**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

**ДИПЛОМНА РАБОТА**

Тема: Търговски софтуер за създаване и обработка на поръчки

Дипломант: Научен ръководител:

*Иван Драганов Юлиян Линев*

СОФИЯ

2019

**Увод**

Връзката ни с дигиталните технологии се вплита все по-дълбоко в живота ни от гласовите помощници на Гугъл и Майкриософт до разпознаването на снимки във Фейсбук и VK.com. В забързаното ни съвремие потребителите очакват всичко да се случва с един клик и минимално усилие.

Въпреки това все още използваме брушури, периодични издания и рекламни материали на физически носители, остаряли идеи, парадоксално запазили се в едни от най-динамичните отрасли – търговия и логистика. Негативите са сериозни както за бизнеса, така и за околната среда - тромава структура за вмъкване на нови продукти, както и екологичното замърсяване, нуждата от голям брой хора за подръжка на услугите.

От друга страна много хора не се престрашават да встъпят в света на дигиталните технологии, несигурност която лесно ще бъде преодоляна с ергономичен дизайн и бързодействие на софтуера. Такъв тип програма ще изиграе ролята на междинно стъпало между онлайн търговията и физическата такава – често искаме да поръчаме или запазим артикул през интернет, но да го закупим по „класическия“ начин – посещавайки магазина и преглеждайки артикула на място, всичко това в удобно за клиента място. Такъв „хибриден“ подход помага и за разрешаване на проблеми, както свързани със самия продукт, така и в случаяй, че се получават чрез куриер. Не на последно място, ползването на програма ще позволи проследяване на продукта и доставки, в случай, че се избере използването на такива.

Проектът е алтернативен вариант за интернет пазаруване вдъхновен, но не ограничаващ се с иновативните идеи за онлайн търговия на Amazon, чрез “Alexa” и плеядата от устройства работещи с него. Тъй като Югоизточна Европа и България в частност не са развит пазар, системите на Amazon, Google и Microsoft, не са подходящи, тъй като обвързват крайния клиент със серия продукти, които или не се предлагат на нашия пазар или са комбинирани със скъпи устройства. Нашия продукт предлага гъвкавостта на проложение за смарт устройство (Android), които така или иначе са широко разпространени и използвани и изключват, първоначалната инвестиция за физическо устройство.

Клиентите имат възможност с едно натискане на бутон да сканират предоставен QR код и да поръчват или запазят в обект дадените продукти. За търговците и производителите това е предпоставка за свобода при създаването и разпространението на QR кодове, било то онлайн или на хартиен носител. Този по-нестандартен начин на пазаруване го прави по-интересно за потребителя,което спомага за по-добра връзка клиент – магазин.

# ПЪРВА ГЛАВА

# Обзор за съществуващите решения

## **1.1 Подобни съществуващи решения**

Много от магазините и търговските обекти в днешно време имат своя онлайн платформа, предлагаща различни услуги. В основата си всички се ръководят на един и същи принцип. Клиентите въвеждат необходима информация и селектират желаните от тях артикули,след което изпращат поръчка към търговският обект.Подобни търговски софтуери са Microinvest Склад Pro и Mistral.

Microinvest Склад Pro е гъвкава система за управление на стоковите наличности и парични потоци на фирмите. Програмата позволява да бъдат контролирани всички процеси, свързани с управление на производствената и търговската дейност. С нея можете да оцените постигнатото от Вашия бизнес чрез подробни справки и

статистически отчети. Едновременно с това с програмата се извършва пълен мониторинг върху дейността на персонала чрез регистрирането на всяко действие и операция, извършени от него.Системата позволява да се проследи във всеки един момент количеството използвани запаси и да се планират необходимите разходи. Microinvest Склад Pro познава нуждите на Вашата фирма и дава съвети за бъдещи покупки и поръчки.

От друга страна няколко от софтуерните решения на Mistral предлагат мобилни справки чрез смартфон(на устройството се визуализира информация за стоковата наличност,оборот,зети маси и доставки и др. ). Също така кухненски дисплей – поръчките се появяват мигновено на дисплей в кухнята,след подаването им от сервитьор или от посещаващите онлайн платформата за поръчки.Готвачът има възможност интерактивно да променя рецептата при липса на продукт и наличие на заместващ.

Има много подобни и интересни софтуерни решения на пазара, но това с което се различава представеното, е неговият начин на работа. Потребителят се нуждае единствено от смарт телефон,който вече е нещо,което всеки притежава. Чрез сканиране на QR код с камерата на устройството,клиентът добавя продукта в количката си и получава информация за него,а благодарение на иновативния си интерфейс с един клик той може да изпрати поръчка към конкретния магазин.От друга страна софтуерната част на продукта,предназначена към търговеца,предоставя лесен начин за създаване на QR кодове,добавяне на артикули към база данни и получаването и обработването на поръчки – всичко това на един екран!

## **1.2 Развойни средства и среди. Проучване.**

Чрез проучване и разговори с хора,свързани пряко с търговията и търговски магазини,достигнах до идеята за направата на интересна и нестандартна система за поръчки. Със съвместна работа с хора,разработващи търговски софтуери на професионално ниво получих необходимите знания за да започна работа,а именно,какви параметри да съдържа един продукт,поръчка,клиент,правилен начин на функциониране на програмата.

Този софтуер може да се развие много.Има добра основа,в която могат да се вграждат нови и нови функционалности свързани с търговията,както за търговеца,така и за клиента.

Системата се състои от три софтуерни части.

Първата и най-важната част,това е уеб услуга,в която са заложени всички взаимодействия с базата,под формата на HTTP заявки. Всяко действие от другите две софтуерни части,например добавяне на нов артикул,потребител,изтриване на поръчка или нещо друго,изпраща заявка към уеб услугата,чрез която се извършва конкретна работа,свързана с базата данни.

Втората част е настолно приложение,което е предназначено за търговеца или търговският обект.Там служител въвежда информация за даден продукт,след което с натискане на бутон се генерира QR код съдържащ определена информация.Едновременно с това се изпраща и заявка към уеб услугата , която да добави даденият продукт към базата данни.В друг прозорец в същото време служителят може да следи новите поръчки от клиенти и да ги приема.

