# 1. Задание

Програмата да предоставя възможност за следене на калории при изяждане на определн вид и количество храна. Да има възможност за работа на много потребители. Да съдържа база данни със историята и възможните продукти. Трябва да могат да се добавят нови храни със въвеждане на съответните им данни/въглехидрати, мазнини и белтъчини/, добавяне на количество изядена храна. Промяна на въведените вече данни в базата от администратора – изтриване на старите или промяна на данните за тях. Да има възможност за изкарване на статистика – за ден, седмица и месец.

# 2. Мотиви за направата на решението.

Съществуват хиляди ,ако не и повече видове диети. Всеки иска да отслабне и да води един по-здравословен живот. Храната е може би най-важната част от всичко. Като трябва да се взема предвид приемането на точната като състав и количество храна. С програмата която направихме, искаме да улесним този процес. Да може потребителя да има всичката му нужна и точна информация на едно място, лесно да борави с нея(с няколко клика да записва необходимата информация). Да може лесно да следи напредъка си и да вижда своята статистика.

# 3. Изисквания

## а) Функционални

### 1. Потребители

- Вход с име и парола

- Изход

- Регистриране на нов потребител

- Промяна при нужда на въведената информация.

#### 1.1. Потребител

- Всеки потребител да влиза със собствен профил

- Добавяне на изядена количество храна

- Изкарване на статистики – за ден, седмица и месец

- Принтиране на статистики – за ден, седмица и месец

- Запазване на статистики в текстов файл – за ден, седмица и месец

#### 2.2. Администратор

- Добавяне на нови продукти към вече съществуващите такива от базата

- Изтриване на продукти от базата

- Промяна на данните за вече въведен продукт във базата.

