



МИНИСТЕРСТВО  
НА ОБРАЗОВАНИЕТО  
И НАУКАТА



# Математическа гимназия “Академик Кирил Попов”

---

## Korhibuck

*Софтуер за резервации*

---

**Автор:**

Кристина Христова Иванова,

088 465 9873,

Гр. Карлово,

ул. „Ал. Стамболийски“ №25,

[krisiivanova24@abv.bg](mailto:krisiivanova24@abv.bg),

СУ „Васил Левски“, гр. Карлово,

XI клас

Пловдив, 2019

# Съдържание

---

## 1. Цели

- ✓ Анализ на потребностите
- ✓ Предимства

## 2. Ниво на сложност на проекта

## 3. Основни етапи в реализирането на проекта

## 4. Реализация

- ✓ ИЗПОЛЗВАНИ ТЕХНОЛОГИИ ЗА ИЗГРАЖДАНЕ НА ЦЯЛОСТНАТА СИСТЕМА

- Visual Studio 2017
- Photoshop CS6
- MySQL DataBase

- ✓ ЛИЦЕНЗ

## 5. Използвани езици за програмиране

- ✓ C#
- ✓ MySQL

## 6. Описание на приложението

## 7. Развитие и нововъведения

## 8. Заключение

## 9. Използвана литература

- ✓ Книги
- ✓ Линкове

# **1. Цели**

Софтуерът за резервации „Корhibиск“ е предназначен за всички, които искат да резервират място в дадено заведение за хрананене, независимо дали е бар, пекарна, ресторант, т.н. Използвайки създаденото десктоп приложение, се дава възможност за най-бърз и лесен достъп до информацията за всички места, намиращи се в базата данни, и всяка секунда може да се добавя ново. Всяка една информационна операция като: Вход в системата, разглеждане на заведения, резервиране на маса вече става по много по-лесен начин с този софтуер, защото цялата необходима информация е на едно място.

## **1.1 Анализ на потребностите**

Основни дейности в системата са следните:

Имаме отдел катко за потребители, така и за лица, наречени „админи“, които работят във заведенията за хранене. Всеки админ може да запълва, и съответно да освобождава свободни маси и места, за да са наясно клиентите, които

ползват софтуера, колко свободни места могат да резервират.

## **1.2 Предимства**

Софтуерът дава възможност на всеки клиент да направи много лесна и бърза справка за място, където да отиде. Цялата информация се пази в облачно пространство и се правят резервни копия с оглед на сигурността на информацията. Ако клиент влиза в приложението за първи път, то той дава кратка информация за себе си, а следващите пъти влиза в системата по най-удобния и лесен за него начин.

## **2. Ниво на сложност на проекта**

По време на изпълнение на софтуера възникнаха проблеми с настройването и подготвянето на базата данни, концепцията изискваше да бъде с глобален достъп и да дава възможност да се вижда и използва от всеки сървър и клиентска машина. Справих се с този проблем, след като качих базата данни в

облачното пространство и след някои допълнителни настройки по достъпа, успяхм да реализирам облачната функционалност без проблем. Информацията, която се пази в базата данни се подчинява на концептуалния модел за изграждане на релационни БД.

Информация в базата данни, както разгледахме, се основава на концептуалния модел за изграждане на системи за управление на бази данни. Изградени са релации между полетата в БД, които спазват нотацията за нормализацията на данните.

Представям иновативен графичен дизайн за лесна употреба на специализираната функционалност за всеки клиент, който организира информацията и данните на разбираем и интуитивен подход за работа с приложението. Основен въпрос при работата с информацията за резервациите беше сигурността и как да защитя данните от неоторизиран достъп или загуба от неволно изтриване. Това беше реализирано със специални привилегии за всяка част от базата данни. Всеки ден през процеса на изграждане на БД се правят резервни копия на основната

информация, за да може данните да не бъдат изтрети или безвъзвратно изгубени. По базата данни и цялостното изпълнение за в бъдеще предстоят още няколко поправки и оптимизации. При всяка обработка на операциите от потребителите и админите се извеждат подходящи съобщения - достатъчно ясно формулирани и описващи последствията от дейността на обработка на информацията.

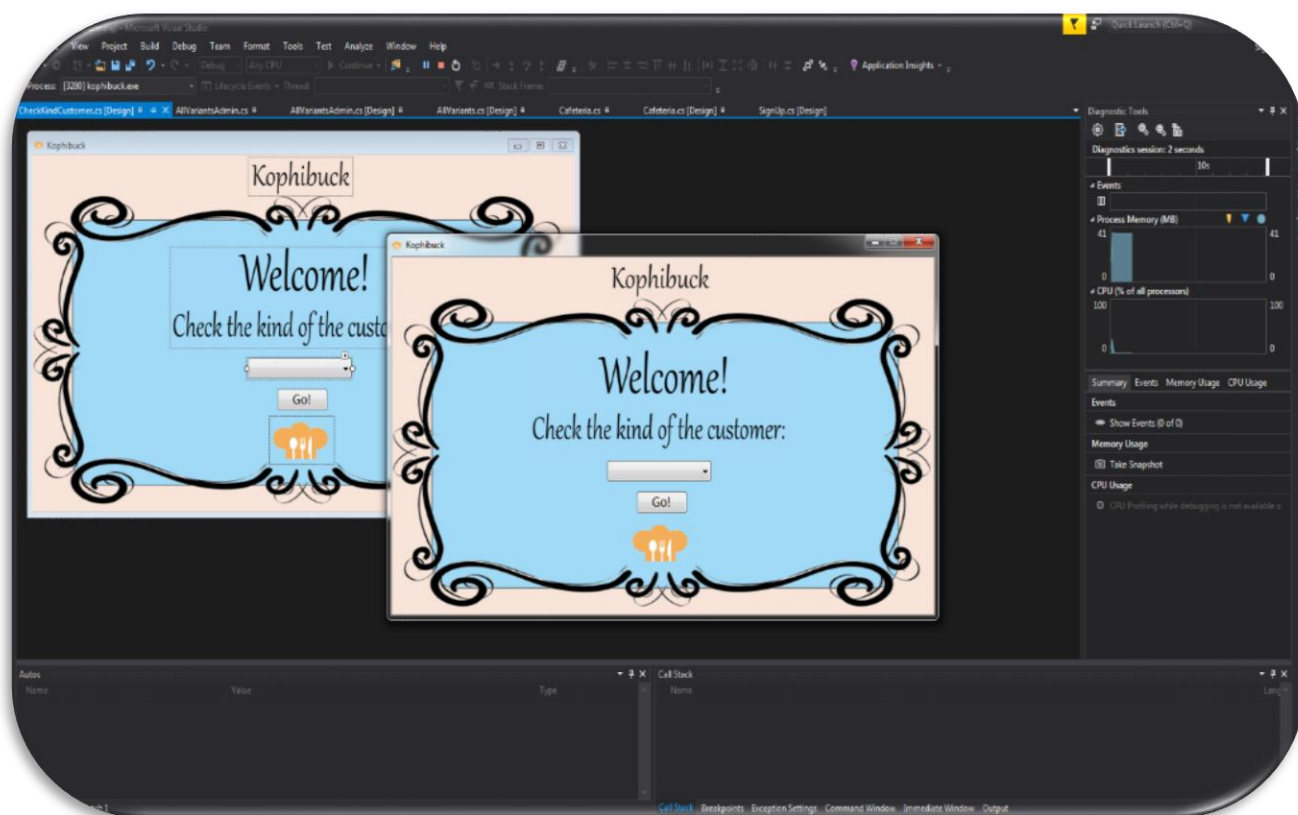
### **3. Основни етапи в реализирането на проекта**

Софтуерът „Korhibusck“ е изграден от няколко модула: форма за влизане в системата, съответстваща на това дали лицето е клиент или работник в дадено място, а спрямо това се променят и функционалностите.