Последната част представлява андройд приложение,предназначено за клиенти на даденият търговец или търговски обект. Чрез едно натискане на бутон,камерата на устройството се превръща в скенер.След сканиране на артикул той автоматично се добавя в количката,където могат да се коригира количеството му. В същото време на екранът се появява повече информация за даденият продукт.Клиентът може да изпрати по всяко време поръчка с наличните в количката продукти и да следи нейният статус във всеки един момент.

Комуникацията на андроид и десктоп приложението с уеб услугата е независима и става посредством интернет връзка в мрежа.

Базата данни е SQL, изградена посредством code-first метод, наличен в “Entity Framework Core”.

# ВТОРА ГЛАВА

# ПРОЕКТИРАНЕ НА СТРУКТУРАТА

## **2.1. Функционални изисквания**

Системита за клиент и тази за търговец или търговски обект са почти независими една от друга и комуникират с уеб услугата,която от своя страна извършва определени действия с базата.

Андроид приложението дава възможност на потребителите да влезнат в свой профил или да се регистрират ,ако нямат такъв.След успешно влизане клиентът може да влезне в режим сканиране и да сканира съответните QR кодове,което автоматично ще добави сканирания продукт в кошницата.В нея може да се коригира количеството на даденият продукт,а над нея се изписва информация за продукта.По всяко едно време може да се приключи пазаруването и да се изпрати поръчка към даденият магазин. Също така приложението съдържа настройки,като смяна на езика и размерът на буквите, но също така препратки към други екрани с различни функционалности.

Андроид приложението се дели на пет основни части:

* Панел за влизане с име и парола. След попълване на полетата, информацията се изпраща към уеб услугата ,която проверява за такъв потребител и връща нужната информация обратно. На този панел се намира и бутон ,който ни отвежда при интерактивна форма за регистрация. Бутон за настройки,намиращ се в хедъра.
* Панел с форма за регистрация.След попълване на всички полета, информацията се изпраща към уеб услугата,която от своя страна регистрира новият потребител.
* Основен екран. На него се намира бутонът за сканиране,количката,бутонът за изпращане на поръчка и полето за информация. При натискане на бутонът за сканиране устройството влиза в режим на сканиране. След успешно сканиране,даденият продукт автоматично се добавя в количката. Там може да се увеличава или намалява бройката му , а също така и да се премахне при необходимост. Бутонът за изпращане на поръчката изпраща заявка към уеб услугата с всички продукти в количката и се създава поръчка за даденият потребител.Бутон за настройки,намиращ се в хедъра.
* Екран за следене на състоянието на поръчките. На него са визуализирани всички сегашни поръчки на потребителя.
* Екран за редакция на лични данни на потребителя. При необходимост за смяна на адрес или други лични данни, пряко свързани с доставките.

Десктоп приложението е предназначено за търговеца или веригата магазини. То дава лесна възможност за добавяне на продукти в база данни,също така и генерирането на QR код за тях. Другата му функционалност е следеното и приемането на поръчки от клиенти.

Десктоп приложението се дели на четири основни части:

* Екран за влизане в системата. . След попълване на полетата, информацията се изпраща към уеб услугата ,която проверява за такъв потребител и връща нужната информация обратно.
* Екран за генериране на кодове. След въвеждане на информация във всички нужни полета и натискане на бутон,на екранът се визуализира QR кодът.Това действие автоматично изпраща заявка към уеб услугата и записва даденият продукт в системата.
* Екран за поръчки. На него се визуализират всички поръчки.Те могат да бъдат приети или отхвърлени.
* Екран за общи настройки.Там потребителят може да зададе желаните от него настройки,с който да работи приложението.

От своя страна уеб услугата просто държи базата данни и служи за връзка между двете приложения и базата. В него са HTTP заявките,който могат да се ползват и от двете приложения при необходимост.Тези заявки от своя страна извършват необходимите манипулации с базата и връщат нужната информация. Като параметри приемат ID на потрбителят,който ги е изпратил и другата нужна инфромация,необходима за нейната работа.

## **2.2. Съображения за избор на програмни средства и развойната среда**

Трите части на системата са ясно разделени и от страна на езиците за програмиране.

Тъй като Android приложението трябва да е лесно за използване от всеки потребител на различна възраст е направено много просто от към front-end. Поради тази причина избрах да използвам един много функционален език за създаване на Android приложения, а именно Basic4Android. Езикът е алтернатива на програмирането с Java и Android SDK. Платформата компилира до Java и след това отново го връща като Basic, използва смесен (Cross) компилатор.

Резултатът е изпълним код, предназначен за платформа различна от тази, на която е извършена компилацията. Използва се например при създаване на код за [вградени системи](https://bg.wikipedia.org/wiki/%D0%92%D0%B3%D1%80%D0%B0%D0%B4%D0%B5%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0" \o "Вградена система) (embedded systems), както и в Basic4Android, където няма възможност, или е по-трудно, да се създават програми.

B4A включва визуален дизайнер, който опростява процеса на изграждане на потребителски интерфейси, насочени към телефони и таблети с различни размери на екрана. Компилираните програми могат да бъдат тествани в емулатори на AVD Manager или на истински Android устройства, използващи Android Debug Bridge и B4A Bridge.  
Самият език е подобен на Visual Basic и Visual Basic .NET, но е адаптиран към родната Android среда. Има разлики и прилики с Java, най-големия плюс е, че се пише много бързо и е много функционално, не е гъвкаво колко Java, но по отношение на приложението, което е нужно за системата и ще се използва, е нужно просто бързодействие и лесно разгадаване на програмата.

За да може да работи приложението освен компилатора, който се предоставя с B4A платформата е нужно да се свърже към нея Java JDK, за да може да се установи компилация, Android SDK за да се инсталират платформите и инструментите за версиите на Andoid-а, който ще се използва. Също така най-новата версия на B4A е платена и нужен лиценз.

По желание може да се използва B4A-Bridge, което компилира на устройството безконтактно чрез Wireless връзка, ако двете устройства са в една и съща мрежа. Може да се изтегли от Google Play безплатно.

Другия малко по-гъвкав, но не практичен вариант, поради усложняване на работата за създаването на Android проложението е писането на Java на Android Studio. Но използването на Android Studio ще създаде повече работа по програмирането на това, не толкова сложно приложение.

