

ДИПЛОМЕН ПРОЕКТ

**Тема:**

**РАЗРАБОТКА НА DESKTOP ЗА ЗА ПОПУЛЯРИЗИРАНЕ И ОРГАНИЗАЦИЯ НА ДЕЙНОСТТА ЗА СЪЗДАВАНЕ НА ПРИЛОЖЕНИЕ ЗА МУЗИКАЛЕН МАГАЗИН**

**Дипломант: Консултант:**

Василена-Мария Момчилова Минева МиленаХристозова

№ 4, 12А клас

**Специалност:** Системно Програмиране

Сопот - 2024

# Съдържание

**1. Увод**

1. Цел на дипломната работа ..........................................................................................3

2. Структура на дипломната работа ...............................................................................4

**2. Проучване**

1. Използвани технологии и инструменти ……………………………………….. 5

1.2. Създаване на менюта във форма на приложение ………………………………. 9

**3 . Проектиране**

1. Модел на случаи на употреба (Use Case) .................................................................10

1.2. Модел на случаите на употреба на приложението.................................................15

**4. Реализация**

1. Структура на приложението ...................................................................................

2. Реализация на базата данни …………………………………………………….

**5. Заключение** ……………………….……………………………………………..

**6. Използвана литература** ………………………………………………………..….

# Ι. Увод

В последното десетилетие се наблюдава нарастване на обвързаността

между бизнеса и новите технологии. Организациите се стремят да се разрастват,осигурявайки на своите служители гъвкав начин на работа, базиран на висока мобилност, и неограничен от времето и мястото достъп до информация. С развитието на бизнеса нараства и конкуренцията и се увеличава необходимостта от софтуерни решения, които максимално да улеснят бизнес процесите в една фирма. Софтуерните приложения се състоят от много елементи, които обединени помежду си, обменят информация и работят като едно. От особена важност за такива решения е възможността за достъп до централизирана информация навсякъде, по всяко време и от различни устройства. Едни от тези приложения са тези направени на платформата .NET, създадена от Microsoft, която обединява различни технологии, чрез които може да се реализира всеки един компонент от информационни системи, улесняващи бизнес процесите във фирмата. В дипломната работа ще бъде представено как са приложени няколко .NET технологии при създаването на приложение за музикален магазин, отговаряща на бизнес нуждите на фирма „ SoundLaud VM“.

## **1. Цел на дипломната работа**

Дипломната работа си поставя теоритично-практически цели в областта

на разработването на MVP приложение , в което да може да се публикуват всички дейности на фирмата. Приложението предлага възможност за разглеждане на висока гама от висококачествени китари и барабани, клавишни музикални инструменти, озвучителна техника, най-доброто ефектно осветление от всички топ марки на световния пазар.

Веригата музикални магазини „SoundLaud VM“ е една от водещите в браншa вече четири години и отговаря на най-високите стандарти в музикалната сфера и се очертава като символ на престиж, висок професионализъм и качество.

Цели се да постигне прецизно регистриране, изложение на продуктите и техните параметри, което е невъзможно за постигане с досегашната система на ръчна обработка на информацията на хартия и таблици на Microsoft Excel или снимки и параметри разпечатани и поставени в списания с цел показване на различни интрументи, оборудване или осветление. За съхранение на данните трябва да се проектира и реализира релационна база данни(Microsoft SQL Server).

Информационната система трябва да има две клиентски приложения за двата типа потребители, които я използват – *Администратор* и *Клиент*. Достъпът от клиентските приложения до базата данни трябва да се осъществява през XML Web

## **2. Структура на дипломната работа**

В дипломната работа ще бъдат разгледани базата данни на системата и двете клиентски приложения, тъй като са реализирани основно от дипломанта. Дипломната работа се състои от увод, три части, заключение, списък с използвана литература и приложения.

