

СОФИЙСКИ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“

Проект по

**Софтуерни системи, базирани на услуги**

Тема:

**Информационна система за наблюдение на трафика на аерогара**

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

**Контакти**: georgild9@gmail.com

**СОФИЯ**

**2017**

Съдържание

1. Цел на проекта 3
2. Използвани програмни технологии 3
3. Архитектура на системата 4
4. Описание на програмната реализация 5
5. Заключение 13

**1. Цел на проекта**

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

Реализираната система извежда информация за пристигащите и заминаващите полети 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 е базиран 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 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 цялостната 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 REST Service)

IdentityService

(.NET REST 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 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 REST протокола.

IdentityService предоставя възможност на клиентите на услугата да се сдобият с access token чрез единствения web endpoint, който услугата предоставя – ‘/token’. Чрез тази услуга е реализиран основният протокол за аутентикация – OAuth 2.0.

RoutesService предоставя възможност за CRUD операции върху маршрутите на аерогарата, като само аутентикиран потребител с access token, получен от Identity може да трие, променя и създава маршрути. Endpoint – ите, които предоставя RoutesService са:

*GET/POST /api/v1/routes*

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

Маршрутите се пазят в JSON формат в базата MongoDB.

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 дадена валута към друга валута. Преобразуващият коефициент се взима от външната услуга, достъпна отново чрез SOAP протокола през endpoint - а [*http://www.webservicex.net/CurrencyConverter*](http://www.webservicex.net/CurrencyConverter)*.*

**5. Заключение**