МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКАТА

**ПРОФЕСИОНАЛНА ГИМНАЗИЯ “ГЕН. ВЛАДИМИР ЗАИМОВ” гр. СОПОТ**

4330 гр. Сопот, ул. ”Иван Вазов” №1, тел./факс: /03134/ 83-31, 83-32, e-mail: [pgzaimov@yahoo.com](mailto:pgzaimov@yahoo.com)

**ДИПЛОМЕН**

**ПРОЕКТ**

**Тема: „ShiftOn“**

**Сайт, организиращ график на ресторант**

*Ученик: Преслава Маринова Петрова*

***Професия:*** *код 481020 „Системен програмист“*

***Специалност:*** *код 4810201 „Системен програмиране“*

***Консултант****: Живко Маджев*

Сопот, 2023 г.

**Съдържание**

**Увод**

Въведение в предметната област 4

**Първа глава. Проучване**

1. Предпоставка за създаване 5

2. Анализ на съществуващи решения 5

3. Цел на проекта 5

**Втора глава. Проектиране**

1. Използвани технологии и подготовка на работна среда 6

2. Изисквания към софтуерния продукт 10

2.1. Софтуерни

2.2. Интерфейсни

2.3. Хардуерни

**Трета глава. Описание**

1. Структурата на базата данни 11

2. Структура на MVC приложението 12

**Използвана литература** 13

**Приложение** 14

**Увод**

**Въведение в предметната област**

**ПЪРВА ГЛАВА.**

***Проучване***

1. **Предпоставка за създаване**

<zadanie na diplomen proekt>

1. **Анализ на съществуващи решения**

<drugi sajt>

1. **Цели на проекта**

Целта на този проект е да се реализира уеб платформа, която да управлява графика на служителите в ресторант, да подобри организацията на заведението, като създаде по-добра и лесна за използване платформа за създаване на графици. Когато влезнем като потребител ще можем да проверяваме графика и смените за седмицата, в която сме на работа. Системният администратор на платформата се грижи за управлението на ресторантите и техните управители. Управителят на ресторант е потребител, регистриран от системния администратор. Управителят може да редактира работните графици на своите служители. Заявлението за почивен ден от служител, може да бъде одобрено или отхвърлено от управителя на ресторанта.

**ВТОРА ГЛАВА.**

**Проектиране**

1. **Използвани технологии и подготовка на работна среда (C#, Html, CSS, ASP .Net, .Net Core, Java Script, Bootstrap)**
   1. **C#** е език за програмиране на високо ниво с общо предназначение, поддържащ множество парадигми. Той обхваща статично типизиране, силно типизиране, лексикален обхват, императивно, декларативно, функционално, общо, обектно-ориентирано и компонентно-ориентиране програмиране. C# е обектно-ориентиран език за програмиране, разработен от Microsoft като част от софтуерната платформа .NET. Стремежът още при създаване на C# езика е бил да се създаде прост, модерен, обектно-ориентиран език с общо предназначение. Основа за C# са C++, Java и донякъде езици като Delphi, VB.NET и C. Той е проектиран да балансира мощност (C++) с възможност за бързо разработване. Те представляват съвкупност от дефинирани на класове, които съдържат в себе си методи, а в методите е разположена програмната логика – инструкциите, които компютърът изпълнява. Програмите на C# представляват един или няколко файла с разширение .cs, в които се съдържат дефиниции на класове и други типове. Тези файлове се компилират от компилатора на C# до изпълним код и в резултат се получават асемблита – файлове със същото име, но с различно разширение (.exe или .dill).
   2. **Html е** основният маркиращ език за описание и дизайн на уеб страници. Текущата версия на стандарта е HTML 5.0 (от 28 октомври 2014 г.), а предходната стабилна версия е HTML 4.1. Описанието на документа става чрез специални елементи, наречени HTML елементи или техните маркери, които се състоят от тагове и съответстващите етикети (HTML tags) и ъглови скоби (като например елемента <html>). HTML елементите са основната градивна единица на кода, който изграждат уеб страниците. Чрез тях се форматира, графично оформя текста и неговите отделните части в рамките на една уеб страница, като например заглавия, цитати, текстови раздели, хипертекстови препратки и т.н.