***Увод* – запознаване с темата и целите на дипломната работа.**

***Основна част***

***Първа глава - Проучване*** – Представлява проучването за дипломния проект. Преглеждат се на съществуващи подобни програмни системи и продукти и преглед на известните развойни средства и среди ;

***Втора глава - Проектиране*** – Описание на изискванията към програмния продукт (SRS, use cases), описание на избраната технология и софтуерните средства, потребителски интерфейс (менюта, екрани, прозорци) ;

***Трета глава - Реализация*** *–* Същинската част на дипломния проект, която е с най-голям обем. Да включва описание на начина на реализация на алгоритмите, фрагменти от сорс кода със съответни  коментарии, структура на базата данни (E/R diagram);

***Заключение*** - включва: обобщение на постиженията в дипломната работа; възможностите за неговото приложение;

***Използвана литература*** - представя списък на използваните в процеса на разработка на системата литературни и електронни източници.

**II. Проучване**

1. **Използвани технологии и инструменти**

Настоящата система е разработена на база на едни от последните технологии, предлагани от Microsoft. Изключение прави само NHibernate, който е проект с отворен код и се разработва от независими разработчици.

* ***Microsoft .NET Framework***

Microsoft .NET Framework представлява софтуерен разработъчен фреймуърк, разработен от Microsoft. Този фреймуърк предоставя обширен и последователен програмен модел за създаване на приложения за операционната система Windows, уеб приложения и уеб услуги. .NET Framework включва голяма библиотека с предварително написани решения за обичайни проблеми в програмирането, както и среда за изпълнение за управление на приложенията. Тя е ключов елемент от стратегията за развитие на Microsoft, чиято цел е повечето нови приложения за Windows да бъдат базирани на .NET Framework.

.NET приложенията се пишат на езици от високо ниво (C#, VB.NET, C++/CLI и други) и се компилират до платформено-независим междинен език, наречен CLI (*Common Intermediate Language*).

Основните компоненти и функционалности на .NET Framework включват:

* *Common Language Runtime(CLR)* е среда за изпълнение, която управлява изпълнението на .NET приложенията. Представлява виртуална машина, която компилира междинния код CLI (*Common Intermediate Language*) за конкретната хардуерна платформа и операционна система, с която работи потребителят. В допълнение средата осигурява редица услуги, като управление на паметта, вкл. автоматична система за почистване на паметта , управление на прекъсванията и други.
* *Framework Class Library(FCL)* е колекция от класове и функции, които .NET Framework предлага. Частта от FCL, известна като BCL (*Base Class Library*, 'стандартна библиотека от класове'), е разработената от Microsoft имплементация на стандарта ECMA-335, Common Language Infrastucture. BCL осигурява голям набор базови функционалности, които програмистите могат да използват директно, например за мрежова комуникация, криптографски услуги и други. В допълнение FCL предлага компоненти за изграждане на потребителски интерфейс (Windows.Form и ASP.NET ),уеб приложения и уеб услуги, достъп до бази данни и др.

Основни принципи и цели на .ΝΕΤ Framework включват:

* Една от основните цели на .NET Framework е да даде възможност за създаването на платформено независими приложения, т.е. на приложения, които могат да работят на различни операционни системи и хардуерни конфигурации. Microsoft са създали имплементации на .NET Framework, които работят на x86 и x64 персонални компютри.
* .NET Framework дава възможност за достъп до функционалност, реализирана в приложения извън .NET средата.
* Приложения за .NET Framework могат да бъдат създавани на няколко съвместими програмни езици, сред които C#, VB.NET и C++/CLI. Това е възможно благодарение на съвместимостта на типовете данни, които отделните езици поддържат.
* Улеснена инсталация
* Сигурност
* ***ASP.NET*** е технология за създаване на уебсайтове, онлайн и уеб приложения, а също особено за уеб услуги, разработвана от Microsoft. 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.

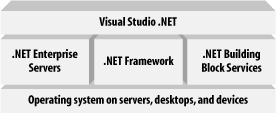
На фиг.1 в файла SampleCode.aspx се съдържа следната информация:

фиг.1 Пример за употреба на ASP.NET

CodeFile уточнява пътя до сървърно изпълнявания файл, който е с разширение в зависимост от .NET езика (C#, VB.NET и др.).