#### **Логин форма – „Вход в системата“**

Формата за влизане е максимално оптимизирана с цел най-малко извършване на

операции и забавяне. Всеки, първоначално избира какъв вид потребител е:



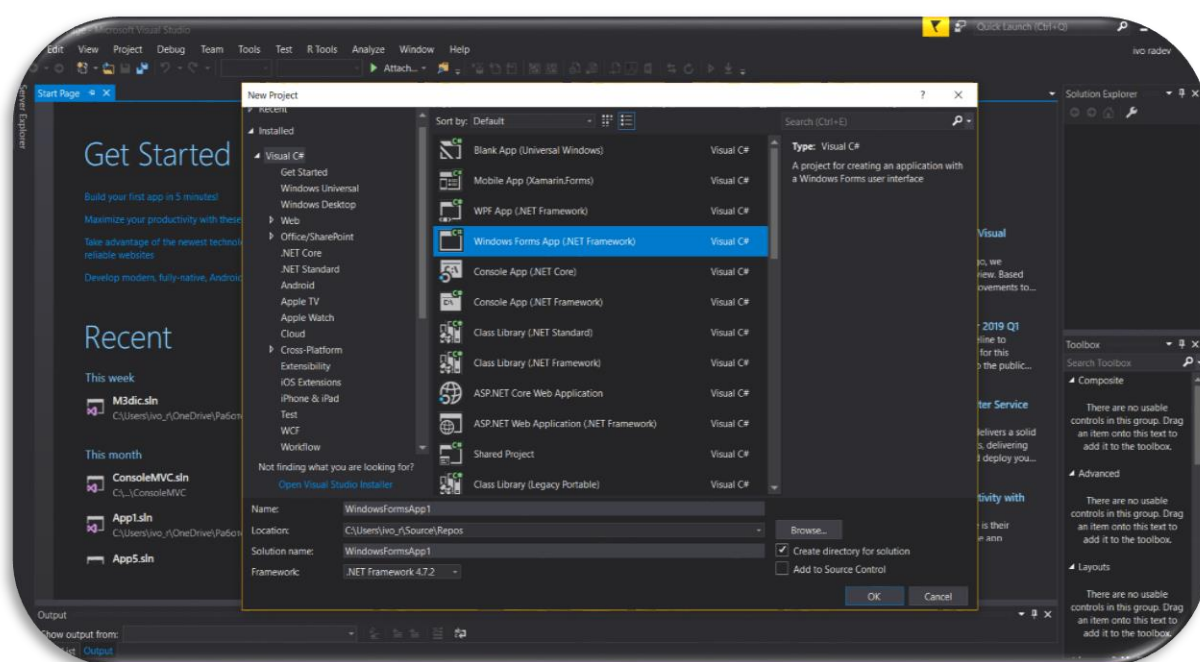
## 4. Реализация

Използвани технологии за изграждане на цялостната система

### 4.1. Visual Studio 2017

Visual Studio 2017 е мощна интегрирана среда за разработка (на английски: *integrated development environment, IDE*) на софтуерни приложения за Windows и за платформата .NET Framework. Използва се за разработка на

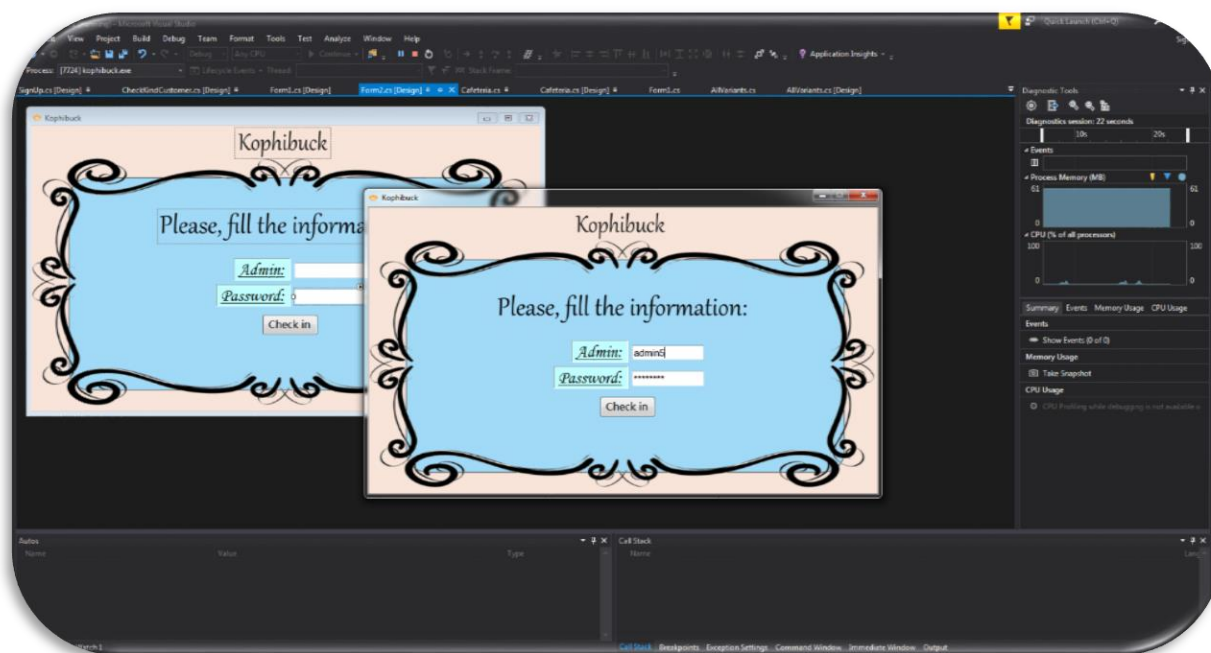
конзолни и графични потребителски интерфейс приложения, както и Windows Forms или WPF приложения, уеб сайтове, уеб приложения и уеб услуги на всички поддържани платформи от Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework и Microsoft Silverlight. Visual Studio предоставя мощна интегрирана среда за писане на код, компилиране, изпълнение, дебъгване (както за високо така и за машинно ниво), тестване на приложения, дизайн на потребителски интерфейс (форми, диалози, уеб страници, визуални контроли и други), моделиране на данни, моделиране на класове, изпълнение на тестове, пакетиране на приложения и стотици други функции.



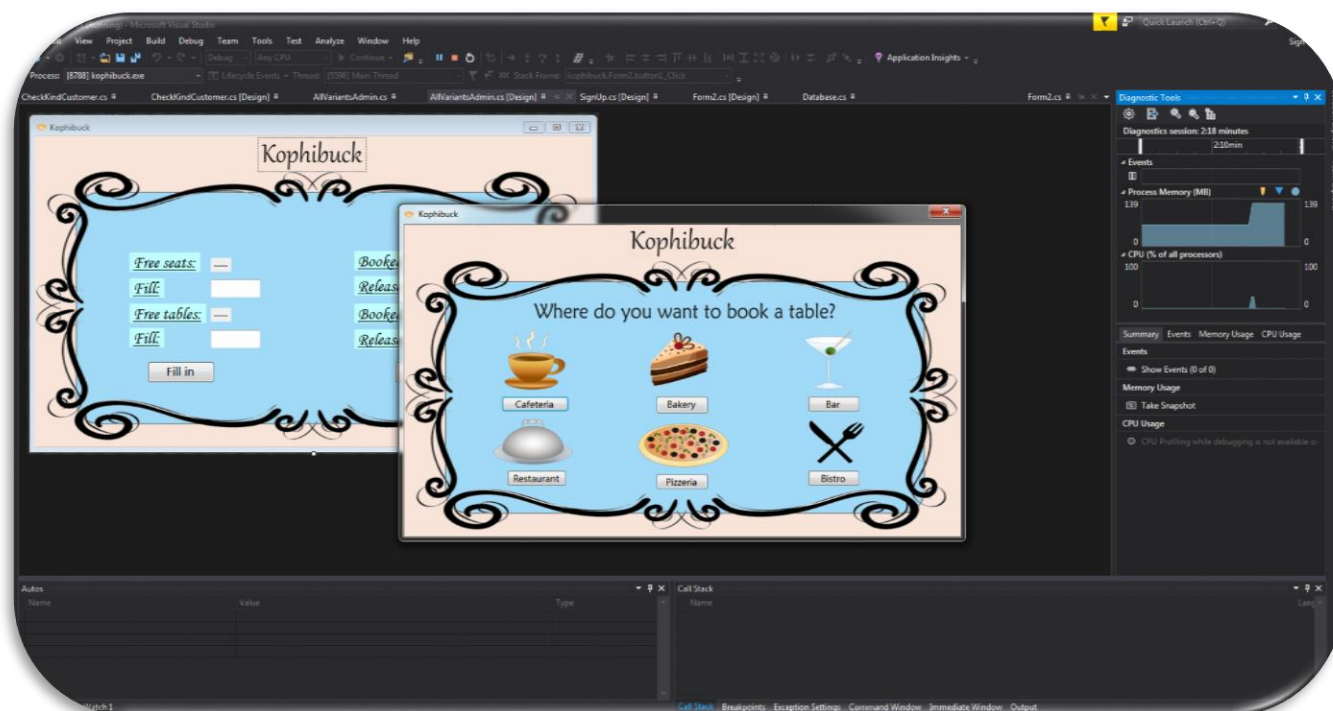


Така изглежда средата за разработка “Visual Studio 2017”. От тук правим нов проект и в нашия случай софтуерът е разработен на Windows Form Application. Използваме най-новия framework - .Net 4.7.2 и C# като език за програмиране на софтуера.

Form Application се използва за създаването на приложения с графичен потребителски интерфейс. Този дизайнер позволява да се добавят и оформят различни менюта и бутони. Полетата, показващи някакви данни, могат да бъдат свързвани с различни източници на данни като бази данни или заявки. Тези полета се добавят чрез изтегляне от прозореца Data Sources върху създадения формуляр. Потребителският интерфейс задължително се свързва с програмен код за създаването на приложение.



