure and destination points)
* DailyTrainSchedule.cs (describes an object of a daily schedule by list of trains schedules, departure and destination points)
* SingleTrainSchedule.cs (describes an object of a train schedule bu list of stops and departure time)
* TrainStop.cs (describes an object of a train stop by station code, arrival and elapsed time)
* Coordinate.cs (describes an object of geographical coordinates)
* KeyData (describes an object of data, which contains API keys)

**MSDatabaseService**

* DayofWeek.cs (describes an object of each day of the week by schedules of buses and trams)
* Dormitory.cs (describes an object of a dormitory by address, geographical coordinates, train or subway station, list of public transport)
* DubkiBusSchedule.cs (describes an object of bus schedule in Dubki by departure time, departure and destination point for each day of the week)
* HSEBuilding.cs (describes an object of HSE building by address, geographical coordinates and the closest subway station)
* LocalTrainPrice.cs (describes an object of train price depending on departure and destination points)
* LocalTrainSchedule.cs (describes an object of train schedule by list of stops, departure time, departure and destination point)
* LocalTrainStation.cs (describes an object of each train station)
* LocalTrainStop.cs (describes an object of each train stop by arrival time and elapsed time)
* PublicTransport.cs (describes an object by departure time, departure and destination point, price for public transport depending on the day of the week)
* PublicTransportPrice.cs (describes an object of a price for public transport)
* SubwayElapsedTime.cs (describes an object of time, taken between 2 subway stations)
* SubwayStation.cs (describes an object of a subway station by geographical coordinates, lists of nearby dormitories, HSE buildings and types of transport)
* TransportType.cs (describes an object of each type of transport)

Models:

* DormitoryData.cs
* DubkiBusData.cs
* HSEBuildingData.cs
* PublicTransportData.cs
* SubwayStationData.cs
* Transport.cs
* Context.cs (creates a database)
* DatabaseQuery.cs (contains requests)
* LoadFromCSV.cs (contains methods for loading the data from CV files)

**UberService**

DTO:

* PriceQueryResponse.cs
* RideOffer.cs
* RequestBuilder.cs (creates an API request)
* UberService.cs (sends a request and receives a JSON response)

**YandexScheduleService**

DTO:

* StationCodes.cs
* TrainStation.cs
* TrainStop.cs
* TrainThreadInfoResponse.cs
* TrainInfo.cs

