ДЕСЕТО НАЦИОНАЛНО СЪСТЕЗАНИЕ ПО ИНФОРМАЦИОННИ ТЕХНОЛОГИИ

гр. Благоевград, 23 март– 24 март 2018 г.

# Тема

MY EXAMS – създаване и проверка на тестове

# Автори

**АВТОР:**

Петър Светославов Бечев, гр. Пловдив, ул. Захари Стоянов 39а,

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

МГ „Академик Кирил Попов“ гр. Пловдив

**РЪКОВОДИТЕЛ:**

Инж. Дарина Брънчева, 0888 992 447, e-mail: dary\_brancheva@abv.bg

Старши учител по информатика и ИТ в МГ

# Изложение

## 3.1. Увод

MY EXAMS е интернет приложение, което има за цел да улесни работата на всеки преподавател в създаването и проверката на тестове, като в същото време допринася за намаляването на преписването, тъй като създадените версии за всеки ученик са различни - тоест разбъркват се въпросите и отговорите на случаен принцип. Заедно с това се генерира и бланка за отговор, която е в абсолютно съответствие с въпросите. След като ученикът отбележи своите отговори, учителят може да сканира документа и системата автоматично ще го провери. Уеб приложението има и роля на място за съхранение и огранизация на всички изпитни варианти, оценки, класове и ученици.

## 3.2. описание на Приложението

### ЗА УЧИТЕЛИТЕ

Уеб приложението е достъпно на адреса myexamsbg.com. Регистрацията за учители става от формата за регистрация на учителите, след въвеждането на основните данни, учителят може да започне работата си с приложението. Първото нещо, което може да направи е да създаде класовете, на които преподава. Това става от бутона „Всички класове“ на менюто в дясно. При липса на класове екранът ще бъде празен, а за да се съдаде нов е необходимо да се натисне бутона „+ Нов клас“, който се намира в горния десен ъгъл, оцветен в зелен цвят. На екрана се визуализира форма, в което се въвеждат името на класа като например „10Ж“ „11В“ и т.н и учебния предмет – например „Математика“, „Физика“ и т.н. След натискането на бутона създай, директно се отваря странизата на новосъздадения клас. Вероятно се забелязва оцветяването на страницата. Този цвят се генерира автоматично на случаен прицип за всеки клас с цел разнообразие и естететичност. Ясно се вижда и уникалният код на клас, изписан в гориня десен ъгъл на заглавното поле. Този код е изключително важен, защото с негова помощ всеки ученик се присъединява към класа и може да получава своите резултати директно в своя профил. Преподавателят има за задача да предостави този код на своите ученици.

В първото поле „Последни тестове“ се визуализират резултатите от последните проверени тестове за съотвения клас. В следващия раздел „Ученици и оценки“ в табличен вид са представени всички ученици въвели уникалния код и присъединили се в съотвения клас, а оценките се изписват за всяко изпитване отново в съответната таблица.

Следващият етап е самото създаване на тестовете. Това става от раздела на навигацията „Всички тестове“. Анологично възможно е екранът да бъде празен, тъй като няма създадени тестове. При натискане на бутона „+ Създай тест“ се зарежда редактора на тестове. В най-горното поле се въвежда заглавието на самия текст. От бутона „Добави секция“ става създаването на всеки нов раздел на теста. Във всеки раздел въпросите се разбъркват в самия раздел и по този начин, логическа подредба на различните типове въпроси може да бъде реализирана по много лесен и интуитивен метод. Препоръчително е въвеждането на заглавие на секцията, но не е задължително. За добавянето на нов въпрос се натиска един от двата бутона „Избор“ или „Текст“, в зависимост от типа на желания въпрос.

В случая „Избор“ означава въпрос, който е с избираем отговор. Добавянето на отговорите става от бутона „+ Добави отговор“ след като естествено първо е създаден въпроса. За да се обозначи даден отговор като верен е необходимо курсора да сочи поленцето на съответния отговор и в края на това поленце ще се появи тикче. Когато бъде натиснато то ще се превърне в поле на което е изписано „Правилен отговор“. По този начин при проверката системата автоматично ще знае кой отговор да се счита за верен. На всеки отговор има бутон, на който служи за изтриването на съотвения отговор. Има опция потребителят сам да избере дали на дадения въпрос да се разбъркват отговорите и това става от обекта „Разбъркай отговорите“. Задължително е и въвеждането на брой точки, защото по този начин системата ще може да изчисли резултат след проверката.