Така изглежда главната “Login”-формата за админите в среда за разработка. Имам бутони, места за попълване на текст (textbox), label-и. Отдясно имаме Solution Explorer където се намират всички класове с код и формите – визуално представени. Имам toolbox, през който се добавят нови бутони, панели и т.н. Всички полета са празни и информацията от тях се обработва по начин, по който да бъде разбрана от кода.



Това е формата, където клиента избира в какъв вид заведение за хранене да запази място. За всеки бутон и цялата функционалност ще опиша подробно по-надолу в документацията. Зад всеки бутон стои написан код, част от бизнес слоя на приложението, без който системата не би функционирала.

#### **4.2. Photoshop CS6**

Photoshop (наричана разговорно просто Фотошоп), е професионална комерсиална програма за обработка на растерна графика от американската софтуерна компания Adobe Systems. Photoshop позволява интерактивна

редакция на сканирани и цифрово заснети графични материали в реално време чрез набор от инструменти. В съвременната си форма програмата поддържа над 20 графични файлови формата и интерфейсът ѝ е преведен на дузина езика. Има собствен файлов формат – PSD (съкращение на Photoshop Document), който запазва всички атрибути, използвани по време на работа (например различните видове слоеве (Layers) и маски (Masks), и история на промените (History). Photoshop е една от първите, а и досега остава водеща програма за редакция на дигитални (цифрови) изображения. Нейното име е станало нарицателно, примерно: „Какво се чудиш на тази снимка – фотошопната е!“. Тоест, често се получава, че това, което се вижда, не е истинска фотография, а е резултат на майсторски компютърен ретуш (retouch) и даже графичен монтаж. Културата на дигиталните художници води до появата на термини като цифрова фотоманипулация (photomanipulation) и дигитално рисуване (digital painting) и цифрово акрилоподобно рисуване (airbrushing).

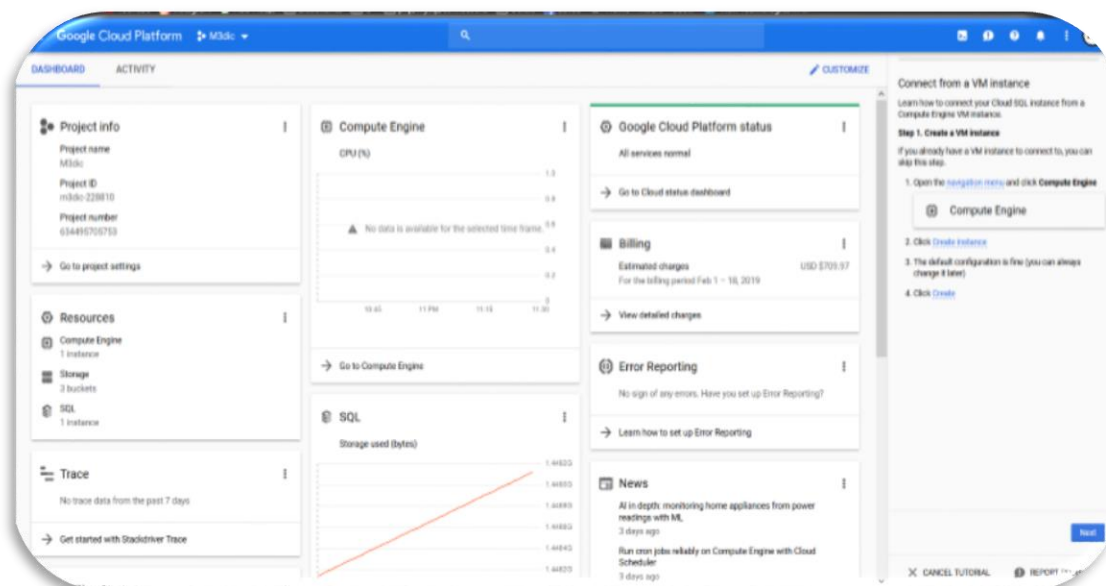


## 4.3 MySQL DataBase

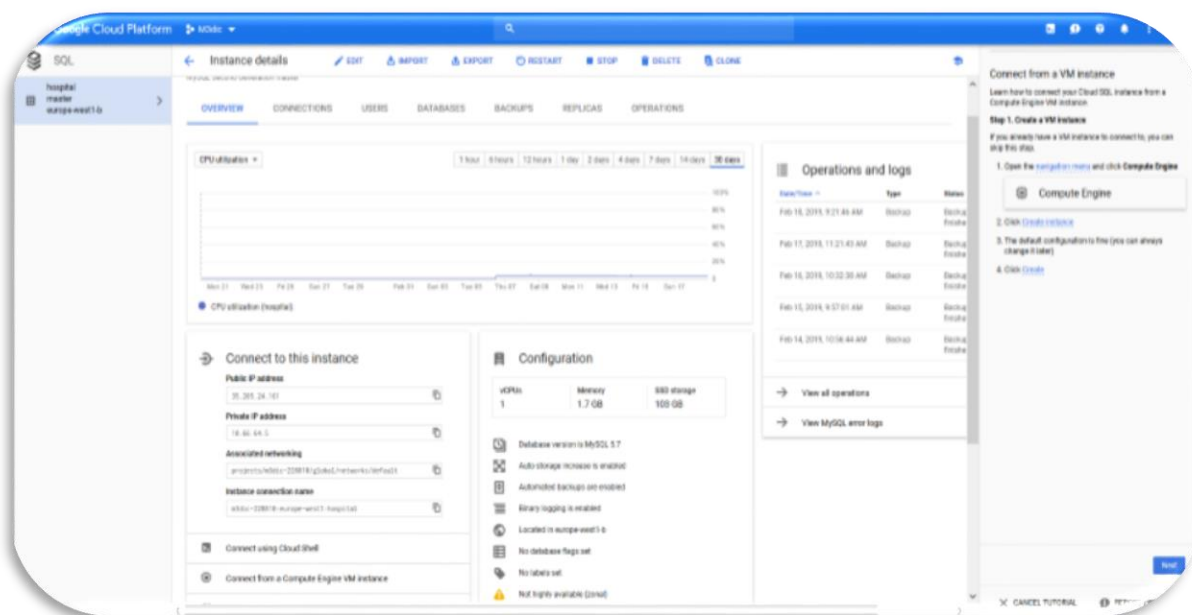
### Представяне

*MySQL DataBase - Workbench 8.0 CE / HeidiSQL е визуален инструмент за проектиране на бази данни, който интегрира SQL разработката, администрирането, проектирането на база данни, създаването и поддръжката в една интегрирана среда за разработка на базата данни на MySQL. Той е наследник на DBDesigner 4 от fabFORCE.net и*

замества предишния пакет софтуер, MySQL GUI Tools Bundle .



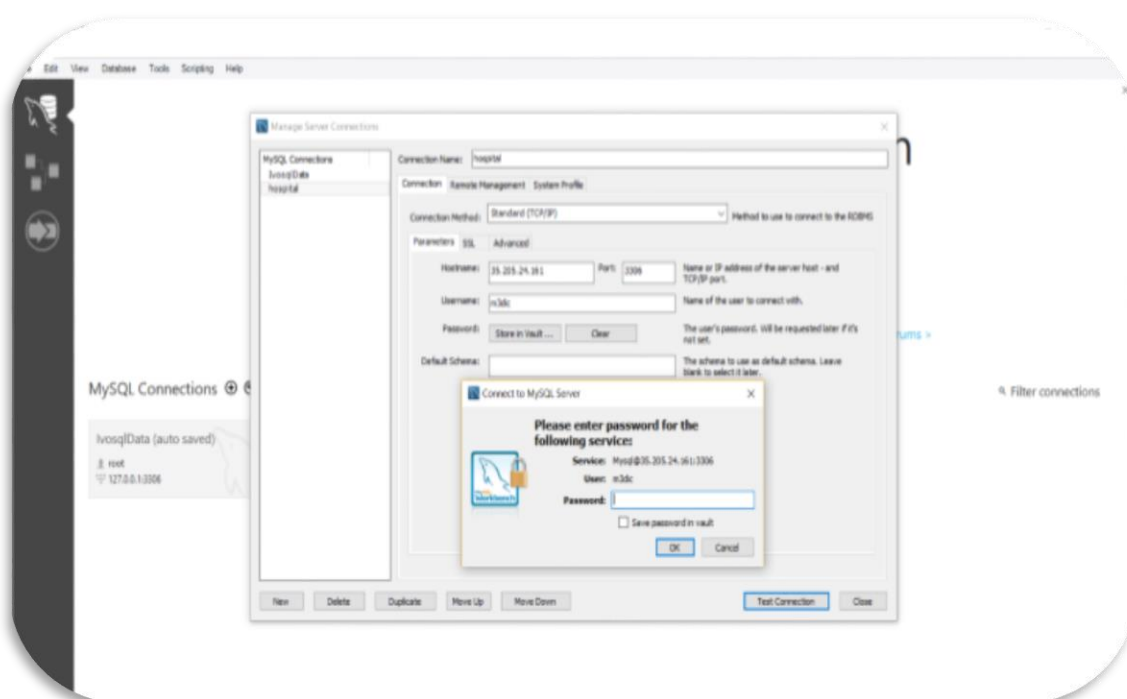
Използвам база данни качена на облачното пространство, а не чрез “localhost” – тоест базата ми данни има възможност да се ползва от всички, навсякъде. Това може да бъде променено – можем да позволим само на конкретни ip-адреси. Възползвах се от едногодишен безплатен хост на базата данни през Google Cloud Platform.