Най-често HTML елементите са групирани по двойки <h1> и </h1>. В повечето случаи HTML кодът е написан в текстови редактори, с файлов формат .html, .htm, dhtml и се качва и хоства на [сървъри](https://bg.wikipedia.org/wiki/%D0%A1%D1%8A%D1%80%D0%B2%D1%8A%D1%80" \o "Сървър), които са онлайн в интернет или са част от [www](https://bg.wikipedia.org/w/index.php?title=Www&action=edit&redlink=1) мрежата. Тези .html файлове съдържат програмно на таговете на HTML и текстово съдържание със маркери и коментари – също инструкции за [браузъра](https://bg.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D1%8A%D1%80" \o "Браузър), за това какъв точно тип е .html страницата, а също за това как да се показва текстът, особено що се отнася до езиковите характеристики. [Уеб браузърите](https://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1_%D0%B1%D1%80%D0%B0%D1%83%D0%B7%D1%8A%D1%80) са програмирани от своя страна така, в повечето случаи, макар че някои браузъри могат да имат съответно проблеми на версията, за да могат да прочетат HTML документите и да ги покажат на екрана като уеб страници. Браузърите не показват самите HTML тагове, освен ако не се отиде в менюто за да се направи това, така че те „интерпретират“ (тоест [парсват](https://bg.wikipedia.org/w/index.php?title=Parsing&action=edit&redlink=1" \o "Parsing (страницата не съществува))) съдържанието на страницата като код и текст за да могат след работа на процесора да покажат желаното уеб-съдържание.

* 1. **CSS** е език за програмиране и също описание на уеб дизайн програмни стилове (език за стилови файлове, style sheet language) – използва се основно за описание на онлайн представянето на уеббазиран документ, който написан на език за маркиране. Най-често се използва допълнително към чистия HTML, но се прилага и върху XML уебстраници и документи. Спецификацията на CSS официално се поддържа от W3C. CSS още в началото на развитието на www започва да се добавя към стандартния HTML с цел да бъдат разделени съдържанието и структурата на уеб страниците отделно от тяхното визуално представяне. Преди стандартите за CSS, установени от W3C през 1995 г., съдържанието на сайтовете и стила на техния дизайн са писани в една и съща HTML страницата. В резултат на това HTML кодът се превръща в сложен и нечетлив, а всяка промяна в проекта на даден сайт изисквала корекцията да бъде нанасяна в целия сайт страница по страница. Използвайки CSS, настройките за форматиране могат да бъдат поставени в един-единствен файл и тогава промяната ще бъде отразена едновременно на всички страници, които използват този CSS файл.
  2. **ASP .Net** е технология за създаване на [уеб приложения](https://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) и [уеб услуги](https://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1_%D1%83%D1%81%D0%BB%D1%83%D0%B3%D0%B0), разработена от „Microsoft “. За първи път е публикуван през януари 2002 г. с версия 1.0 на [.NET Framework](https://bg.wikipedia.org/wiki/.NET_Framework) и е наследник на Microsoft Active Server Pages (ASP) технологията, но не е подобрена версия на ASP. ASP.NET е изградена въз основа на Common Language Runtime (CLR), което позволява на програмистите да пишат ASP.NET код, като използват .NET език по избор.  
     ASP.NET цели производителност спрямо останалите скрипт базирани технологии (включително класическия ASP), като компилира сървърно кода в един или повече DLL файлове на уеб сървъра. Тази компилация става автоматично, когато страницата бива заредена за пръв път (което от своя страна означава, че програмистът не трябва да изпълнява отделни компилации за страниците). Това комбинира лекотата на разработване, предлагана от скриптовите езици, с производителността на бинарните операции. Трябва да се има предвид обаче, че самата компилация може да причини забележимо забавяне при потребителя, когато редактираната страница бива изискана за пръв път от уеб сървъра, но това забавяне не би се появило отново преди следваща промяна.
  3. **.Net Core** е най-новата платформа за разработка с общо предназначение, поддържана от Microsoft. Той работи на различни платформи и е преработен по начин, който прави .NET бърз, гъвкав и модерен. Това се оказва един от основните приноси на Microsoft. Разработчиците вече могат да създават приложения за Android, iOS, Linux, Mac и Windows с .NET, всички с отворен код. Това е рамка с отворен код, между платформи, разработена както от Microsoft, така и от неговата общност. Това е пълна реформа на ASP.NET, която комбинира структурата на MVC и Web API в една рамка. Някои от основните промени включват - междуплатформени ASP .NET приложения на Windows, Mac, Linux; здраво представяне; съвременни инструменти за опростяване на разработката; рамка с отворен код; по-бързо компилиране, асинхронни операции.
  4. **Java Script** е интерпретируем език за програмиране, разпространяван с повечето уеб браузъри. Поддържа обектно ориентиран и функционален стил на програмиране. Създаден е в Netscape през 1995 г. Най-често се прилага към HTML-а на интернет страница с цел добавяне на функционалност и зареждане на данни.

Може да се ползва също за писане на сървърни скриптове JSON, както и за много други приложения. JavaScript не трябва да се бърка с Java, съвпадението на имената е резултат от маркетингово решение на Netscape. Javascript е стандартизиран под името EcmaScript. JavaScript е програмен език, който позволява динамична промяна на поведението на браузъра в рамките на дадена HTML страницата. JavaScript се зарежда, интерпретира и изпълнява от уеб браузъра, който му осигурява достъп до Обектния модел на браузъра. JavaScript функции могат да се свържат със събития на страницата (например: движение/натискане на мишката, клавиатурата или елемент от страницата, и други потребителски действия). JavaScript е най-широко разпространеният език за програмиране в интернет. Прието е JavaScript програмите да се наричат скриптове.