От друга страна уеб услугата и десктоп приложението са написани на C#,Visual Studio(IDE).

**Microsoft Visual Studio** е мощна [интегрирана среда за разработка](https://bg.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%B0%D0%BD%D0%B0_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D0%B7%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0" \o "Интегрирана среда за разработка) (на [английски](https://bg.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8_%D0%B5%D0%B7%D0%B8%D0%BA): *integrated development environment*, *IDE*) на софтуерни приложения за [Windows](https://bg.wikipedia.org/wiki/Windows) и за платформата [.NET Framework](https://bg.wikipedia.org/wiki/.NET_Framework" \o ".NET Framework). Използва се за разработка на конзолни и [графични потребителски интерфейс](https://bg.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D0%BD_%D0%BF%D0%BE%D1%82%D1%80%D0%B5%D0%B1%D0%B8%D1%82%D0%B5%D0%BB%D1%81%D0%BA%D0%B8_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81" \o "Графичен потребителски интерфейс) приложения, както и Windows Forms или WPF приложения, [уеб сайтове](https://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1%D1%81%D0%B0%D0%B9%D1%82" \o "Уебсайт), [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) на всички поддържани платформи от Microsoft Windows, [Windows Mobile](https://bg.wikipedia.org/wiki/Windows_Mobile" \o "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. Могат да се добавят и плъгини, които повишават функционалността на почти всяко ниво – включително добавянето на поддръжка за [source-control системи](https://bg.wikipedia.org/w/index.php?title=Source-control_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B8&action=edit&redlink=1) (като [Subversion](https://bg.wikipedia.org/wiki/Subversion_(%D1%81%D0%BE%D1%84%D1%82%D1%83%D0%B5%D1%80)) и Visual SourceSafe), добавяне на нови инструменти като редактори и визуални дизайнери за [domain-specific languages](https://bg.wikipedia.org/w/index.php?title=Domain-specific_languages&action=edit&redlink=1) или инструменти за други аспекти (като например: [Team Foundation Server](https://bg.wikipedia.org/wiki/TFS), Team Explorer).

Visual Studio поддържа различни [езици за програмиране](https://bg.wikipedia.org/wiki/%D0%95%D0%B7%D0%B8%D1%86%D0%B8_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5) (например: [C#](https://bg.wikipedia.org/wiki/C_Sharp), VB.NET, [C](https://bg.wikipedia.org/wiki/C_(%D0%B5%D0%B7%D0%B8%D0%BA_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5))/[C++](https://bg.wikipedia.org/wiki/C%2B%2B), [F#](https://bg.wikipedia.org/w/index.php?title=F_Sharp&action=edit&redlink=1), [XML](https://bg.wikipedia.org/wiki/XML)/[XSLT](https://bg.wikipedia.org/wiki/XSLT), [HTML](https://bg.wikipedia.org/wiki/HTML)/[XHTML](https://bg.wikipedia.org/wiki/XHTML), [JavaScript](https://bg.wikipedia.org/wiki/JavaScript), [CSS](https://bg.wikipedia.org/wiki/CSS) и други) и различни технологии за разработка на софтуер (Win32, COM, [ASP.NET](https://bg.wikipedia.org/wiki/ASP.NET), ADO.NET Entity Framework, Windows Forms, WPF, Silverlight и още десетки други Windows и .NET технологии). Съществуват и отделни езикови версии на Visual Studio, които обаче предоставят по-ограничени услуги за потребителя: [Microsoft Visual Basic](https://bg.wikipedia.org/wiki/Microsoft_Visual_Basic), Visual J#, Visual C#, and Visual C++.

## **2.3. Принцип на софтуерен модел**

След стартиране на Android приложението се отваря екран за влизане в програмата, за идентификация на потребителя. След въвеждане на име и парола се изпраща заявка към уеб услугата за валидация на данните. След идентифициране се преминава в следващия екран на приложението , където потребителят има възможността да сканира кодове,работи с количката си и да управлява неговите поръчки

Всяка операция изпраща заявка към уеб частта и очаква даден отговор, затова още при стартиране на приложението се проверява връзката на устройството с интернет.

'Ping Google DNS - if you can't reach this you dont have internet on the device!

Sub **Connected** As Boolean

Dim p As Phone

Dim Response, Error As StringBuilder

Response.Initialize

Error.Initialize

p.Shell("ping -c 1 8.8.8.8",Null,Response,Error)

Log("======= Response ========")

Log(Response)

Log("======= Error ===========")

Log(Error)

Log("======================")

If Error.ToString="" Then

Return True

Else

Return False

End If

End Sub

Tози метод се ползва като параметър при инициализацията на HTTP модула.

'Initializes the object. You can add parameters to this method if needed.

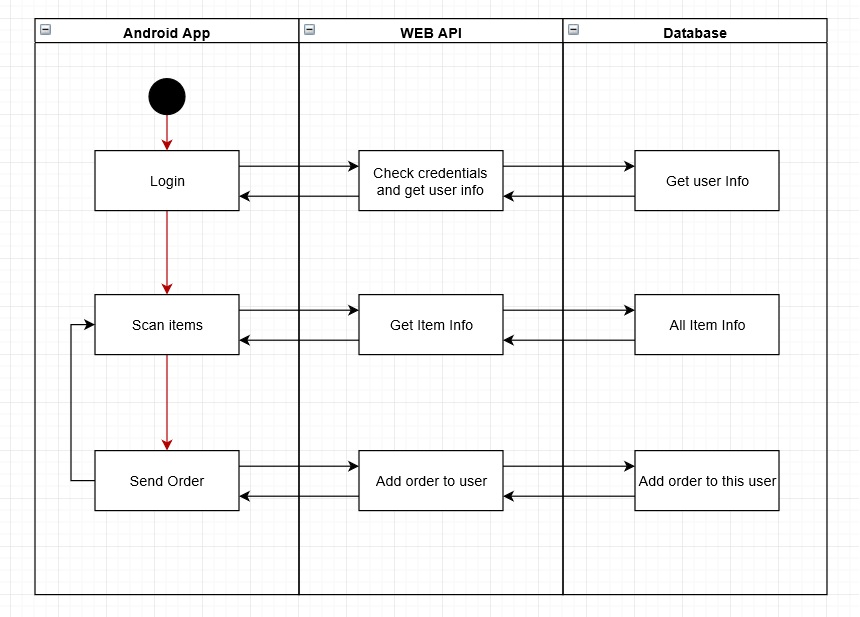
Public Sub **Initialize**(Connection As Boolean)