Това е базата данни - тук имаме инстанция, която пуска базата и конфигурация на машината. За момента базата е конфигурирана на най-ниска скорост понеже се ползва само за проби и тествания. Имаме “connections”, където задаваме типа на осъществяването на връзка. В нашия случай имаме `private ip` и `public ip`, които се ползват за сдвояване между базата данни и софтуера.

Системата иска да дадем позволение на конкретни `ip`-та, които да имат възможността да се ползват от нея, за по-голяма сигурност, но за момента сме задали “0.0.0.0/0”, което дава възможност на всички хостове от всяко местоположение да се

свържат към базата. Можем да направим връзката сертифицирана връзка за защита на заявките чрез SSL сертификата. Имаме “Users”, където задаваме позволения на софтуера да се свърже с базата данни. “Backups” позволява на базата данни да прави автоматични запаметявания на информацията, с цел тя да не бъде изгубена и ако се случи нещо с нашите данни, можем да ги възстановим. “Backups” са зададени автоматично да се правят всеки ден.



### Връзка с БД и Workbench

Ето как осъществявам връзка с базата данни през Workbench. Използвам Workbench за



структурирането на базата данни и за направата ѝ. На снимката се вижда средата за разработка. Отворена е връзка, зададено е `public ip` и е осъществена връзка към базата. Сега се иска да въведем паролата, която се задава в секция “Users” на `google clouds`, и влизаме в базата данни. Стандартния “connection method” е “TCP/IP” със стандартно `ip` и порт по подразбиране.

### Лиценз

`Workbench` е свободноразпространяваща се и предлага цялата необходима функционалност за изграждане на БД.

## 5. Езици за програмиране

### 5.1 C#

C# е обектно-ориентиран език за програмиране, разработен от Microsoft, като част от софтуерната платформа .NET.

Стремежът още при създаването на C# езика е бил да се създаде един 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 (Visual Basic и Java). Те представляват съвкупност от дефиниции на класове, които съдържат в себе си методи, а в методите е разположена програмната логика – инструкциите, които компютърът изпълнява.*

### *Версии на езика:*

| C#     |                |                      |  |
|--------|----------------|----------------------|--|
| Версия | Дата           | .NET Framework       | Visual Studio                            |
| C# 1.0 | Януари 2002    | .NET Framework 1.0   | Visual Studio .NET 2002                  |
| C# 1.2 | Април 2003     | .NET Framework 1.1   | Visual Studio .NET 2003                  |
| C# 2.0 | Септември 2005 | .NET Framework 2.0   | Visual Studio 2005                       |
| C# 3.0 | Август 2007    | .NET Framework 3.5   | Visual Studio 2008<br>Visual Studio 2010 |
| C# 4.0 | Април 2010     | .NET Framework 4     | Visual Studio 2010                       |
| C# 5.0 | Август 2012    | .NET Framework 4.5   | Visual Studio 2012<br>Visual Studio 2013 |
| C# 6.0 | Юли 2015       | .NET Framework 4.6   | Visual Studio 2015                       |
| C# 7.0 | Март 2017      | .NET Framework 4.6.2 | Visual Studio 2017                       |
| C# 7.1 | Август 2017    | .NET Framework 4.6.2 | Visual Studio 2017                       |

### **Методи**

*Чрез употребата на методи в C#, при реализирането на даден код, се постига разделяне на множество подпрограми (subroutines), сглобени в една обща програма.*

Целта на това разделяне на методи е, програмния код да бъде по-ясно четим, което от своя страна води до по-лесно писане на програмата, намиране на неточности и преправяне на вече написания код.

Най-общо принципа за дефиниране на методи в C# е: всеки метод да върши конкретна задача, името му да бъде показателно за работата която върши, като ако поради някаква причина метода не успее да изпълни работата, за която е предназначен, трябва да хвърли грешка (exception). Обикновено, размера на кода на един метод не трябва да е повече от един екран.

Друга много важна причина, заради която е добре да се използват методи е, че по този начин се избягва повторението на код. Това е пряко свързано с концепцията за преизползване на кода.

### **Правила за именуване на метод**

Името на методите трябва да започва с главна буква. Трябва да се прилага правилото „PascalCase“: (всяка нова дума, която се долепя

като част от името на метода, трябва да започва с главна буква) Имената на методите е препоръчително да бъдат съставени от глагол или от глагол и съществително име.

### **Класове и обекти**

C# е измислен да бъде лесен за работа, универсален, съвременен обектно-ориентиран език за програмиране от високо ниво.

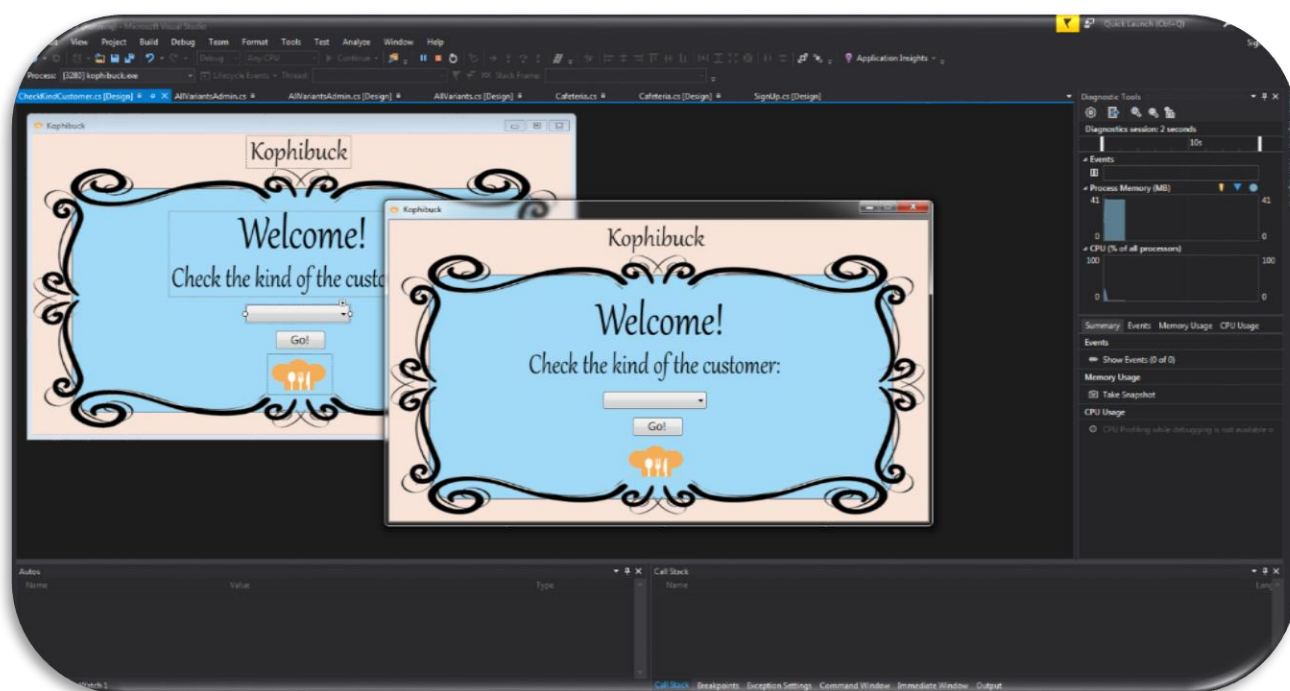
Обектно-ориентираното програмиране е модел на програмиране, който използва обекти и техните характеристики, и взаимодействия за изграждането на компютърни програми. Софтуерните обекти моделират обекти от реалния свят или абстрактни концепции, които също се разглеждат като обекти.

Класът дефинира абстрактните характеристики на даден обект. Той е план или шаблон, чрез който се описва даден обект. Класовете са градивните елементи на ООП и са неразделно свързани с обектите. Всеки обект е представител на точно един клас.