- Възможност за изглед на всички потребители и промяна на данните им

## б) Нефункционални

- Интуитивен user-friendly интерфейс

- Възможност за лесно избиране на желаната стойност

- Интерфейс не позволяващ направата на случайни грешки от невнимание

- Достатъчно ясна и разбираема графика

# 4. Реализация

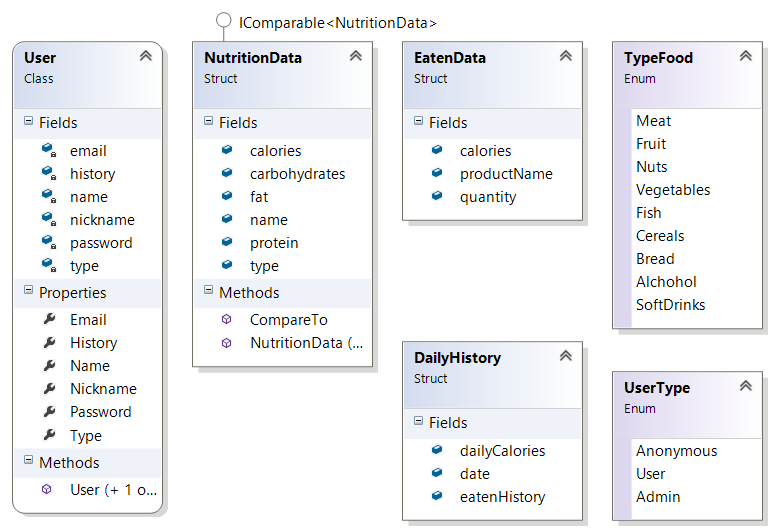
## а) Клас диаграми

### 1. Класове реализиращи достъпа до базата и манипулацията й.

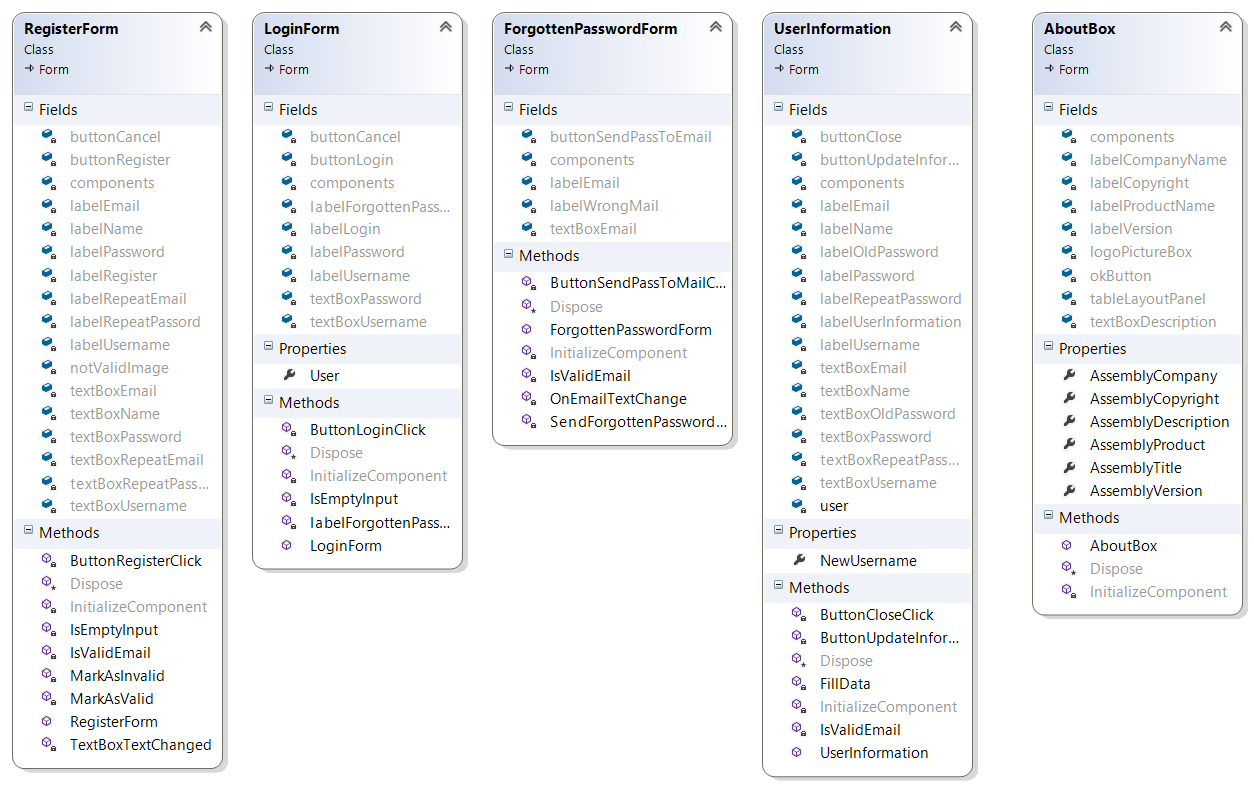
Чрез класа „DBManager“ всички форми и останали класове осъществяват достъпа до базата данни. В класа са реализирани операции за въвеждане във базата, промяна на данните както и проверка за наличност.