IsConnected = Connection

End Sub

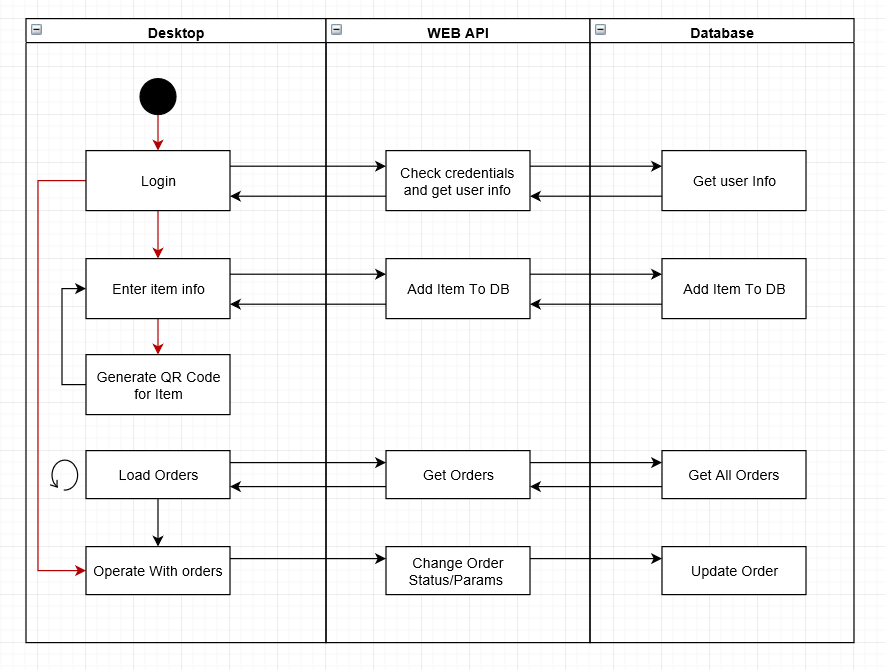
Така при неуспешна връзка с интернет не може да се ползва нито една заявка и потребителят не може да използва приложението.

След като влезе, потребителят може да сканира желаните от него продукти, за всеки продукт се взима нужната информация от базата, а при изпращане на поръчка,тя се финализира и изпраща отново към базата.

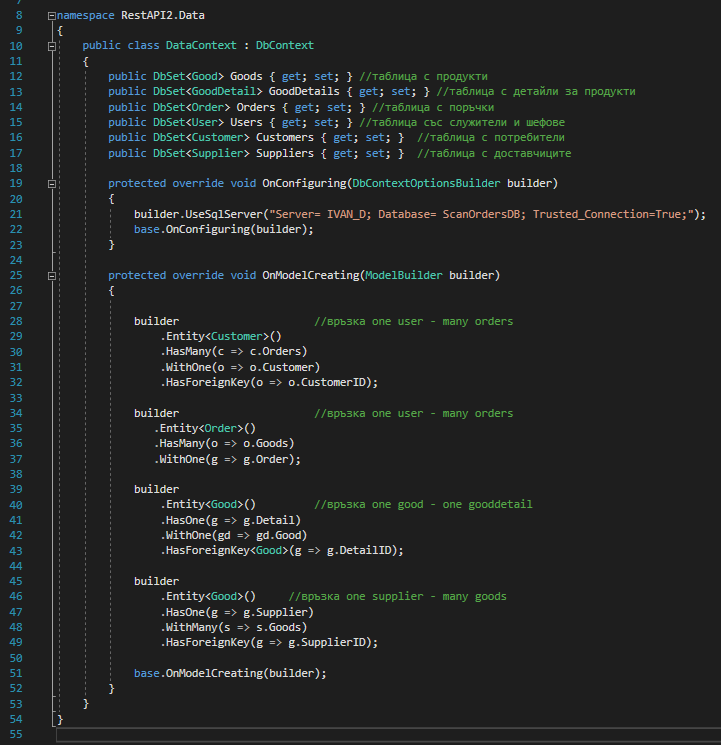


Настолното приложение работи на същият принцип, с разликата ,че то е предназначено за персонал и има различна функционалност. Създава и добавя продукти в базата, като в същото време им създава QR код. Също така получава и обработва поръчки , дошли от клиенти чрез мобилното приложение.

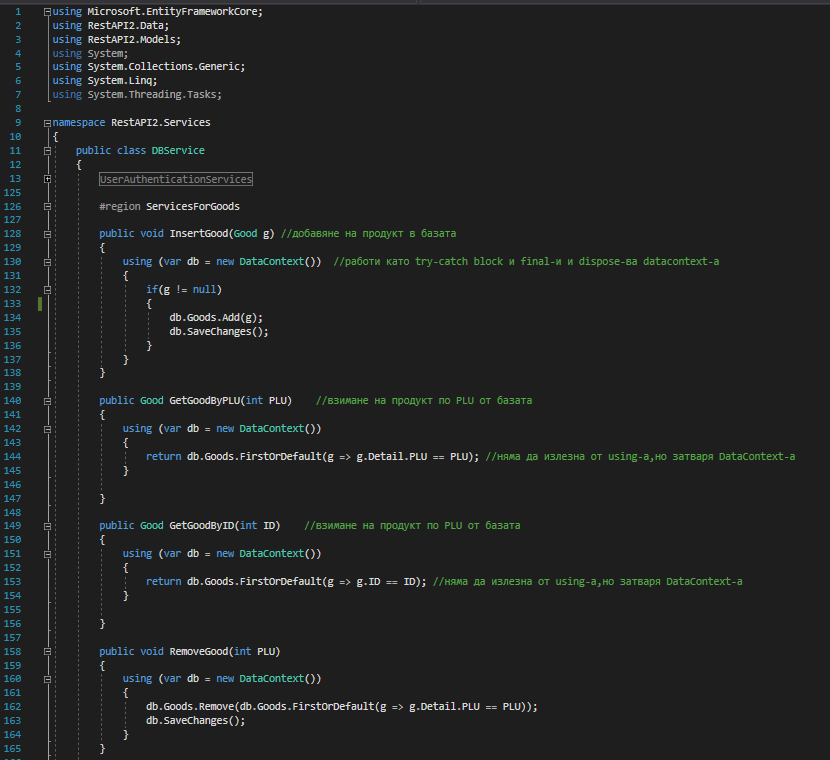
Потребителите за настолното приложение и това за андроид са различни и не могат да се вписват в чуждите платформи, поради различното предназначение на двете програми.