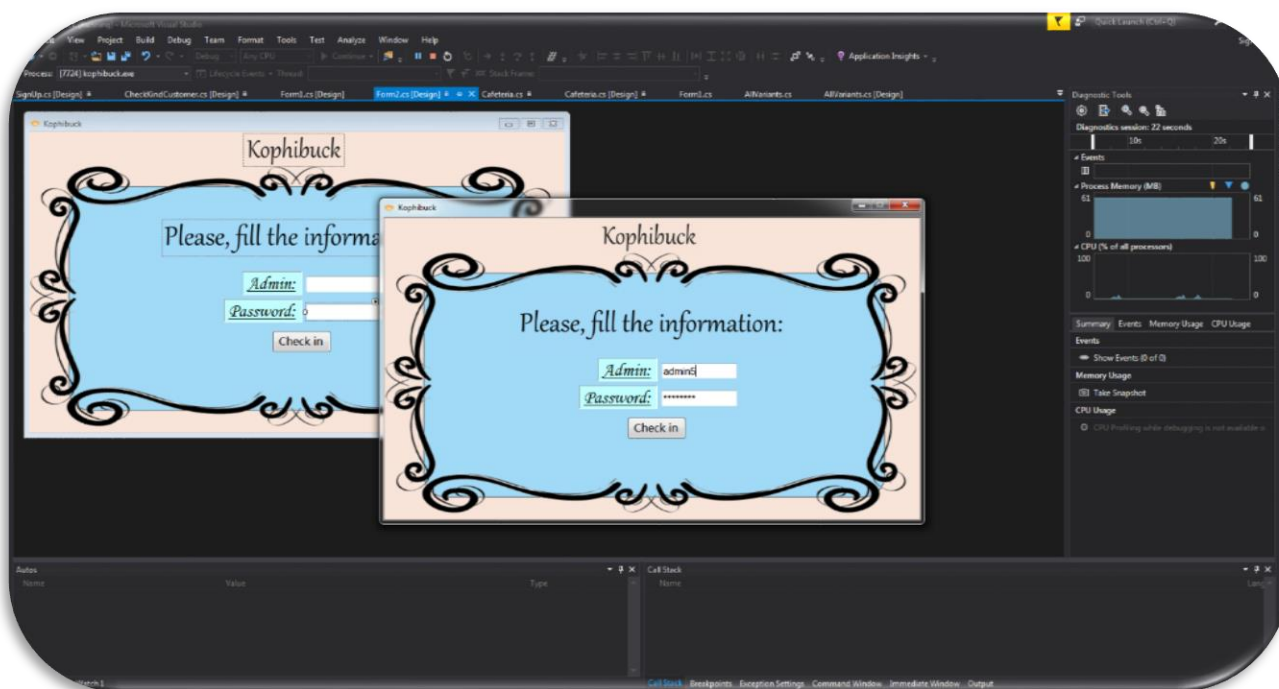
## 5.2 MySQL

MySQL е многопоточна, многопотребителска, SQL система за управление на бази данни (СУБД) с повече от шест милиона инсталации. MySQL AB разпространява MySQL като свободен софтуер под GNU General Public License (GPL), но също така под традиционните за комерсиален софтуер лицензи за случаи, когато използването е несъвместимо с GPL. Подобно разпространение е известно като двойно лицензиране. MySQL се разработва, разпространява и поддържа от Шведската компания MySQL AB, която държи авторските права за голяма част от програмния код. Подобен е моделът на JBoss, а също и начинът, по който Free Software Foundation работи с авторските права по нейните проекти в отличие от Apache проекта, където софтуерът е обществено разработван, а авторските права за програмния код принадлежат на индивидуалните автори.

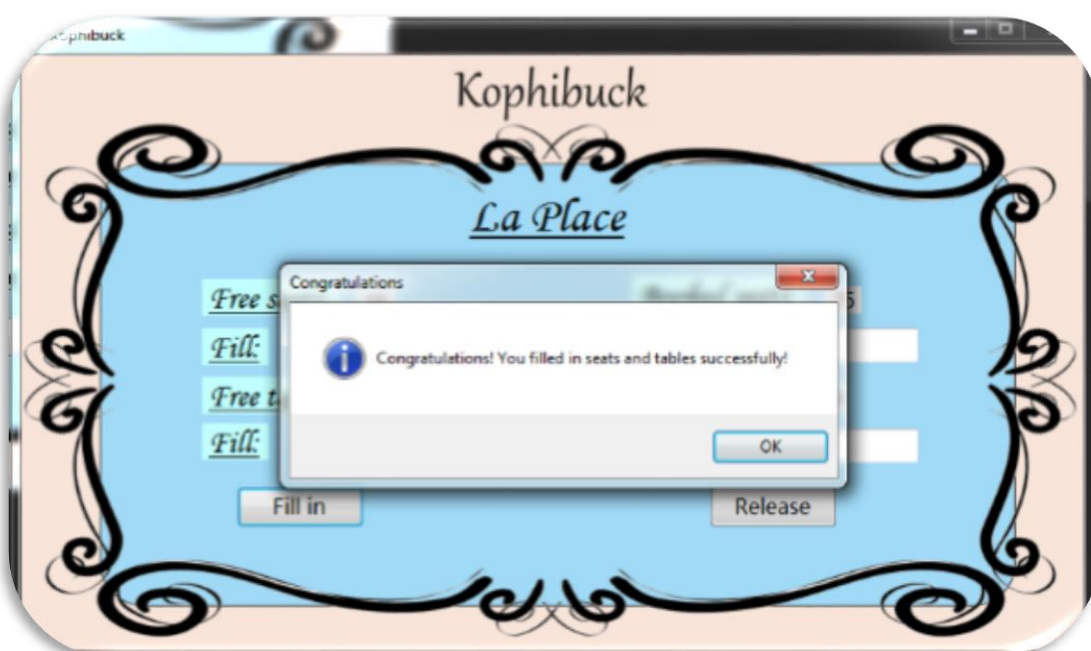
## 6. Описание на приложението



Това е разработването на първата форма на софтуера, където от падащо меню потребителят избира дали ще влезе в системата като „админ“ (работник в съответно заведение) или клиент, който ще запазва места. Ако потребителят не е избрал опция, но натисне бутона, се извежда подходящо за случая съобщение.



Това се извежда, когато бъде избрана опцията “Admin”. При въведени коректни данни излиза съобщение, че лицето е влязло в системата и му се отваря нов прозорец, а в противен случай – излиза съобщение, гласящо за некоректните данни.

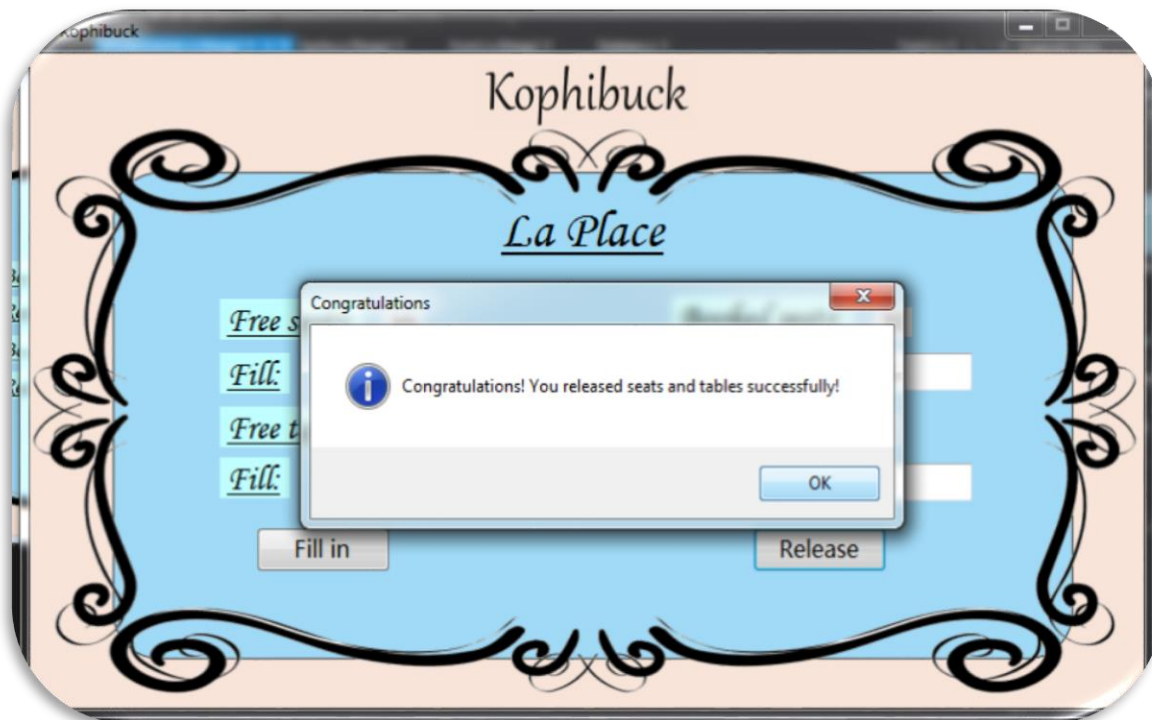


The screenshot shows a web application window titled 'Kophibuck' with a sub-header 'La Place'. The interface is framed by a decorative black scrollwork border. Inside, there are two main sections for managing seating. The left section is for 'Free seats' and 'Free tables', both currently showing counts of 22 and 12 respectively. Each has a 'Fill:' label and an empty input field. The right section is for 'Booked seats' and 'Booked tables', also showing counts of 22 and 12. Each has a 'Release:' label and an empty input field. At the bottom of each section are buttons labeled 'Fill in' and 'Release'.