Чрез бутона „Текст“ се създава въпрос със свободен отговор, на който всеки ученик трябва да отговори писменно. На въпроса от този тип има избор на размер за отговора – „Кратък“, „Среден“ и „Дълъг“ като това означава колко голямо да бъде полето на бланката. Препорачително е да се въведе правилен отговор, тъй като при проверката по-късно ще бъде много по-лесто и бързо. Всички останали неща са абсолютно аналогични като тези при въпроса с избираем отговор.

Единствено с цел потребителско улеснение и сигурност всяка промяна по теста се запазва моментално абсолютно автоматичто, тоест дори и да се получи срив в браузъра учителя може да бъде сигурен, че неговата информация е запазена на сървъра.

За да бъдат генерирани всички групи се натиска бутона „Принтирай“, който се намира в долния десен ъгъл на страницата. Веднага се зареждат всички класове, като само с един клик става избирането на желаните класове и бутона „Избери“ ще задейства процеса по генерирането на изпитните варианти. В общия случай процесът продължава не повече от 3-4 секунди и след това се отваря файлът в нов раздел. Той е във формат .pdf, който може директно да бъде разпечетан или да се запази на текущото устройство.

След като се попълнят бланките за отговор от всички ученици, учителите трябва да сканират файловете в Jpg формат. Страницата за качване на файловете се достъпва от бутона „Качи файлове“. От бутона „Избери файлове...“ се отваря файловата директория на компютъра и потребителят избира съответните сканирани файлове. След като качването приключи учителят може да се отдаде на заслужената почивка докато системата поверява тестовете. Те са поименни – тоест на всеки от вариантите има написно името на ученика, за когото е предназначен и резултатът ще бъде записан в неговия профил.

Когато проверяването приключи ще се изведе известие за успеха на проверката на всеки един от файловете. В случайте когато на сканираните тестове е имало въпроси с отворен отговор, то ще има известие, че за задършване на проверката е необходимо учителят ръчно да потвърди тези отговори. Чрез бутона „Започни сега“ един по един ще се визуализират въпросите, правилния отговор и изрярък от бланката, където се съдържа отговора на въпроса, учителят ще трябва да избере броят на точки в зависимост от точността и пълнотата на отговора. След като всички въпроси бъдат потвърдени резултати ще се появят в таблицата с оценките на всеки от класовете.

3.3. основните етапи в реализирането на проекта

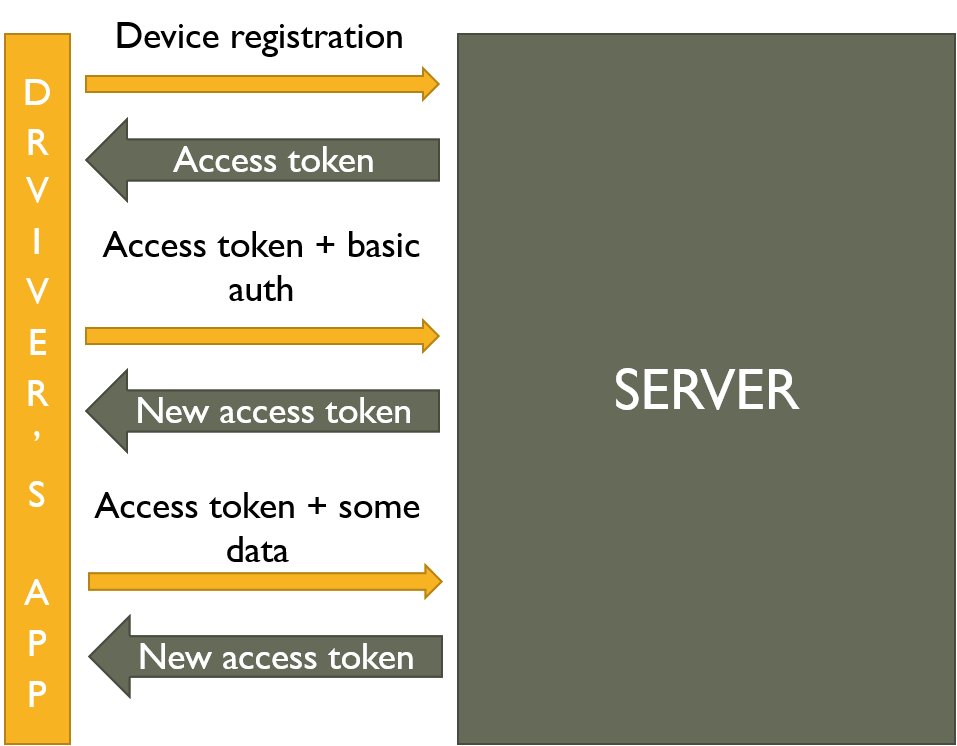
При реализирането на проекта преминах през следните етапи:

1. Обмислянето на структурата на проекта
2. Изграждане на Data layer – бази данни
3. Проектирането на сървърната част
4. Създаване на двете потребителски роли – учител и ученик
5. Имплементиране на логиката свързана със създаването и модификацията на класовете и тестовете
6. Разработване на метод за генериране на различните варианти на тестовете
7. Обмисляне и изграждане на процеса на проверяване, който да работи на заден план (Background process)
8. Проверяване и отстраняване на грешките
9. Документиране и внедряване

3.4. Ниво на сложност

По време на реализирането на проекта се справих с някои основни проблеми като:

1. Изясняването на структурата на проекта и връзката между отделните компоненти
2. Избор на подходящи технологии, за доброто реализиране на проекта
3. Изработване на мобилно приложение, което да работи на трите основни мобилни платформи: Android, Windows Phone, IOS
4. Следване на добри практики при използването на програмните средства
5. Свързването на всички части в един цялостен проект

Графиката по-долу обуславя начина, по който се усъществява комуникацията между приложенията и сървъра: 

При първоначално стартиране на приложението се изпраща заявка, с която се регистрира съответното устройство в системата и сървърът изпраща AccessToken или с други думи казано уникален ключ за достъп. Когато се изпратят паролата и имейла (Credentials), естествено в кодиран вид, се изпраща и този уникален ключ за достъп заедно със заявката и сървърът освен отговор за валидността на данните връща и нов уникален ключ за достъп, а старият става неактивен. Ако бъдат валидни данните, всека следваща заявка към сървъра ще използва ключа за достъп за аутентикация и няма да се налага пращането на паролата и имейла при всяка заявка. Ползите са следните: имейла и паролата (sensitive data) няма да се пазят под никаква форма на устройството; ключът за достъп се изменя постоянно и всички предходни стават невалидни тоет дори и някой да успее да вземе този ключ, той ще бъде изтекъл в момента в който реши да го използва; ключът за достъп позволява да се реализира функционалност, която не допуска две устройства да са влезнали в един и същ профил по едно и също време.

## 3.5. Реализация

Проектът е реализиран с помощта на следните технологии:

* ASP.NET MVC - е платформа, която служи за изработване на уеб приложения, използвайки модела Model-View-Controller (MVC). Тя позволява имплементирането на администраторската част така и на web services. При изпращане на HTTP заявка към сървъра той върща данни във JSON формат, който се презентират по подходящ начин на потребителите.
* Telerik Platform – уеб базирана платформа за създаване на хибридни мобилни приложения, с помоща на HTML, CSS, JAVASCRIPT, за трите основни мобилни платформи.
* Google Console API – уеб сървиси, които връщат определен тип данни в JSON формат като например разстоянието между две точки или адреса на точка с определени координати.
* SQL Database – за data layer е използвана SQL база данни с помощта на Entity Framework Code First подхода

Използвани библиотеки (всички open source) :

jQuery – DOM манипулация

Bootstrap – front-end framework за дизaйн

Ninject- open source dependency injector

Knockout.js – библиотека за направата на single-page apps

Chart.js – библиотека за създаването на графиките

Системата е защитена от популярните уеб атаки :

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

Мобилното приложение за шофьори :

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

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

Мобилното приложение за клиенти:

Android – [http://bgtaxi.net/download/client/apk](http://bgtaxi.net/download/client/apk%20)

Windows Phone - [http://bgtaxi.net/download/client/xap](%20http:/bgtaxi.net/download/client/xap)