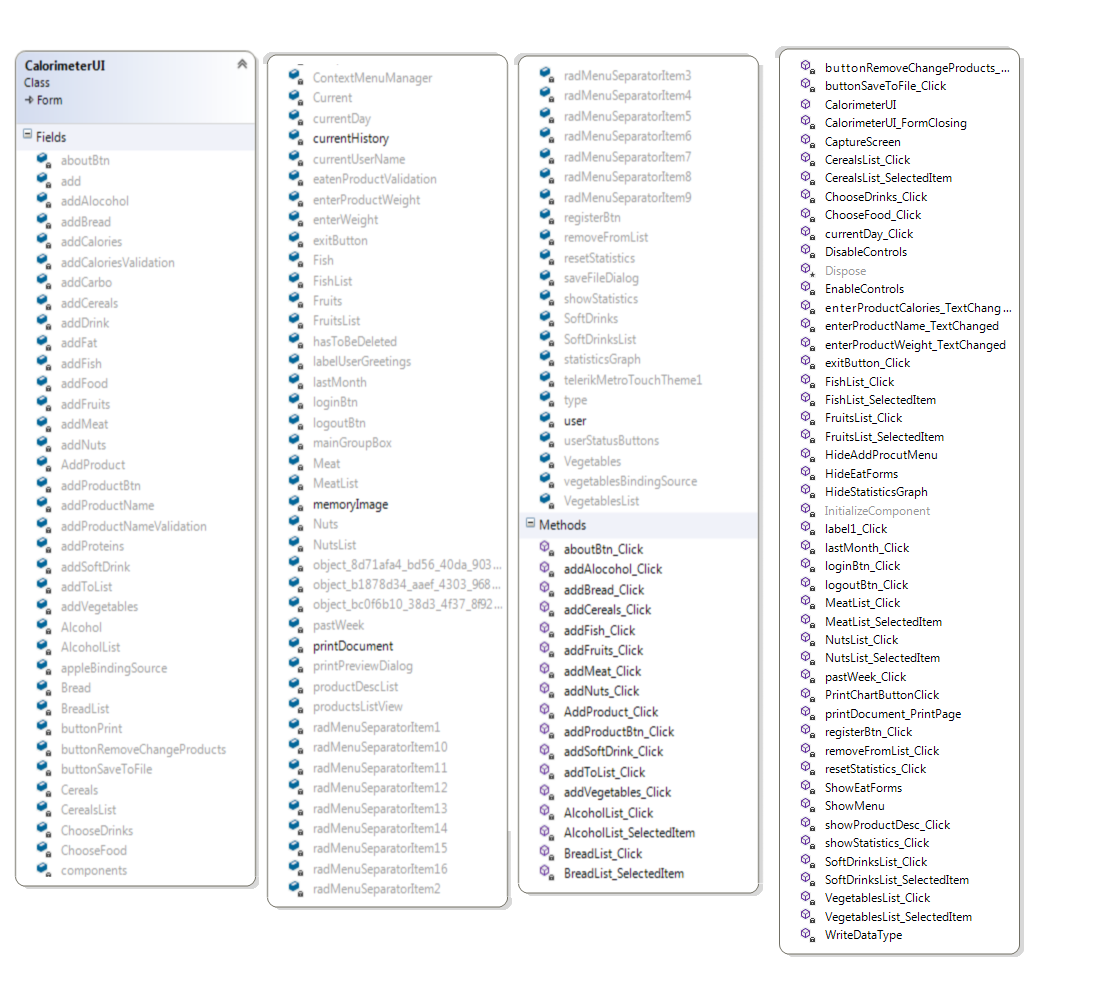
„RijndaelEncryptDecrypt“ класа служи за криптиране/декриптиране на паролите, за да може да не се пазят в чист вид.

“FileManager” класа осъществява записа на статистиката във файл.

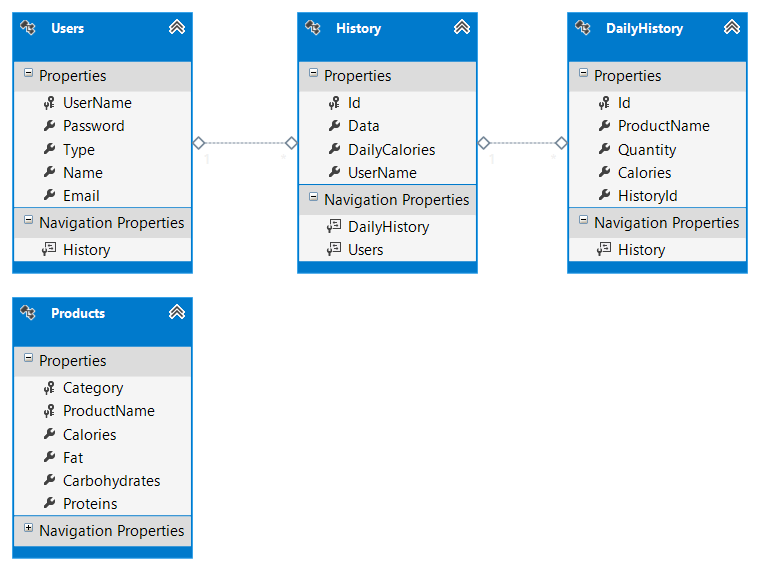
Клас на потребителя и помощни класове за съхранение на данните.

Класа „User“ служи за съхранение на данните на потребителя който е влезнал във системата. „NutritionData, EatenData, DailyHistory“ служат за зареждане на данните от базата, извеждането им към формата и записа на промените в базата.

Класове представляващи „формите“

Основната „форма“ – CalorimeterUI

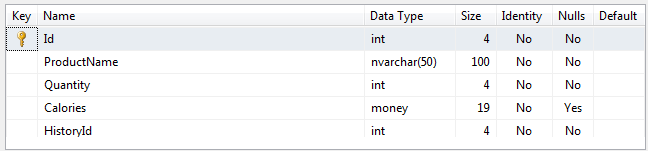
## в) База данни

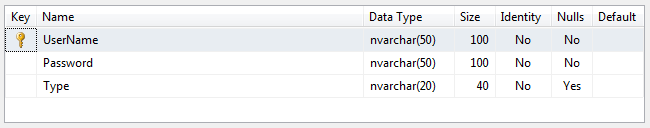
Relationship диаграма на базата данни.