| Category      | Current Count | Action Label | Action Input         | Action Button |
|---------------|---------------|--------------|----------------------|---------------|
| Free seats    | 22            | Fill:        | <input type="text"/> | Fill in       |
| Free tables   | 12            | Fill:        | <input type="text"/> |               |
| Booked seats  | 22            | Release:     | <input type="text"/> | Release       |
| Booked tables | 12            | Release:     | <input type="text"/> |               |

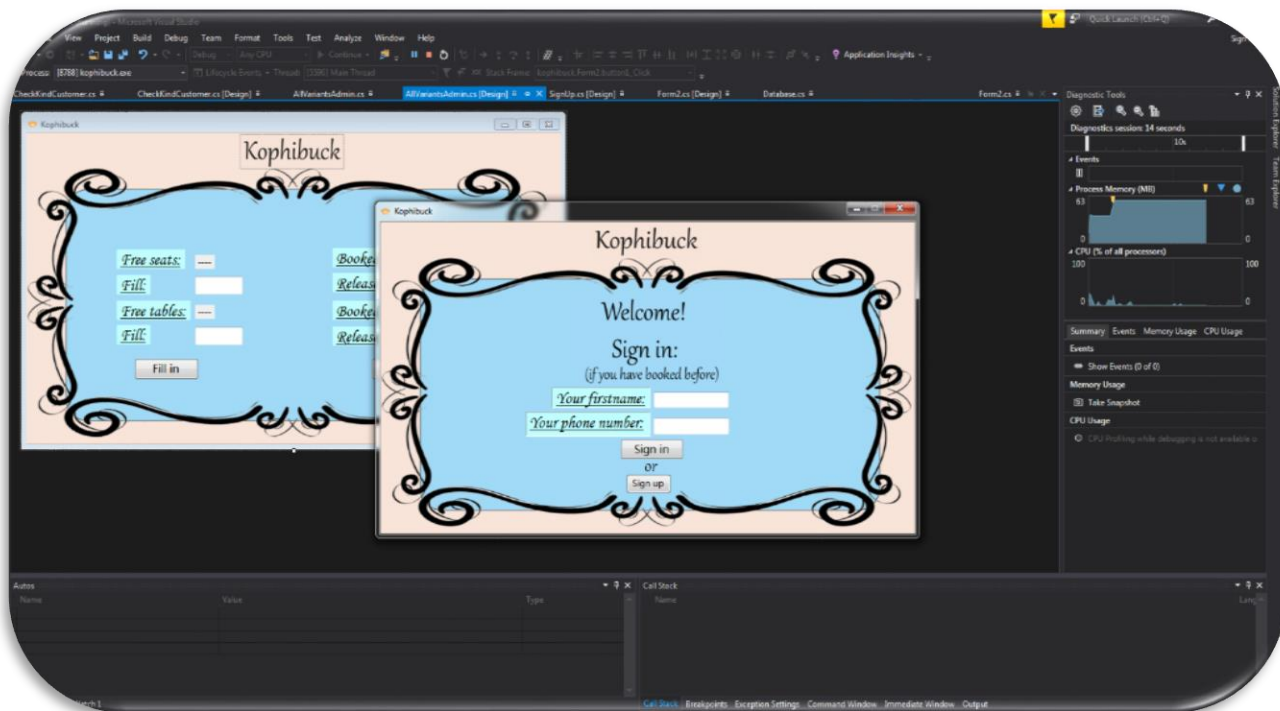
Това е формата, чрез която админа на заведението за хранене управлява броя на местата и масите. В базата данни в информацията на админа пише къде той работи (в кое заведение, като това става чрез id-то на мястото) и какъв тип е това заведение. Връзката със заведението става чрез тези две полета – от съответната таблица (вид заведение) се избира id-то на мястото и в label-ите излизат името, свободните маси и места. След като админа е въвел колко маси и места иска да освободи или да напълни, при успешно update-ване на информацията в базата данни, излиза съобщение, гласящо за успешната





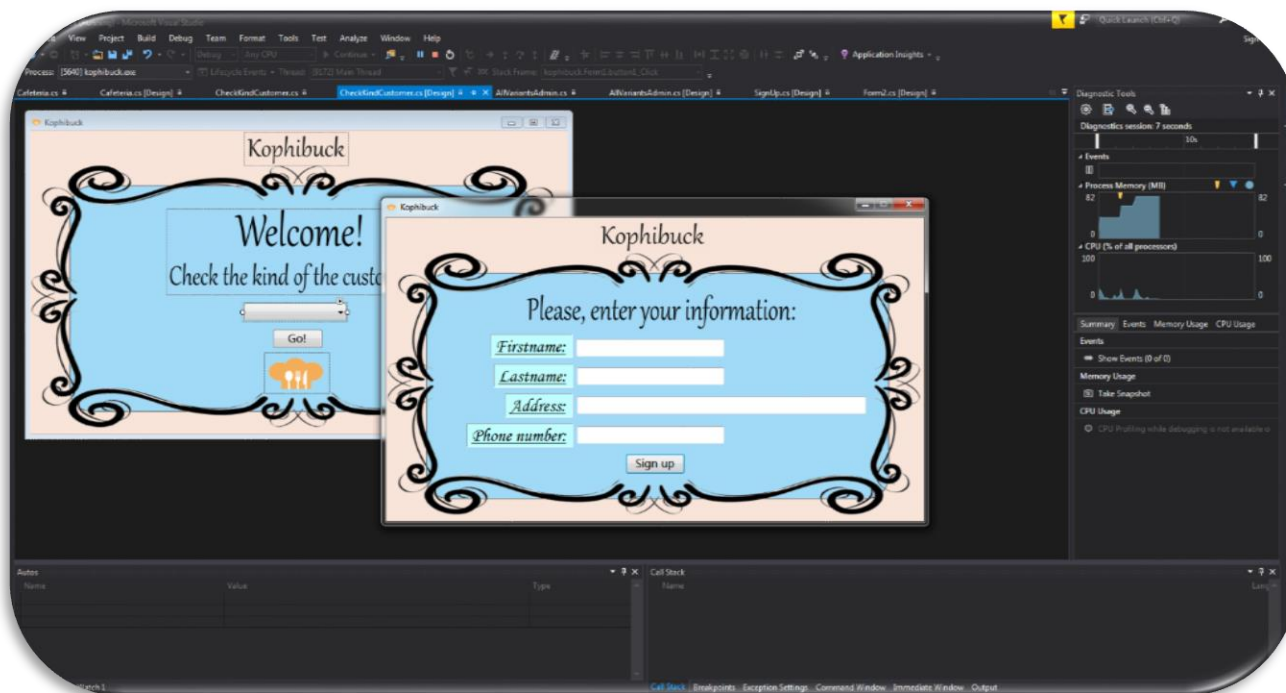
промяна и информацията във формата се актуализира мигновено, за да е актуална с останалите бройки. Ако няма въведени данни, но бъде натиснат бутон, отново излизат подходящи съобщения, искащи input-и.

 The screenshot shows the same web application window. The 'Fill in' button has been clicked, and the form is now populated with input fields. On the left side, there are labels 'Free seats:', 'Fill:', 'Free tables:', and 'Fill:'. On the right side, there are labels 'Booked seats:', 'Release:', 'Booked tables:', and 'Release:'. Each label is followed by a text input field. The 'Free seats:' field contains the number '10'. The 'Free tables:' field contains the number '8'. The 'Booked seats:' field contains the number '10'. The 'Booked tables:' field contains the number '8'. The 'Release:' fields are empty. At the bottom, there are two buttons: 'Fill in' and 'Release'.



Така изглежда login-фотмата за клиентите, като има две опции – да въведат данни, ако преди са влизали в приложението и имат регистрация, или да си направят. Ако се log-нат успешно, се появява съобщение с текста: “Welcome, ...!” и името на вече съществуващия клиент. Не липсват и съобщения за невалидни данни.