* ***Windows Presentation Foundation(WPF)*** е графична подсистема за визуализация на потребителски интерфейси в Windows приложения. Тя предоставя богат набор от контроли, механизми за разпределение на компонентите и възможности за свързване на данни.
* ***Windows Communication Foundation(WCF)*** е фреймуърк за създаване на услуги, ориентирани към услуги. Той позволява на разработчиците да създават сигурни и съвместими услуги с използване на различни протоколи за комуникация.
* ***Entity Framework*** е фреймуърк за отображение на обекти върху релационни данни. Той опростява достъпа до данни в .NET приложения, като позволява работа с данни във вид на обекти и класове, вместо чрез таблиците в базата данни и SQL заявките.

Ново обявената платформа се състои от 5 основни компонента, които са показани на фиг. 2



фиг. 2 Компоненти на платформата .NET

На най-ниското ниво се намира операционната система, която може да бъде някоя от фамилията операционни системи Windows7,8,10,11 (Windows 2003, Windows XP и др.).

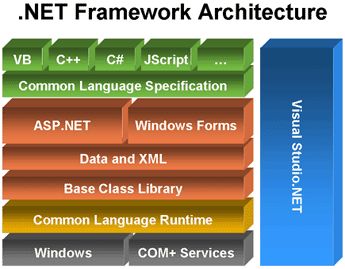
В центъра на .NET платформата се намира .NET Framework – нова среда за разработка и изпълнение на приложения. Тя предлага независимост от използване на езика за програмиране, т.е едно приложение може да бъда написано на няколко различни програмни езици. Освен независимост, се предлага и интеграция между различните езици. Това означава, че разработчикът може да наследява класове, обработва изключения и използва полиморфизъм във взаимодействащи си части на приложението писани на различни езици.

NET Framework се състои от следните части:

***Common Language Runtime (CLR)*** – Тази среда е ключова част от Microsoft .NET Framework и играе важна роля в изпълнението на .NET приложения.

***Framework Class Library (FCL)*** – Представлява стандартна библиотека в рамките на .NET Framework, предоставяща набор от готови за използване класове, типове данни и функции за създаване на различни видове приложения. Тя съдържа хиляди класове, които предоставят реализации на общи задачи, както и поддръжка за различни операции, като вход и изход, обработка на данни, мрежово взаимодействие, работа с изображения, работа със страници, управление на ресурсите и много други.

Архитектурата на .NET Framework е показана на фиг. 3



фиг. 3 Архитектура на .NET Framework

* ***Microsoft SQL Server или Access***

Microsoft SQL Server и Microsoft Access са две различни *системи за управление на бази данни (СУБД)*, предоставящи средства за съхранение, управление и извличане на данни, но с различни характеристики и употреба.

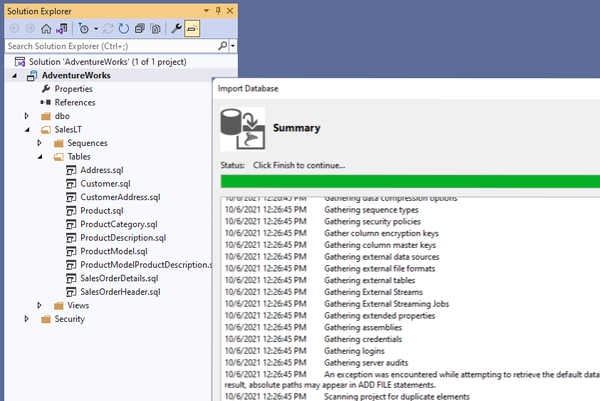
* *Microsoft SQL Server:*

- Е мощна релационна база данни, което значи, че данните се организират в таблици с редове и колони, и се използва структуриран език за запитвания (SQL) за тяхното манипулиране.

- Често се използва за средни и големи приложения с по-големи обеми от данни и по-голяма нужда от сигурност, мащабируемост и производителност.

- Характеристики - поддържа транзакционно управление, сигурност, репликация, мащабируемост, аналитични функционалности и други.

-Употреба - обичайно се използва за управление на данни на приложения, където има нужда от бърза възможност за обработка на данни. Пример за такава база можете да видите на фиг. 4.