3.6. Заключение

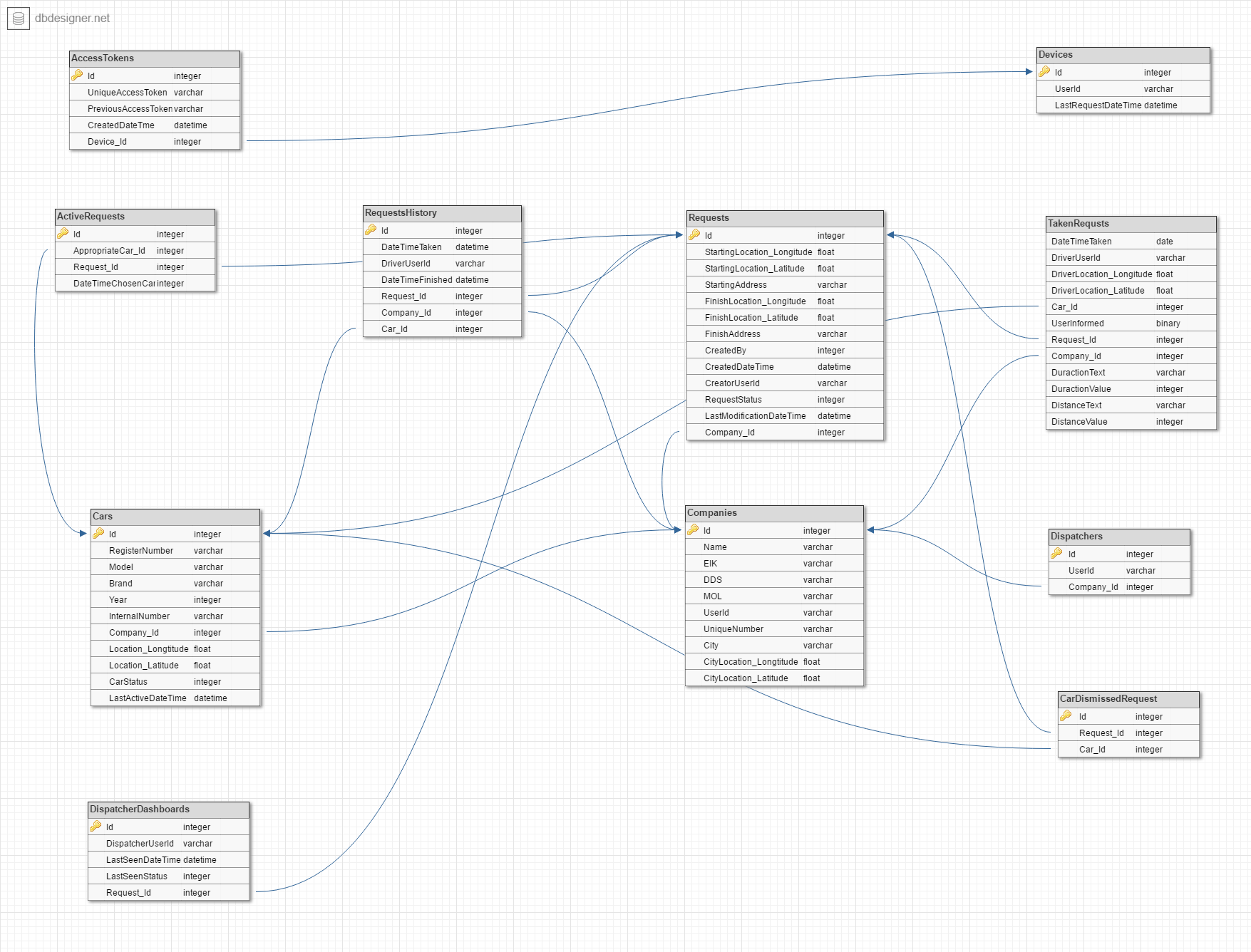
В заключение мога да заявя, че въпреки същесвтвуването на подобни приложение и проекти аз успях да създам една система, която превъзхожда останалите по няколко ключови аспекта. BG TAXI изпраща заявките само на близките коли в региона, което не се наблюдава при някои от вече съществуващите системи, които изпращат заявките на всички коли дори и да се намират на огромно разстояние от заявката, което много прилича на остарелия метод с използването на радиостанции. Другото нещо е, че проектът предпоставя бъдещото премахване изцяло на диспечерите и преминаването към алтернативен нов метод, а именно приложението за клиенти, който е по-лесен, удобен и предпочитан от хората в днешно време.

## 3.7. Използвана литература

За реализацията на проекта използвах следните източници на информация:

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

<http://telerikacademy.com/> - онлайн курсове по C#, ASP.NET MVC; <http://www.w3schools.com/> - HTML, JAVASCRIPT, CSS, BOOTSTRAP

СХЕМА НА БАЗАТА ДАННИ НА ПРИЛОЖЕНИЕТО