* 1. **Bootstrap** е безплатна рамка за [преден край за разработка с](https://www.techtarget.com/whatis/definition/front-end)[отворен код](https://www.techtarget.com/whatis/definition/Free-and-open-source-software-FOSS-or-free-libre-open-source-software-FLOSS) за създаване на уеб сайтове и уеб приложения. Създаден, за да даде възможност за бързо [реагиране](https://www.techtarget.com/whatis/definition/responsive-design) на уеб сайтове, ориентирани към [мобилни устройства](https://www.techtarget.com/searchmobilecomputing/definition/mobile-first) , Bootstrap предоставя колекция от синтаксис за дизайн на шаблони. Като рамка, Bootstrap включва основите за адаптивна уеб разработка, така че разработчиците трябва само да вмъкнат кода в предварително дефинирана мрежова система. Рамката [Bootstrap](https://www.theserverside.com/quiz/12-tough-Bootstrap-5-quiz-questions-for-web-developers) е изградена върху Hypertext Markup Language ( [HTML](https://www.theserverside.com/definition/HTML-Hypertext-Markup-Language)) , cascading style sheets ( [CSS](https://www.theserverside.com/definition/cascading-style-sheet-CSS)) и [JavaScript](https://www.theserverside.com/definition/JavaScript) . Уеб разработчиците, използващи Bootstrap, могат да създават уебсайтове много по-бързо, без да губят време да се тревожат за основни команди и функции. Bootstrap прави отзивчивия уеб дизайн реалност. Това дава възможност на уеб страница или приложение да разпознават размера и ориентацията на екрана на посетителя и автоматично да адаптират дисплея съответно. Подходът за мобилни устройства предполага , че [смартфоните](https://www.techtarget.com/searchmobilecomputing/definition/smartphone) , [таблетите](https://www.techtarget.com/searchmobilecomputing/definition/tablet-PC) и мобилните приложения за конкретни задачи са основните инструменти на служителите за извършване на работа. Bootstrap отговаря на изискванията на тези технологии в дизайна и включва [UI](https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI) компоненти, оформления, JavaScript инструменти и рамката за изпълнение. Софтуерът се предлага предварително компилиран или като [изходен код](https://www.techtarget.com/searchapparchitecture/definition/source-code) .

1. **Изисквания към софтуерния продукт**
   1. **Софтуерни**

**Windows Server**

**IIS Express**

**SQL Server**

**SSMS sql server management studio**

**Azure WebApps**

**Static IP**

**DNS**

* 1. **Интерфейсни <opisanie na stranici (wsichki)>**
     1. **Функционални изисквания:**

За да се реализира уеб приложението „ShiftOn“, което организира графика на служители в ресторант, то трябва да поддържа следните функционалности:

1. Роли за работа с приложението:

* Администратор;
* Служител;

1. Регистрация на администратор:

* Име и фамилия;
* Имейл адрес;
* Парола;
* Фирма;
* Служители във фирмата;

1. Регистрация на служител:

* Име и фамилия;
* Потребителско име;
* Имейл адрес;
* Парола;

1. Като администратор:

* Добавяне на график;
* Променяне на график;
* Изтриване на график;
* Добавяне на служители;
* Изтриване на служители;
* Приемане на заявка за почивни дни;
* Отхвърляне на заявка за почивни дни;

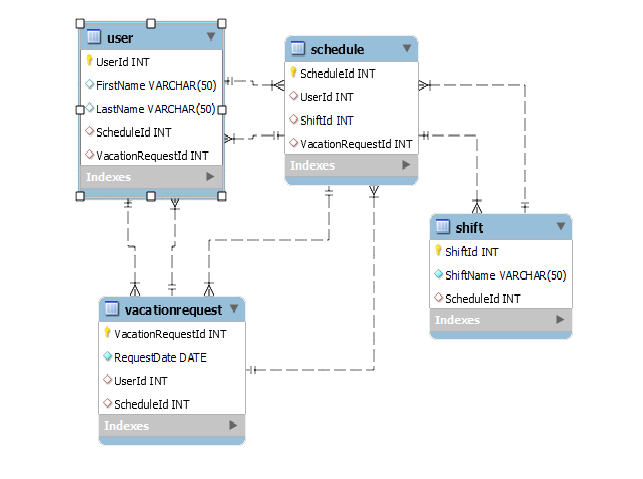
1. Като служител:

* Проверка на длъжност (готвач, сервитьор);
* Проверка за работни дни в седмицата;
* Проверка на смяна за седмица (първа, втора);
* Проверка на служители в същата смяна;
* Проверка на диапазон на смените (от колко, до колко часа);
* Даване на заявка за почивни дни;
  + 1. **Изготвяне на Use cases Diagrams:**
  1. **Хардуерни**

**ТРЕТА ГЛАВА.**

**Описание**

1. **Структура на базата данни**

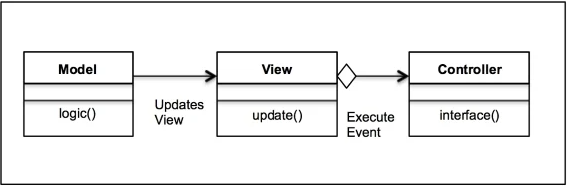
База от данни е интегрирана съвкупност от данни, които са взаимно свързани, съхраняват се съвместно, и с минимално дублиране. Такава база данни осигурява оптимално използване на данни от много потребители отделно или едновременно. База данни се усъвършенства и утвърждава паралелно с развитието на електронно-изчислителна техника и алгоритмичните езици от високо ниво. Заедно с това се създават и системи за управление на база от данни (СУБД). Системи за управление на база от данни са компютърно базирани системи, които осигуряват връзка на потребителя с база от данни.

*фиг.… ER Diagram*

* Таблица User – съдържа информация за потребителите;
* Таблица Shift – съдържа информация за смените;
* Таблица Schedule – съдържа информация за графика;
* Таблица VacationRequest – съдържа информация за почивните дни;

1. **Структура на MVC приложението**

Model-View-Controller(MVC) / Модел-Изглед-Контролер е архитектурен [шаблон за дизайн](https://bg.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD_%D0%B7%D0%B0_%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD" \o "Шаблон за дизайн) в програмирането, основан на разделянето на бизнес логиката от графичния интерфейс и данните в дадено приложение.



*фиг. ... Model-View-Controller*

* **Модел –** ядрото на приложението, предопределено от областта, за която се разработва. Oбикновено това са данните от реалния свят, които се моделират и над които се работи – въвеждане, промяна, показване и т.н. Трябва да се прави разлика между реалния обкръжаващ свят и въображаемият абстрактен моделен свят, който е продукт на разума, който се възприема като твърдения, формули, математическа символика, схеми и други помощни средства.
* **Изглед –**  тази част от изходния код на приложението, отговорна за показването на данните от модела. Например изгледът може да се състои от [PHP](https://bg.wikipedia.org/wiki/PHP) шаблонни класове, [JSP](https://bg.wikipedia.org/w/index.php?title=JSP&action=edit&redlink=1) страници, [ASP](https://bg.wikipedia.org/wiki/ASP) страници, [JFrame](https://bg.wikipedia.org/w/index.php?title=JFrame&action=edit&redlink=1) наследници в [Swing](https://bg.wikipedia.org/w/index.php?title=Swing&action=edit&redlink=1) приложение. Зависи от това какъв [графичен интерфейс](https://bg.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D0%BD_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) се прави и каква платформа се използва;
* **Контролер –** тази част от сорс кода, взима данните от модела или извиква допълнителни методи върху модела, предварително обработва данните, и чак след това ги дава на изгледа. Също така когато се прави уеб графичен интерфейс това би довело до много лесна модификация на [HTML](https://bg.wikipedia.org/wiki/HTML) кода дори от човек, който не е програмист – той ще гледа на шаблона просто като на обикновена HTML страница.

1. **Описание на CRUD операциите**

CRUD функциите обхващат всички основни функции, които трябва да са имплементирани в приложенията, базирани на релационна база данни. Всяка една от буквите на акронима е свързана със Structured Query Language (SQL) изявление – "C" с "Create", "R" с "Read", "U" с "Update" и "D" с "Delete". Съществуват вариации на модела, но всяка една от тях включва тези четири основни функционалности под една или друга форма.

**Използвана литература**

1. ***Програмиране -*** *https://csharp-book.softuni.bg/first-steps-in-programming.html*
2. ***ASP .NET*** *–* [*https://bg.wikipedia.org/wiki/ASP.NET*](https://bg.wikipedia.org/wiki/ASP.NET)
3. ***C#*** *- <https://en.wikipedia.org/wiki/C_Sharp_(programming_language)>*
4. ***CRUD*** *–* [*https://en.wikipedia.org/wiki/Create,\_read,\_update\_and\_delete*](https://en.wikipedia.org/wiki/Create,_read,_update_and_delete)
5. ***HTML*** *- <https://bg.wikipedia.org/wiki/HTML>*
6. ***CSS*** *- <https://bg.wikipedia.org/wiki/CSS>*
7. ***JavaScript*** *- <https://en.wikipedia.org/wiki/JavaScript>*
8. ***Bootstrap*** *- <https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)>*