Уеб услугата е предназначена за комуникация и различни операции с базата. Двете приложения комуникират чрез заявки с нея, а тя изпълнява нужните операции с базата данни. Самата база е писана с подход Code First , на Entity Framework Core. Това ще рече,че моделите на приложението са създадени първи и спрямо тях Visual Studio е направило базата и необходимите връзки. Обработката на заявките се извършва в класа Controller, a връзките с базата са описани в класа DataContext.



Всички операции свързани с базата са записани под формата на методи в класа DBServices. След всяка манипулация на информацията записана в базата, се извиква методът SaveChanges() , чрез който се запазва сегашното състояние на базата.



## **2.6. Преглед на библютеки използвани за приложенията**

* Андроид

JSON

* Taзи библиотека помага с сериализацията и десериализацията на JSON обекти и създаването на JSON string.

B4aZXing

* Taзи библиотека трансформира камерата на устройството в скенер и предоставя съответните методи за работа със сканирани данни.

ОkHttpUtils2

* Тази библиотека създава и изпраща необходимите HTTP заявки

Runtime Permissions

* Ако версията на андоид SDK е по-малка от 23, тогава стандартната система за позволения ще бъде използвана на всички устройства включително Android 6+, но скоро всички Google Play апликации ще трябва да използват версия на андоид SDK 26+. Чрез runtime permissions потребителят не е питан за позволения когато инсталира приложение от Google Play.Вместо това искането за позволение е по време на изпълнение.

Camera

* Дава достъп до камерата на дадено усройство и съдържа проверки за намирането на налична такава.

Reflection

* Тази библиотека съдържа обект Reflection, който позволява достъп до методи и полета от обектите на Android, които не са изложени на езика Basic4Android. Това прави това чрез съоръжение, наречено "Reflection", което използва метаданни за обекти, които са включени в пакета за приложения и позволява динамичен достъп до полета и методите по време на изпълнение.
* Настолно приложение и Уеб услуга

Мicrosoft EntityFrameworkCore

* Entity Framework (EF) Core e лека cross-platform версия на популярната Entity Framework data access technology. EF Core може да служи като object-relational mapper (O/RM), позволяващо на .NET разработчиците да работят с база данни използвайки .NET обекти и елиминирайки нуждата на по-голямата част the data-access кодът , който попринцип те трябва да напишат.

Microsoft AspNetCore Mvc

* ASP.NET Core MVC е framework за разработката на web приложения и APIs използвайки Model-View-Controller модел на разработка. Този модел разделя приложението на три главни групи от компоненти : Models, Views, и Controllers.По този начин,потребителските заявки са насочени към Controller, който е отговорен за работата с Model ,за да изпълни потребителски действия или да всеме резултат от queries.

## **2.7. Получени резултати**

Полученият резултат е едно интерактивно и нестандартно за пазара приложение, което улеснява търговията. Системата дава свобода на рекламна изява от страна на продавача, тъй като той може да разпространява където поиска свойте QR кодове – брошури,списания,билборди и други реклами. Потребителя може по всяко едно време да сканира и поръча конкретните продукти.

a. Мнението на хората

Приложението е все още в тестов период и не е достъпно публично.

b. Плюсове

Интересен и различен начин на работа

Улеснен интерфейс

Улеснява поръчването на продукти

Оползотворява използването на брушури и хартиени рекламни материали

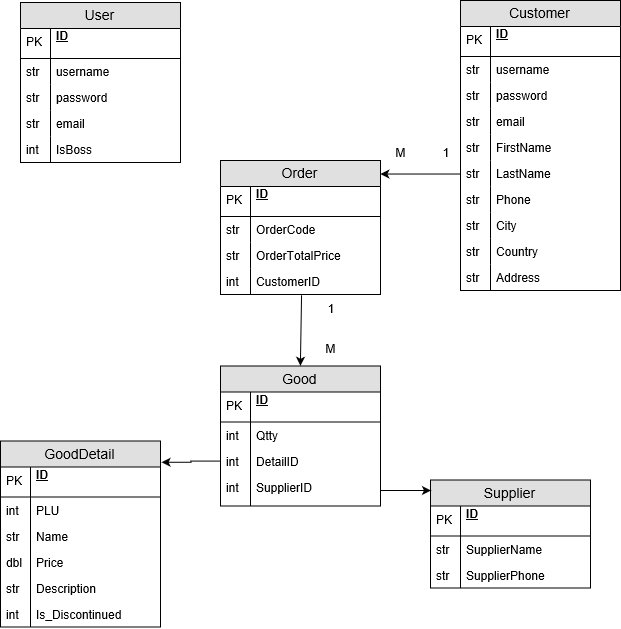
c. Минуси

За сега минуси не са установени, тъй като не е пуснато в употреба в реална среда.

## **2.8. Проектиране на структурата на базата от данни**

Базата данни се състои от няколко таблици – User,Customer,Order,Good,GoodDetail,Supplier.

Tаблицата User съхранява информацията за потербителите на конкретния магазин.Тя се използва за тези потребители,който се вписват в настолното приложение. В таблицата Customers се съхраняват регистрациите на потребителите на андроид приложението. Всеки Customer има много Order(Поръчки). Връзката е one-to-many.Всяка поръчка съдържа информация за сеге си и списък от продукти(Good).Tук връзката е пак one-to-many. Eдна поръчка може да има много продукти. От своя страна всеки Good има по един GoodDetail и един Supplier. В GoodDetail се съхранява информацията за даден продукт, а в Supplier – информацията за доставчика. Всеки Supplier обаче има много Goods,които той доставя.



Таблицата User е независима и не е обвързана с останалите таблици.