фиг. 4 – Пример за свързана SQL Data base

* *Microsoft Access:*

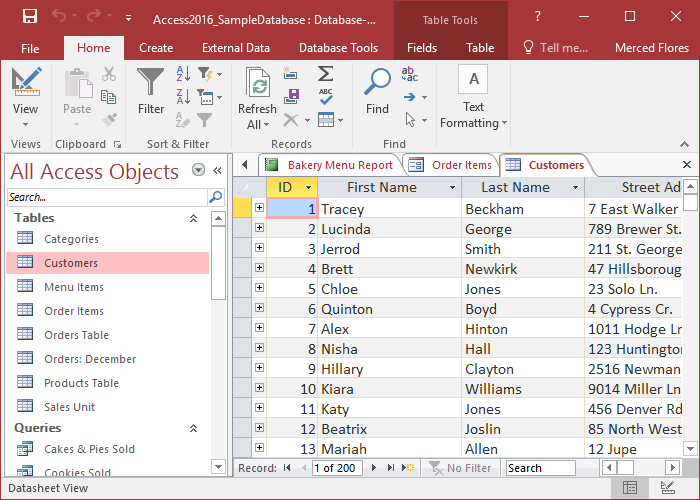
- Access е лека, десктоп база данни с еднофайлов формат. Предоставя средства за създаване на бази данни с таблици, форми, запитвания и отчети.

- Често се използва за малки и средни проекти, които имат нужда от проста и лесна за употреба база данни. Често е интегриран с Microsoft Office приложения като Excel и SharePoint.

-Характеристики - включва базови функционалности за управление на база данни, форми и отчети. Гъвкава и лесна на използване, но обикновено не предоставя мащаба и напредналите функции, каквито предлага SQL Server.

-Употреба - често се използва за индивидуални проекти, малки бизнеси или като лесен начин за създаване на бази данни от неспециалисти в областта на базите данни.

Общо взето, SQL Server и Access са предназначени за различни цели и използват различни модели за управление на данни в зависимост от нуждите на потребителя. На фиг. 5 можете да видите примерна база данни на Access.



фиг. 5 – Microsoft Access DB

**1.2. Създаване на менюта във форма на приложение**

Менютата са важен елемент на формите в едно приложение с графичен потребителски интерфейс. Чрез тях потребителят има възможност за бърз достъп при изпълнение на избрана от него операция чрез директен избор на съответната команда от менюто на приложението. Менютата са организирани в йерархична структура. Даден елемент от меню, сам по себе си може да представлява меню, когато се използва за визуализиране на елементи от подменю.

# III. Проектиране

В дизайна на почти всички приложения, едно от първите решения, които се взимат е как да се имплементира графичния потребителски интерфейс, така че да предостави на потребителите интуитивен подход за работа с програмата.

Реализацията на софтуерни системи с нарастваща сложност, паралелно от множество екипи от специалисти изисква прилагането на общи стандарти и ефективни методологии за разработка.

При изграждането на клиентския слой на системата е използвана библиотеката на .NET Framework за изграждане на прозоречно-базиран графичен потребителски интерфейс - Windows Forms. Windows Forms предоставя възможност за бързо изграждане на потребителски интерфейс тъй като е базирана на RAD(Rapid Application Development) концепцията. RAD е подход за разработка, при който приложенията се създават визуално, чрез сглобяване на готови компоненти посредством помощници и инструменти за автоматично генериране на голяма част от кода. Windows Forms дефинира набор от класове и типове, част от които са използвани при реализирането на двете клиентски приложения на системата:

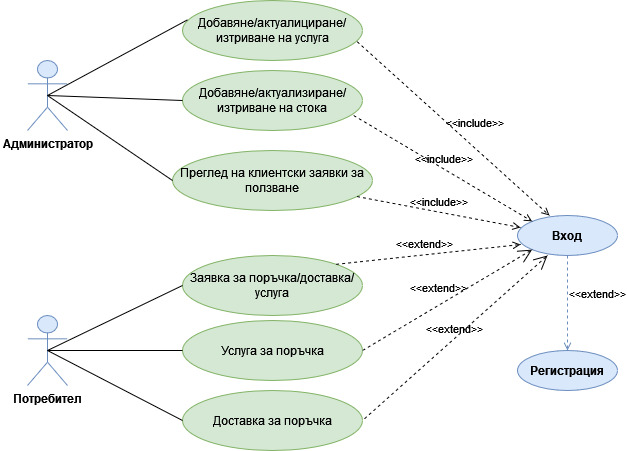
1. **Функционални изисквания на случаи на употреба- (Use Case) модел**

Функционалните изисквания на случаи на употреба, или Use Case моделът, представляват метод за анализ и документиране на софтуерни изисквания чрез описание на интеракциите между различни актьори (потребители или системи) и системата. Те описват различните сценарии, в които системата ще бъде използвана и как ще се държи във всеки от тези сценарии. Ето някои основни елементи на функционалните изисквания на случаи на употреба:

* Актьори: Актьорите са всички участници (потребители или външни системи), които взаимодействат с системата. Те могат да бъдат човек, друга система или дори време. Всяка случай на употреба ще има поне един актьор.
* Сценарии на случаи на употреба: Това са конкретните сценарии или инстанции, в които актьорите използват системата за постигане на определена цел. Всяка ситуация или сценарий се описва като случай на употреба.
* Описание на случаи на употреба: За всеки случай на употреба се предоставя подробно описание на сценария, включително входни и изходни действия, стъпки, актьори, предусловия и постусловия.
* Асоциирани данни: Описват се данните, които се въвеждат, извеждат или манипулират в рамките на сценария.
* Алтернативни пътища и изключения Показват се алтернативни пътища на сценария, както и начините за обработка на изключения и грешки.
* Системни граници: Описват се ограниченията и ограниченията на системата в рамките на конкретния случай на употреба.
* Взаимодействие на актьори със системата: Описва се начинът, по който актьорите взаимодействат със системата чрез въвеждане на данни, стартиране на процеси и получаване на резултати.

Функционалните изисквания на случаи на употреба могат да бъдат представени чрез диаграми на случаи на употреба (Use Case Diagrams) и детайлни описания на отделните случаи на употреба (Use Case Descriptions). Те се използват за определяне на функционалността на системата и за уточняване на изискванията от страна на потребителите.

### **Модел на случаите на употреба на приложението (**фиг.1**)**

****

*фиг. 1 Модел на случаи на употреба – Use Case Model за приложението*

* + АДМИНИСТРАТОР
  + Потребител
* Добавяне/актуализиране/изтриване на услуга: /с Администраторски функции/
  + Снимки на услугата