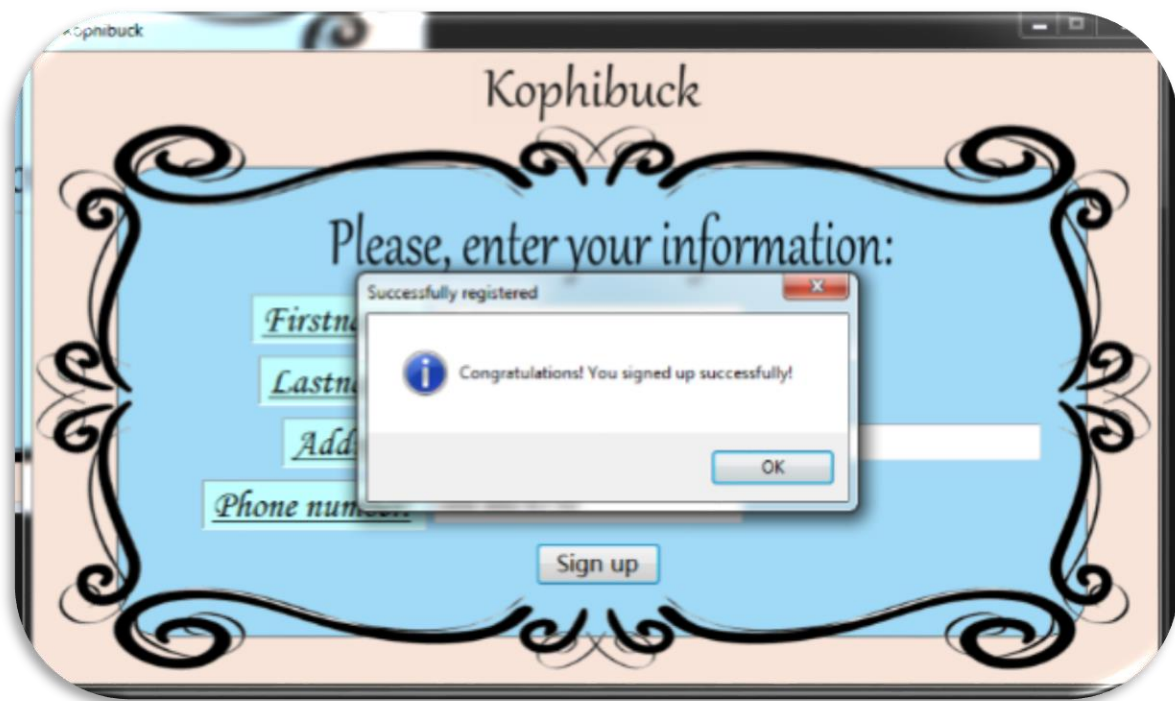




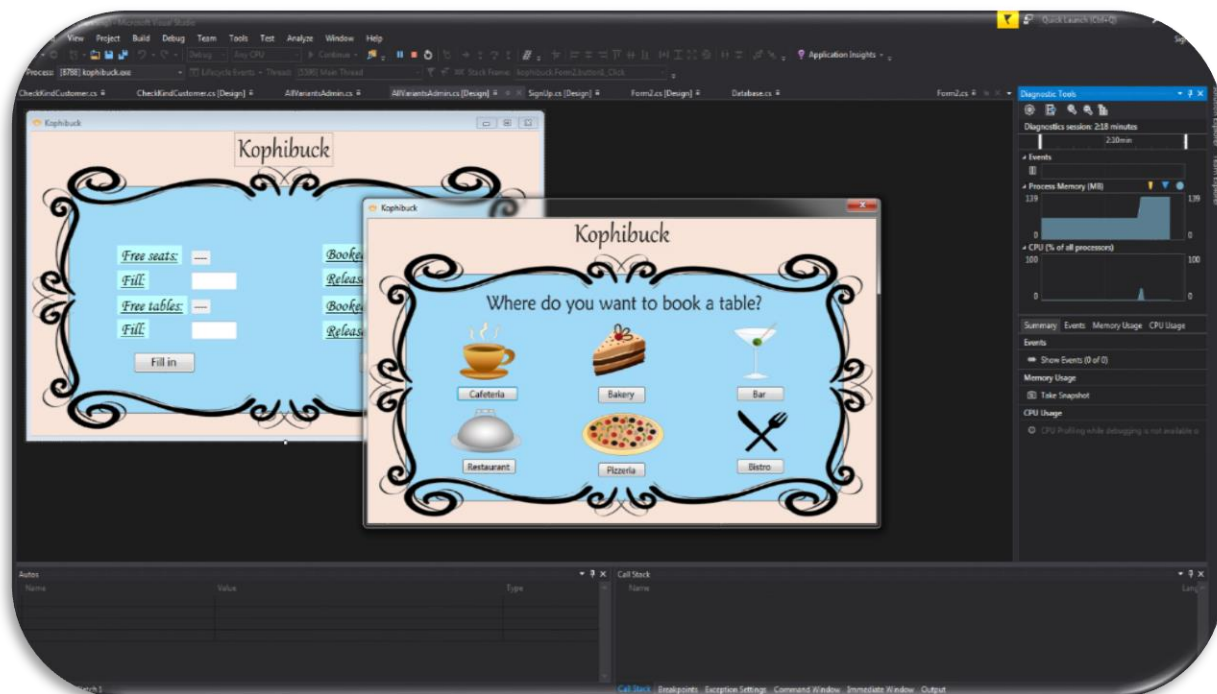
*В случай че потребителят е нов и избере опцията за регистрация, мъ се отворя този прозорец, където той трябва да въведе кратка информация за себе си.*







*Пример за успешна регистрация, а данните се записват в БД, за да може следващия път да се log-не ц тях потребителят.*