# ТРЕТА ГЛАВА

# ПРОГРАМНА РЕАЛИЗАЦИЯ

Реализацията на продукта започва с направата на Уеб услугата, която е най-основната част от системата. Приложението се дели на четири основни части:

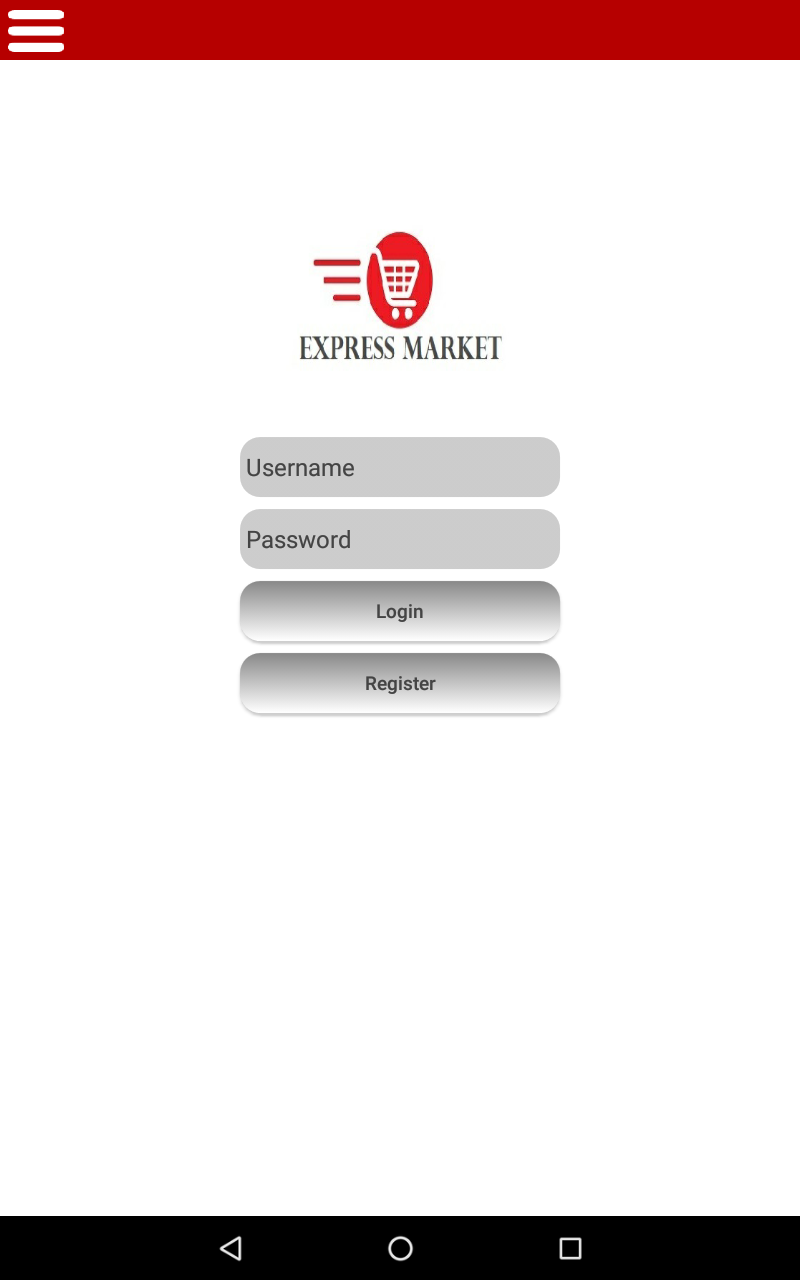
* Модели на обектите,който ще съхраняваме в базата данни.
* DataContext клас в който създаваме таблиците, връзките между тях и връзката със сървър.
* DBServices клас в който са описани различните методи свързани с работа с базата.
* Controller клас който съдържа всички HTTP заявки, нужни за работа с приложенията.

При стартиране на Уеб услугата се създават таблиците на базата данни,описани в DataContext класа, и връзките между тях.

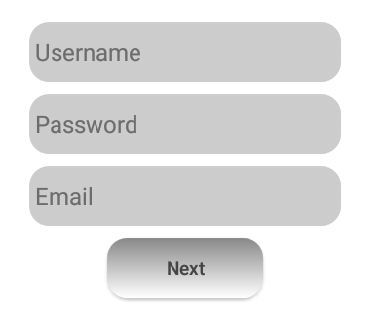
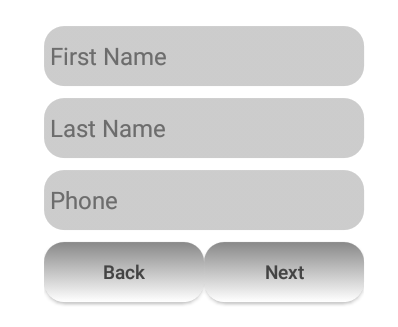
Системата приема HTTP заявки от останалите приложения, които се обработват в Controller класа.Te може да са POST,GET,PUT,DELETE. В зависимост от подадената заявка се извикват методи от DBServices класа,който са отговорни за директна работа с базата данни. Всичко свързано със SELECT,INSERT,UPDATE и тн. е в този клас.

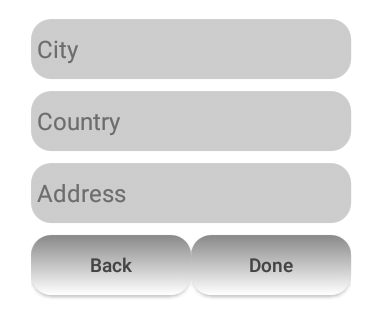
От своя страна Android приложението се дели на пет основни части:

* Панел, върху който се построява екран за влизане в системата с име и парола, след попълване на полетата, информацията се изпраща чрез заявка към уеб частта за валидация.