- TrainStation.cs

- TrainThread.cs

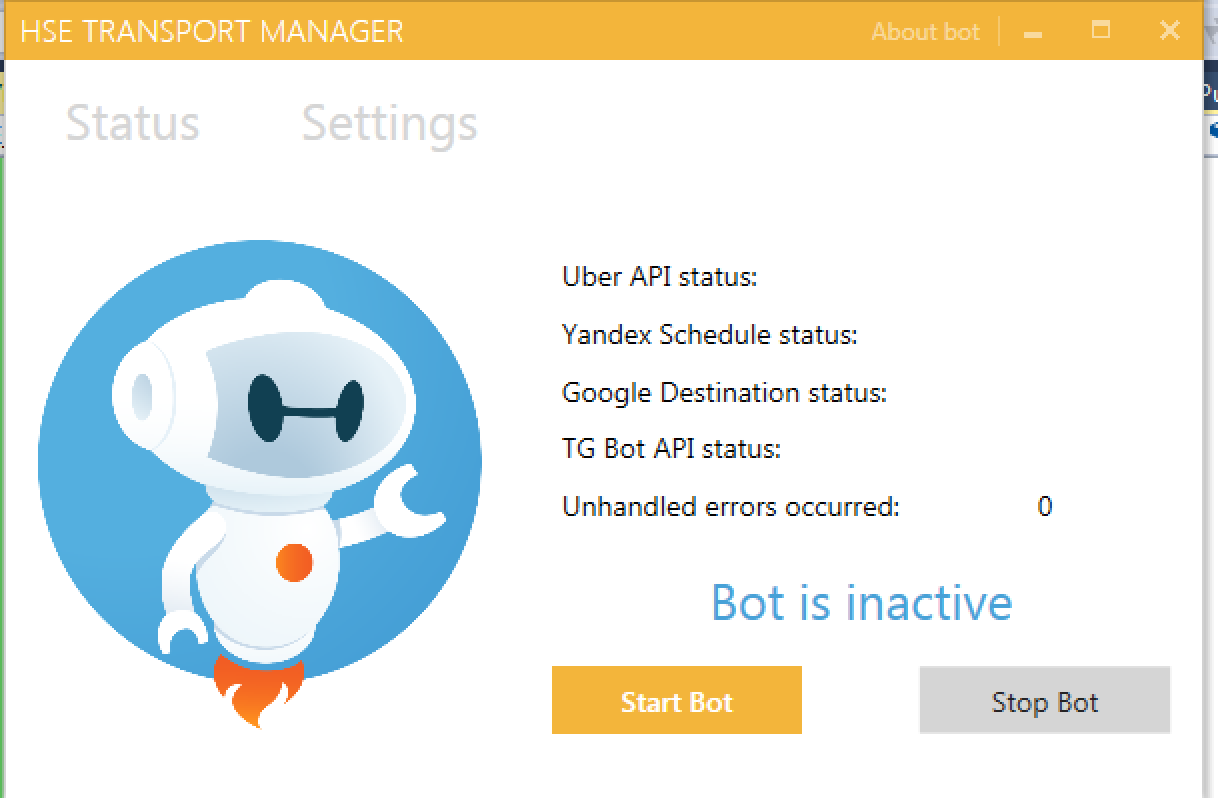
- TrainThreadsListResponse.cs

- RequestBuilder.cs (creates an API request)

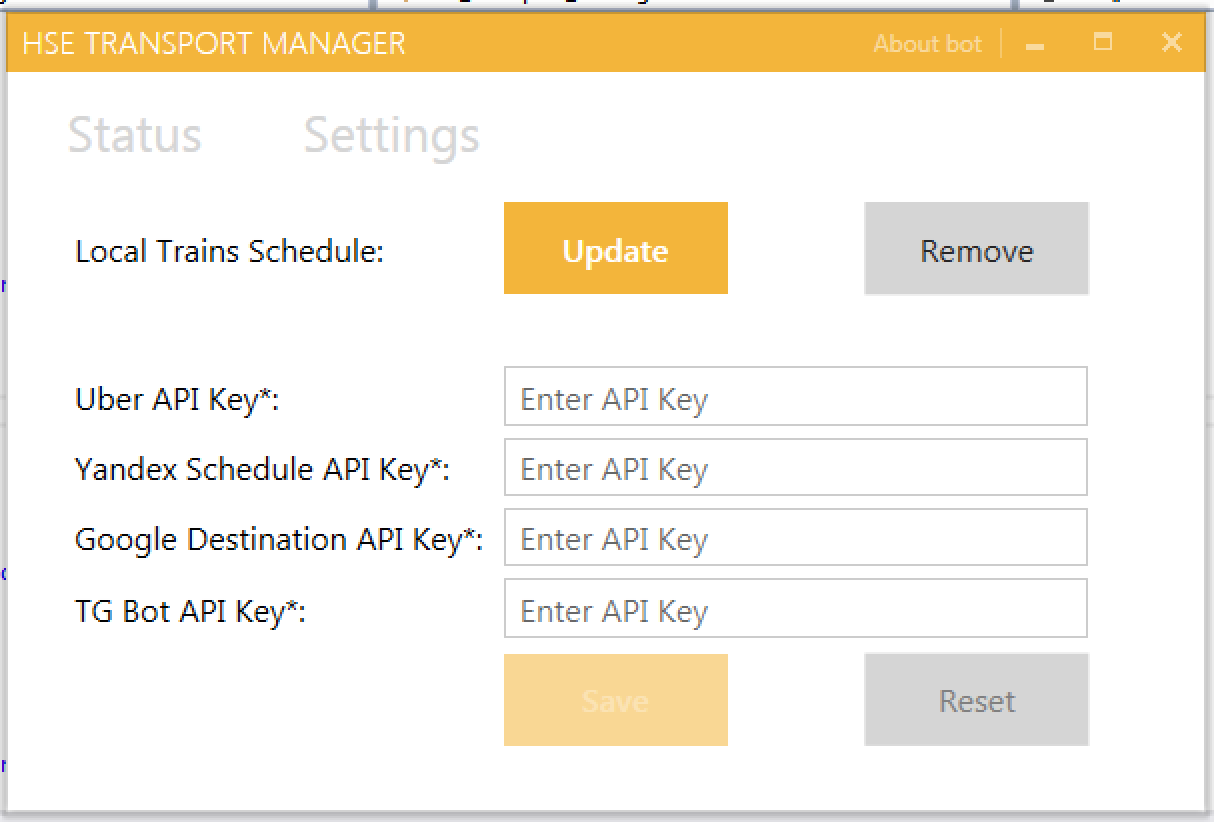
- YandexScheduleService.cs (sends a request and receives a JSON response)

1. *Program interface*

**StatusView**

****

**SettingsView**



**AboutView**