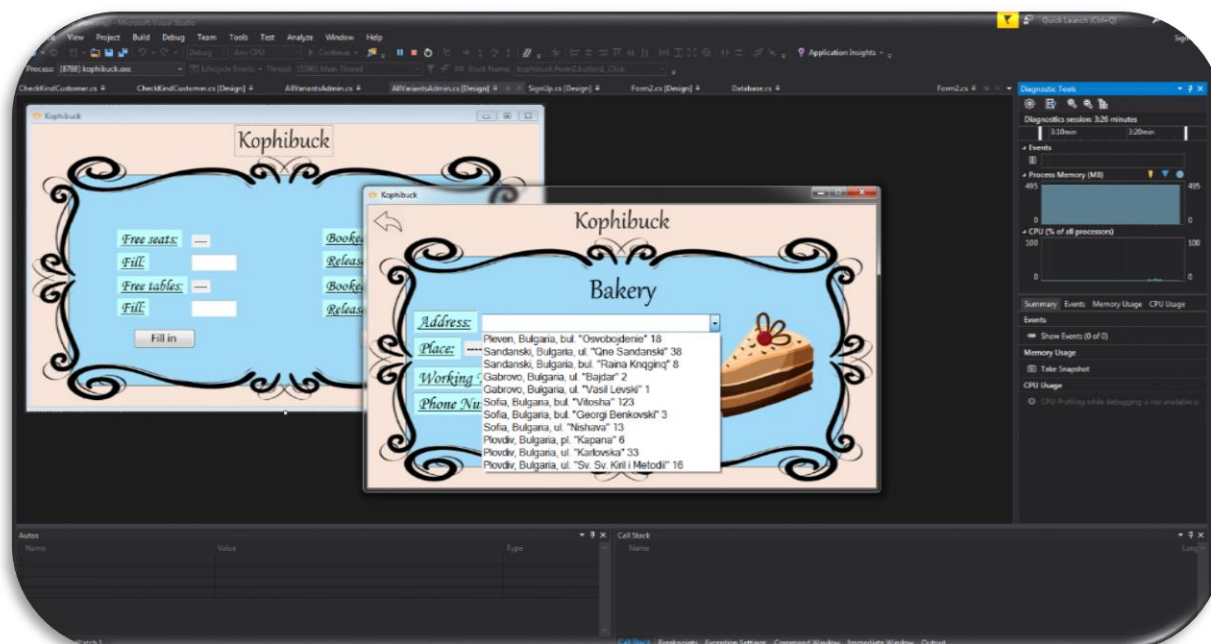


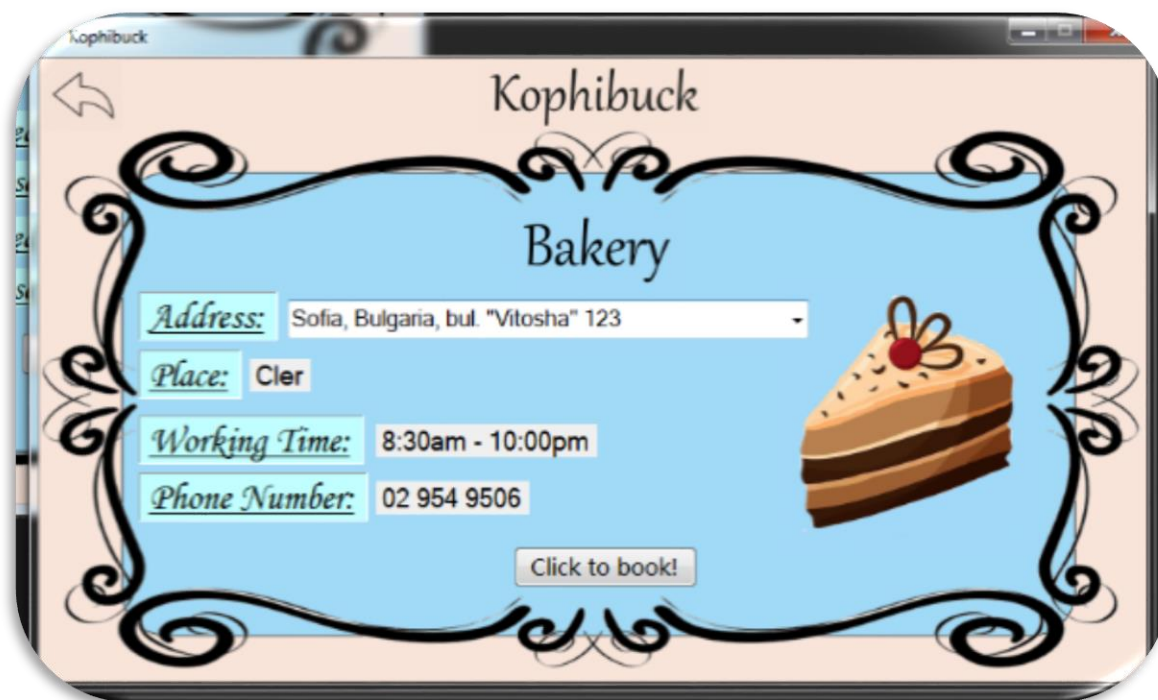
*А ето това е прозорецът, от където клиента избира в какъв вид заведение да направи*

своята резервация. С приятен, говорещ и красив дизайн.



След като бъде натиснат някой бутон от менюто, излиза ето такъв прозорец, като разликата във всяка форма е в текста на label-а отгоре и картинката – всички съответстват на вида заведение за хранене. Клиента избира адреса на пекарната, в случая, и му се извежда информация за мястото от БД.

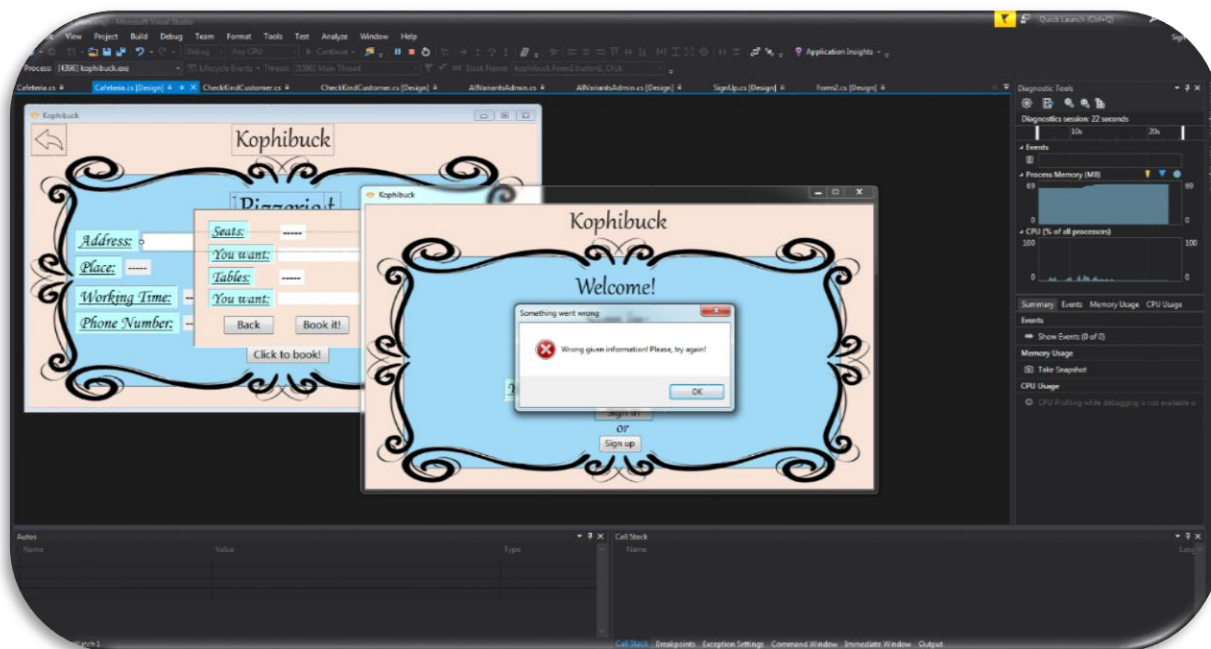




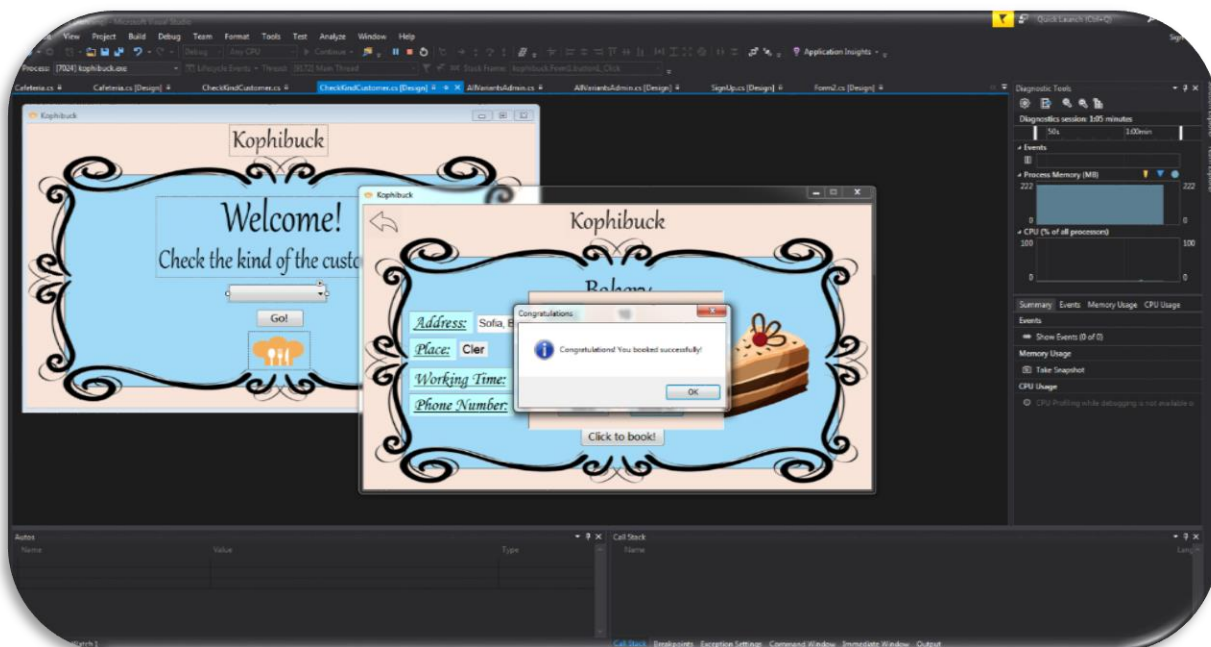
Ако клиента го устройва  
разположението и работното време,  
натиска бутона, за да разбере свододните  
места и да резервира, а ако не – избира ново  
място от падащото меню или нов вид  
заведение от back-бутона.







Съобщение, ако бутонът за резервиране  
буде натиснат, но няма въведени от  
клиента брой места и маси.



Съобщение за коректни данни и успешна  
резервация, при което информацията в  
базата данни, разбира се, се обновява.

## 8. Заключение

Софтуерът „Korhibusk“ представлява платформа за управление свободните и заетите места и маси в места за хранене. Предвидени са повечето необходими функционалности и са добавени нови такива. Софтуерът има възможност да бъде подобряван и променян в бъдеще. Все още е в начален стадий на разработка и предстоят промени. Някои ще бъдат екзистенциални, с цел подобряване на производителността на софтуерния продукт. „Korhibusk“ може да бъде използван от всички, искаше събрана на едно място информация. Идеята е да получим една по-добра система, един по-добър конкурентноспособен софтуер в сферата на резервациите.



## **9.Използвана** **литература**

### **Книги:**

- ✓ Колисниченко, Д. (2016). *.Net framework*;
- ✓ Светлин Наков, Св. (2018). *Принципи на програмирането със C#*;
- ✓ “Алекс Софт”, (2013). *Adobe Photoshop CS6. Официален курс на Adobe Systems*;

### **Линкове:**

- *Photoshop*

<https://helpx.adobe.com/photoshop/tutorials.html>

- *C#*

<https://www.learncs.org/>