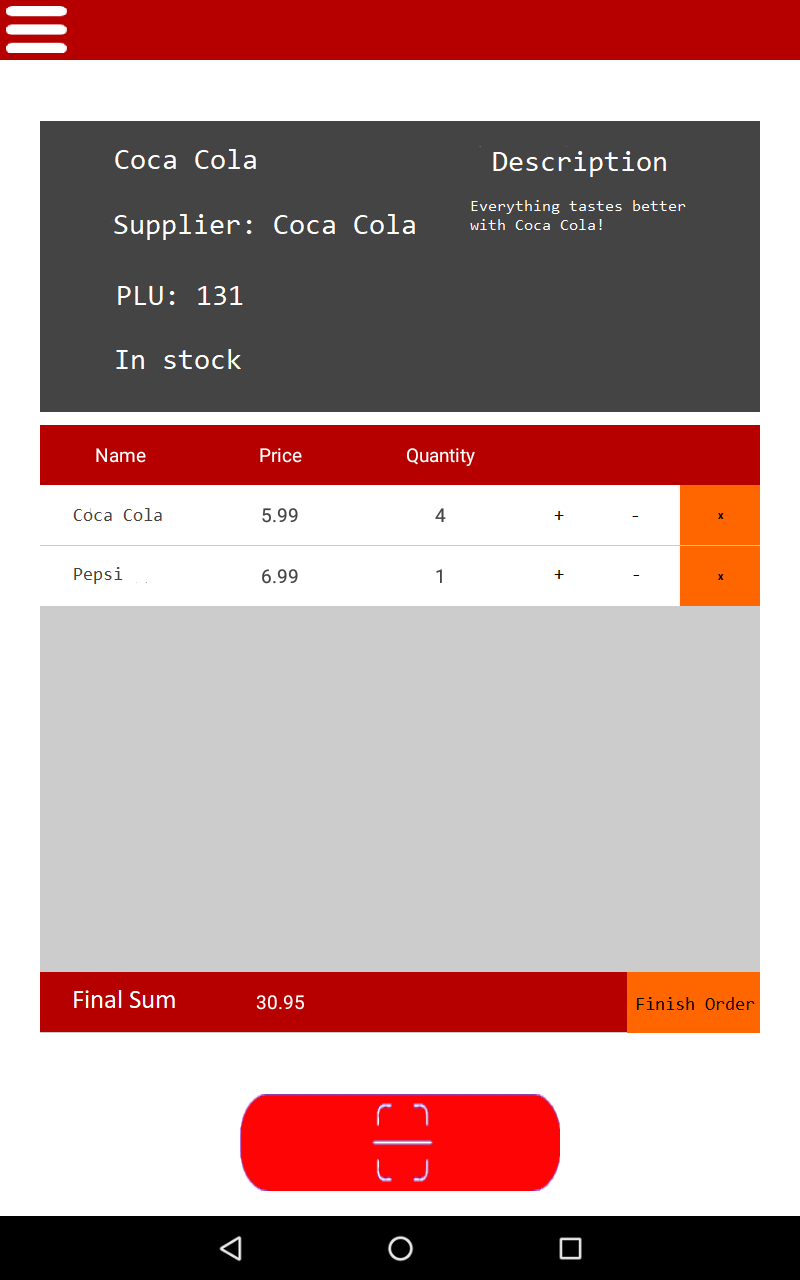


* Екран за регистрация.При натискане на бутонът за регистрация се построява друг екран. На него потребителят следва необходимите стъпки и се регистрира.След натискане на бутонът “Done” се изпраща заявка,съдържаща необходимата информация за регистрация.

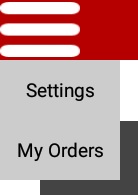


* Главен екран на приложението. Визуализира се след успешен вход в системата. На него се намират бутонът за сканиране,количката с продукти и бутон за настройки.

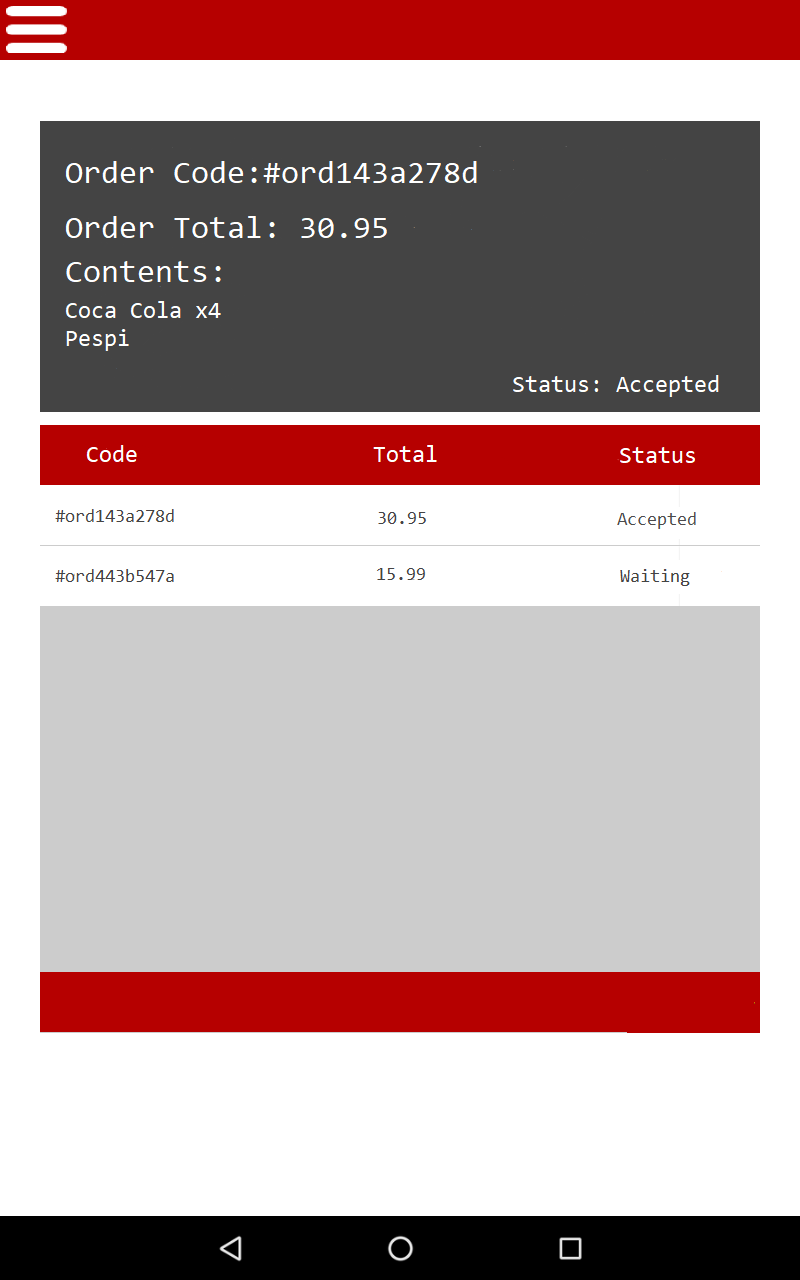


При селектиране на продукт от количката, допълнителна информация за него се изписва на екрана. Всеки продукт в количката има бутони за манипулация на количеството му и бутон за премахване. На количката е бутонът за приключване на поръчки,който при натискане отвежда потребителят на друг екран,където да финалицира своята поръчка.