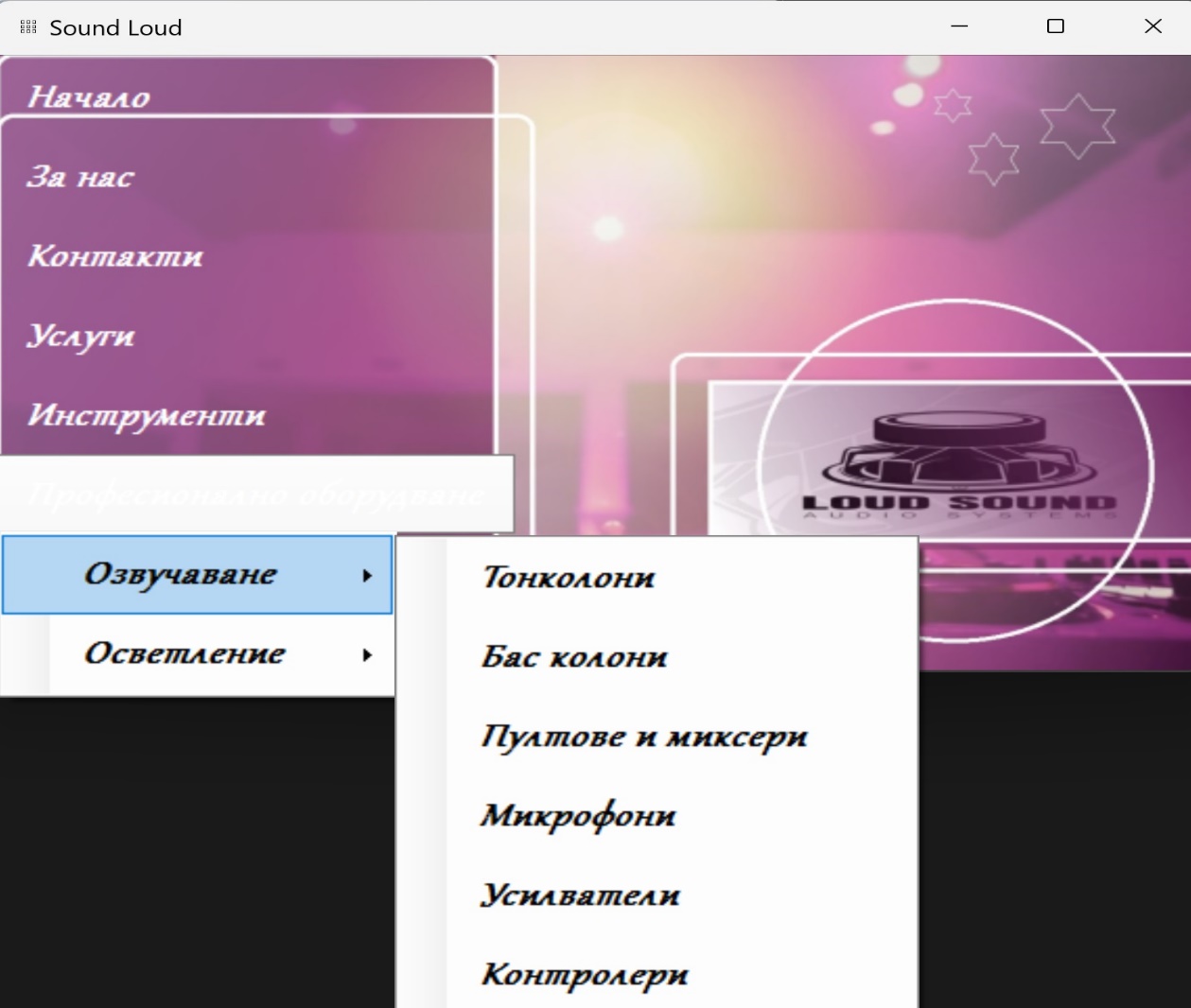
Тип услуга –

* + Цена
  + Вид услуга - инструмент
* Добавяне/актуализиране/изтриване на стоката: /с Администраторски функции/
  + По вид и име на услугата
  + Описание на услугата
  + Дата на вписване в системата на поръчката
  + Начин на плащане
  + Статус на поръчка
* Заявка за поръчка/доставка / услуга – от клиент в меню „марки и фирми“:
  + Име на услугата и име на клиента
  + Поръчка №
  + Дата на приета поръчка
  + Дата на изпълнение
  + Дължима сума
  + Начин на заплащане
* Преглед на клиентски заявки за ползване /с Администраторски функции/. Добре е да се реализира и възможност за отговор /по ел. поща/ към клиента.

**IV.Реализация**

#### **4.1. Структура на приложението**

#### При създаването на приложението първоначално създадох три папки отговарящи на името на изискването ми проекта да е MVP. В папката Model както се вижда на фигура 6 поставих базата данни, в папката View поставих класовете и моделите, а в най-важната папка Presentation поставих изгледният слой или така нареченият презентационен слой. Наименувах всеки елемент подобаващо за да ми е по-лесно при търсене и попълване на данните в класовете.

****Service. На фиг.1 е показана началната страница на приложението, а на фиг.2 форма за вход на потребител.

Фиг.1 – Начална страница и падащо меню с категориите продукти.

Фиг.2 – Форма за вход на потребител в приложението.

**2.Pealizaqiq na BD**

**V.Заключениe**

**VI.Използвана литература**