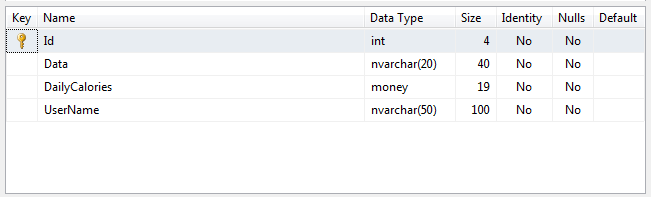
Данните на регистрираните потребители се съхраняват в *Users* таблицата. Всеки потребител има много записи в *History* за всеки ден в който е записал изядена храна. За всяка такава имаме запис в *DailyHistory* , като сумата на дневните калории е сума от записите съответния ден от *DailyHistory.*

По-надолу са показани и структурите на всяка таблица поотделно.

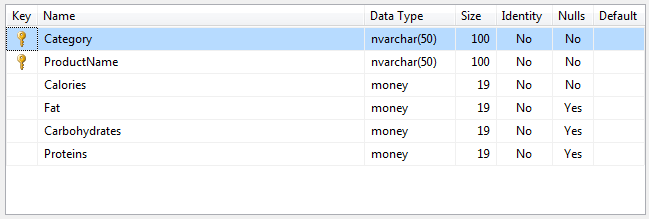
Daily History



Users

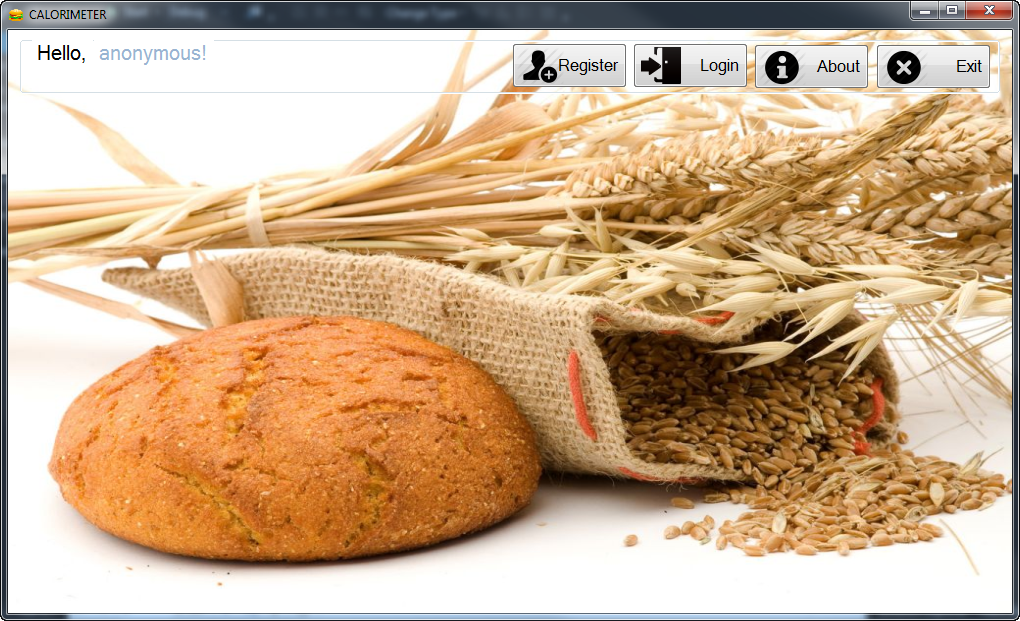
History

Products



# 4. Демо

## Начален прозорец.



Това е прозорецът който се зарежда при стартиране на приложението. От тук потребителя може да се логне ако има регистрация или да си направи такава.

## About форма



## Регистър прозорец



Прозореца който излиза при регистрация. Потребителя трябва да въведе коректно данните си. За всяко от полетата има нужната валидация.

## Логин прозорец.

От тук потребителя въвежда своите данни, за да влезне със профила си в приложението.