Бутонът за настройките е един и същ в екрана за вход и в главния екран.При натискане се показват менюта,който ни отвеждат в други екрани. Единствената разлика е визуализацията на менюто ,което ни отвежда към сегашните поръчки на потребителя.Това меню не е налично в началният екран на приложението по обясними причини.



* Екранът в който се визуализират поръчките на потребителят прилича на главният екран. В него се построява Scroll View , което съдържа поръчките на клиента. При селектиране на дадена поръчка на екрана се показва повече информация за нея.



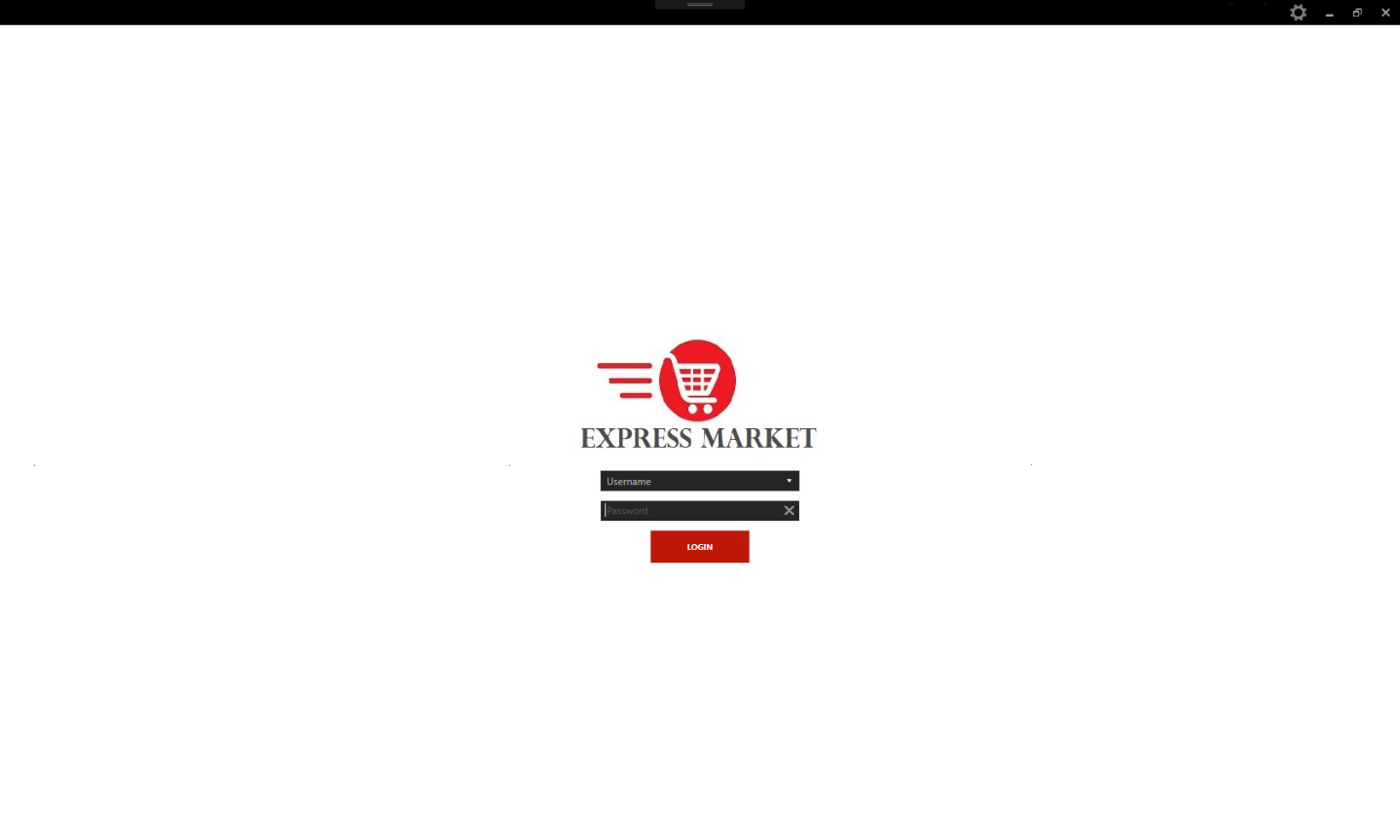
* Екранът за общи настройки на приложението. Достъпен е от във всеки един момент. В него потребителят има избор на език и размер на текст в приложението. При вече влезнал потребител се визуализират настройки за смяна на адреса и местонахождението на потребителят, при необходимост.



След натискане на бутона за връщане назад на устройството или след смяната на екран от бутонът за настройки,въведените данни се запазват.Тези който са за приложението се запазват локално,а тези свързани с локацията се изпращат със заявка.

Настолното приложение е съставено от четири части,подобни на мобилният софтуер,но с различно практическо предназначение.

* Началният екран се състои от форма за въвеждане на име и парола,както и бутон за общи настройки на приложението. Не съдържа бутон за регистрация,тъй като това е служебен софтуер и се раздават готови потребители на служителите. След въвеждане на правилна информация се изпраща заявка към уеб услугата.



След успешно влизане служителите попадът в основния прозорец на програмата,където имат възможност да сменят между два изгледа. На единият могат да видят всички налични поръчки и да ги обработят, а на другият да въведат необходима информация за генерирането на QR код.



* В изгледа за поръчки новите такива се визуализират от лявата страна и с тях е представена основна информация за клиента. При селектиране на поръчка в центъра на екрана се визуализира нейното съдържание.