

СОФИЙСКИ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“

Проект по

**Софтуерни системи, базирани на услуги**

Тема:

**Информационна система за наблюдение на разписанието на автогара**

**Изготвил:** Георги Любенов Димитров, Ф.Н.: 25416

**Контакти**: georgild9@gmail.com

**СОФИЯ**

**2017**

Съдържание

1. Цел на проекта 3
2. Използвани програмни технологии 4
3. Архитектура на системата 5
4. Описание на програмната реализация 6
5. Заключение 9
6. Връзки 10

**1. Цел на проекта**

Целта на проекта е да се реализира уеб-базирана информационна система за наблюдение на разписанието и маршрутите на автобусите в дадена автогара.

Реализираната система извежда информация за пристигащите и заминаващите автобуси на гарата. Потребителят може да филтрира пристигащите по следните категории: дата на пристигане, от къде пристига(град), име на компанията. Заминаващите могат да се филтрират по: дата на заминаване, за къде заминава(град), име на компанията. Информацията, която се извежда показва и цена на билета. При филтрацията потребителят може да избере в каква валута да се показва цената.

Системата предлага и административен интерфейс, през който администратор може да добавя маршрути на заминаващите и пристигащите на гарата.

**2. Използвани програмни технологии**

Системата е разделена на два основни компонента: frontend част, реализирана като SPA (Single Page Application) с ReactJS и backend част, представляваща няколко уеб услуги.

Реализирайки frontend частта като отделно SPA приложение получаваме няколко основни предимства: независимост и слаба свързаност на frontend частта спрямо сървърната имплементация, гъвкаво и бързо уеб приложение работещо на съвременни устройства, възможност за използване на компоненти реализирани чрез една от най-използваните javascript работни рамки.

Backend частта е базирана на уеб услуги, реализирани върху платформата .NET и използващи два комуникационни протокола: REST и SOAP. Реализирайки тази част чрез уеб услуги ни дава следните предимства: ясно описание на възможностите на сървърната имплементация, лесно интегриране от и с външни услуги, лесно поддържане и скалиране на услугите, реализирани като отделни логически единици. Интегрирана е и външна SOAP уеб услуга, даваща спомагателни функционалности.

REST базираните уеб услуги са реализирани като http self-hosted независими услуги, чакащи уеб заявки на даден порт. SOAP помощната услуга е реализирана стандартно като .NET Web Service.

За съхраняване на информацията се използва MongoDB база данни в стандартен JSON формат.

**3. Архитектура на системата**

От високо ниво общата архитектура на системата по компоненти изглежда по следния начин:

Data Store (MongoDB)

External SOAP Web Service

Web Server

Frontend

(ReactJS App)

UtilityService

(.NET SOAP Service)

RoutesService

(.NET RESTful Service)

IdentityService

(.NET RESTful Service)

**4. Описание на програмната реализация**

Frontend компонентът е реализиран на езика JavaScript, използвайки framework – ът ReactJS. Форматът на данните, които използва приложението е JSON, доставени чрез AJAX заявки към сървърната част.

Имаме две основни view – та достъпни чрез маршрутите *‘/main’* и *‘/admin’*.

В *‘/main’* е изградена основната таблица представяща информацията за пристигащите и заминаващите. Файлът Main.js съдържа и показва един от двата основни компонента изградени като ReactJS компоненти – съответно във файловете Arrivals.js и Departures.js. Таблицата с данни се изгражда от помощния ReactJS компонент ReactDataGrid. Имаме и две форми за филтрация на данните, използвани в компонентите Arrivals и Departures, реализирани във файловете FilterFormArrivals.js и FilterFormDepartures.js

Чрез *‘/admin’* се достъпва логин прозореца, който води до администраторския панел. Логин панела е реализиран като ReactJS компонент – Login (във файла Login.js), а администраторския такъв в Admin.js

Backend уеб услугите, а именно Identity и Routes изграждат JSON Web API, достъпно през HTTP протокол.

IdentityService предоставя възможност на клиентите на услугата да се сдобият с access token през web endpoint, който услугата предоставя – *‘/token’.* Чрез тази услуга е реализиран широко използваният протокол за аутентикация – OAuth 2.0. Също така услугата предоставя endpoints за CRUD операции върху потребителите:

*GET /api/v1/users*

*POST /api/v1/users*

*PUT /api/v1/users/{id}*

*DELETE /api/v1/users/{id}*

Потребителите се пазят в JSON формат в колекцията *users* в базата MongoDB.

Примерен JSON user:

*{  
    "\_id" : ObjectId("59047bc73522edc64b6413c1"),  
    "UserName" : "*[*admin@admin.com*](mailto:admin@admin.com)*",  
    "Password" : "1234"   
}*

RoutesService предоставя възможност за CRUD операции върху маршрутите на аерогарата, като само аутентикиран администратор с access token, получен от Identity може да трие, променя и създава маршрути.

Endpoint – ите, които предоставя RoutesService са:

*GET /api/v1/routes*

*POST /api/v1/routes*

*PUT /api/v1/routes/{id}*

*DELETE /api/v1/routes/{id}*

Маршрутите се пазят в JSON формат в колекцията *routes* в базата MongoDB.

Примерен JSON маршрут:

*{  
    "\_id" : ObjectId("592954f1b04f2b2cd0447782"),  
    "CompanyName" : "Company 5",  
    "InitialStop" : {  
        "ArrivalDate" : NumberLong(1509100037000),  
        "DepartureDate" : NumberLong(1509100037000),  
        "City" : "Bourgas"  
    },  
    "IntermediateStops" : [   
        {  
            "ArrivalDate" : 0,  
            "DepartureDate" : 0,  
            "City" : "Sofia"  
        },   
        {  
            "ArrivalDate" : 0,  
            "DepartureDate" : 0,  
            "City" : "Sofia"  
        }  
    ],  
    "FinalStop" : {  
        "ArrivalDate" : NumberLong(1509100039000),  
        "DepartureDate" : NumberLong(1509100039000),  
        "City" : "Sofia"  
    },  
    "TicketPrice" : 22.2,  
    "Type" : 0  
}*

UtilityService е помощна услуга, достъпна чрез SOAP комуникационен протокол. Тя предоставя методът *ConvertCurrency(double amount, string fromCurrency, string toCurrency),* който предоставя възможност за преобразуване на дадена стойност от една валута към друга валута. Валутният курс се взима от външната услуга, достъпна отново чрез SOAP протокола през endpoint – а: [*http://www.webservicex.net/CurrencyConvertor.asmx*](http://www.webservicex.net/CurrencyConvertor.asmx)

Backend RESTful уеб услугите използват общ *Data Layer,* през който минават всички заявки към базата данни MongoDB. Реализиран като отделен .NET модул, той лесно може да бъде преизползван в останалите backend модули, напълно изолиран и лесен за поддръжка и разширение.

**5. Заключение**

Проектът реализира уеб-базирана информационна система за следене разписанието и графика на автогара. Базира се на няколко RESTful и SOAP реализирани уеб услуги. Те изграждат лесно достъпно JSON API. Услугите се интегрират от уеб сайт, изграден като SPA приложение чрез ReactJS.

Архитектурата на системата позволява бъдещото й разширение с още функционалности, както откъм frontend, така и от към backend гледна точка.

Като бъдещи подобрения може да се разшири административният интерфейс с възможности за въвеждане на информация за транспортните фирми, както и създаване и промяна на други администратори на системата. Основният интерфейс също може да бъде подобрен с възможности за сортиране по различните колони, както и повече информация за маршрутите.

**5. Полезни връзки**

GitHub repository на проекта: *https://github.com/georgild/transport-system*