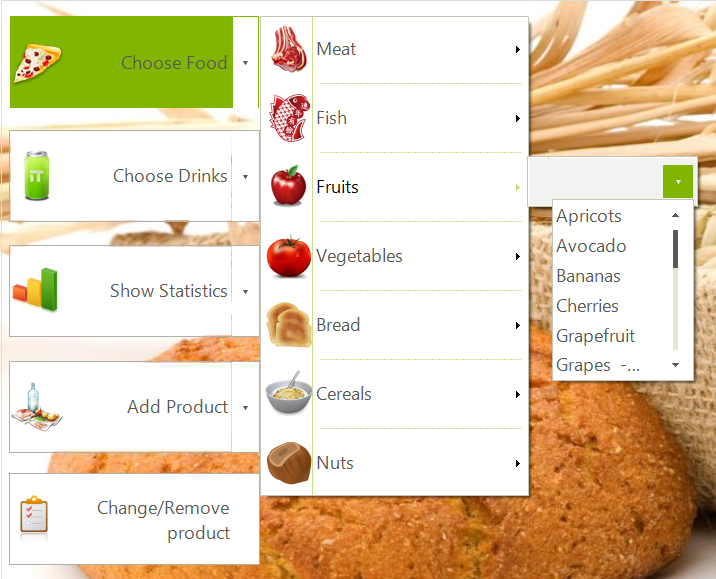
## Прозорец на потребителя.

След логване излиза този прозорец със основните дейности които може да извършва потребителя. В горната лява част се изписва и текущото потребителско име.

## Прозорец на администратора.

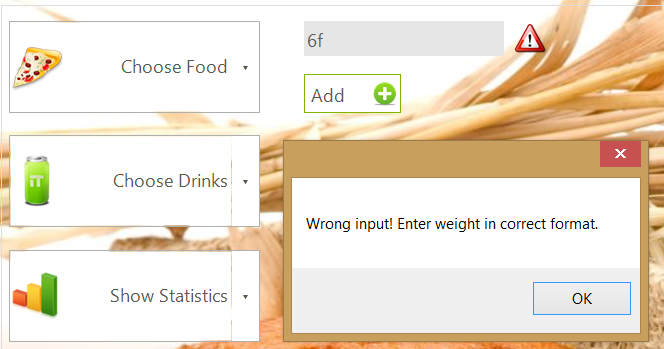
Ако потребителят е от тип Admin, му се показва подобен прозорец, но с допълнителна функционалност. Той има възможност да добавя нови продукти към базата, да вижда всички налични и да променя информацията за въведените вече. Има възможност също и да вижда всички регистрирани потребители, както и да променя данните им.

## Добавяне на изядена храна.



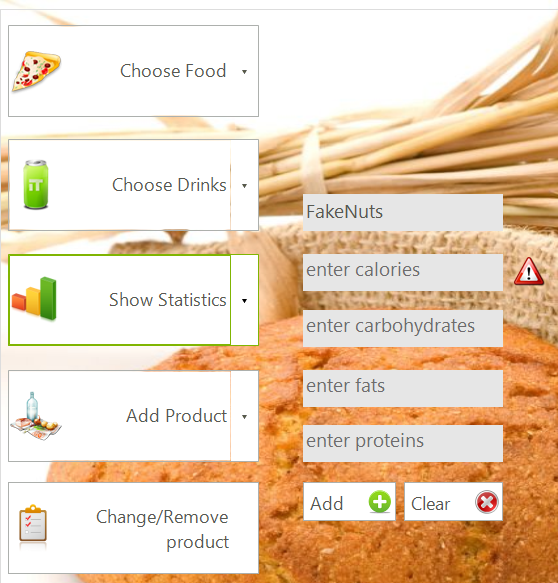
В *Choose Food и Choose Drinks* имаме странично и след това вертикално падащо меню.От там избираме типа храна и конкретния продукт.

След което ни излиза поле в което въвеждаме количеството изядена храна за в грамове и милилитри съответно. При въвеждането на данните имаме и валидация.



## Добавяне на нова храна

Добавянето на нова храна се случва по аналогичен начин. След избиране на категорията към която ще се добавя новия продукт се появяват полетата в които трябва да се въведат нужните данни. За всяко поле има и валидация.



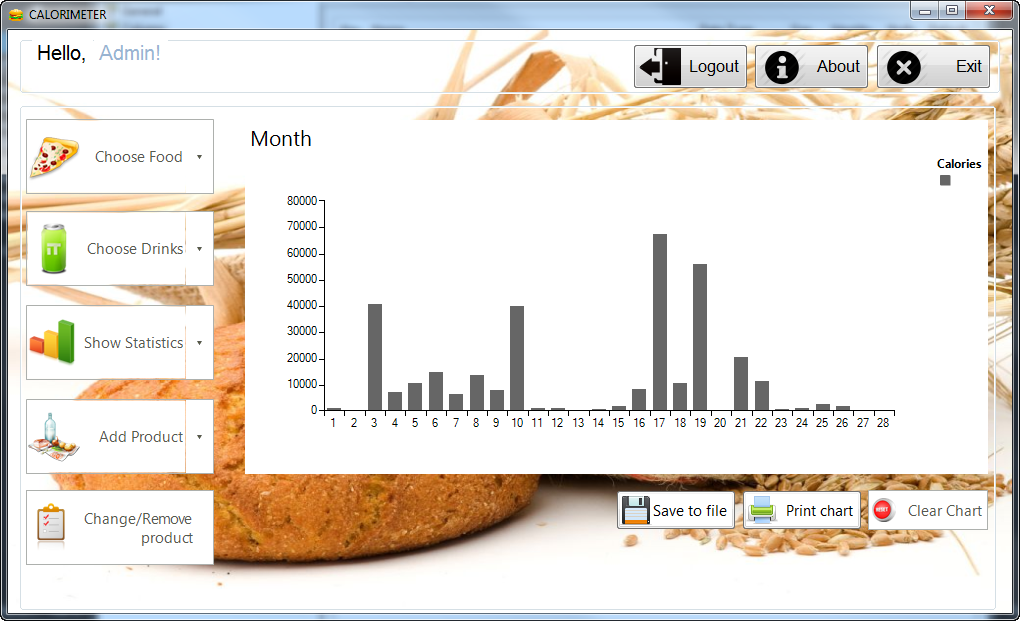
## D:\School\PS-Teamwork\Documentation\Demo\Show statistic buton.pngСтатистика

Чрез *Show Statistic* може да ни се изкарат графиките за ден, седмица и месец, ако са налични.

### 1. Дневна

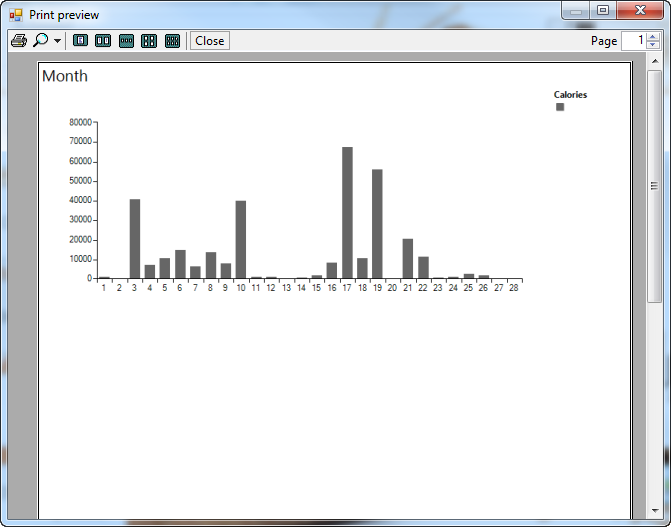
### 2. Седмична

### 3. Месечна



## Промяна/премахване на продукт – админска част

## Принтиране на статистиката



## Запис на статистиката във файл

Тази функционалност позволява записване на статистиката на потребителя във текстов вид. Като има избор между различни формати.

## D:\School\PS-Teamwork\Documentation\Demo\Userinformation window.pngИзглед на регистрираните потребители – админска част

## D:\School\PS-Teamwork\Documentation\Demo\User information.pngИзглед на информацията за потребителя и възможност за променя на данните.

# 5. Бъдещо развитие

# 6. Използвани технологии

* За реализиране на дизайна
* Windows Forms
* Telerik Rad Controls
* Paint.NET
* За реализиране на логиката на приложението
* Програмния език C# версия 5.0
* Платформата .NET Framework 4.5
* За базата данни
* Езикът MySQL
* Microsoft® SQL Server® Compact
* За UML клас диаграмите
* Microsoft Visual Studio 2012