

ПРОФЕСИОНАЛНА ГИМНАЗИЯ ПО ЕЛЕКТРОТЕХНИКА – ВАРНА



Дипломен проект

за придобиване на професионална квалификация

Тема: ПОС Терминал за хранителен магазин

Специалност: Приложен програмист

Професия: Приложно програмиране

Дипломант: Петър Стефанов Стефанов

Ръководител: инж. Андрей Тодоров

ПГЕ Варна, 2022-2023г.

СЪДЪРЖАНИЕ

Увод	3
ГЛАВА 1: Цели и задача на дипломната работа	4
ГЛАВА 2: Изследователски въпроси и теоретична рамка	5
ГЛАВА 3: Използвани технологии и приложен софтуер	7
3.1. C#	7
3.2. SQL (MySQL)	8
3.3. Microsoft SQL Server Management Studio (SSMS)	9
3.4. Visual Studio	11
3.5. .NET Framework	12
ГЛАВА 4: Конструкция на приложението	13
4.1. Регистрационна форма	13
4.2. Административна форма	14
4.3. Работна форма	19
4.4. Бази данни	22
ГЛАВА 5: Системни изисквания	25
ГЛАВА 6: Ръководство на потребителя	26
6.1. Влизане в системата	26
6.2. Работа с работна форма	26
6.3. Работа с административна форма	28
ГЛАВА 7: Бъдещо развитие на приложението	30
ГЛАВА 8: Тестове и заключение	31
Използвана литература	32

УВОД

В днешно време, въвеждането на компютърните и информационните технологии станаха неизменна част от нашето ежедневие. Хората започват да търсят по-рационален и универсален начин за извършването на техните задължения или за удобството им в тяхното ежедневие. Един чудесен пример за това са компютрите. Те се използват непрестанно, както у дома, така и на работното място. Също така, те са от решаващо значение за икономическия растеж на фирмите, които ги използват.

Според Националния статистическия институт на Република България, след направено изследване за използването на информационни и комуникационни технологии в домакинствата в страната, 87.3% от домакинствата в България имат достъп до интернет, като 79.0% от лицата на възраст между 16 и 74 навършени години използват интернет всеки ден или поне веднъж седмично на работа, вкъщи или на друго място. Според друго изследване, направено през 2019 от Българската софтуерна асоциация, около 57% от компаниите в България използват софтуер по поръчка.

Но колкото и да е разпространено използването на технологии, не всички сектори в бизнеса могат да защитят тази теза. Например, много фирми в малкия и средния бизнес като хранителните магазини не се водят систематизирани. Една от основните причини може да се посочи липсата на основна компютърна грамотност или високото нивото на сложност при повечето софтуерни програми.

В хранителния сектор се използват така наречените „ПОС Терминали“. Това е система, от английски означаваща „Точка на продажба“ което се води при нас така известните касови апарати. Те имат за цел да се създаде контрол върху продуктите излизащи от обекта, да се улесни калкулацията на голямо количество продукти и да се осчетоводят както и за фирмата под формата на фискален отчет, така и за клиента под формата на касов бон или фактура.

Повечето такива фирмени приложения, те трябва да бъдат рационално направени, удобни за потребителя, лесни разбираеми и привличащи окото с цел бързо запомняне на неговите основни функции. Трябва да извършват ефикасно своята логистична и административна функция, без това да се отрази като проблем за потребителя, независимо дали той е служител или управител. Също така трябва да бъде изработено по такъв начин, че да служителят да не изоставя със своите поръчки или заявки.

ГЛАВА 1: ЦЕЛИ И ЗАДАЧА НА ДИПЛОМАНТА РАБОТА

1.1. Удобство

- 1.1.1. Удобен, лесен и сигурен начин за извършване на многофункционални задачи от страна на потребителя, чрез лесен за използване интерфейс, който да осигури бърз и ефективен достъп до всички нужни инструменти и функции.
- 1.1.2. Стабилна структурирана система за следене на данните от страна на фирмата. Това ще позволи по-ефикасно извършване на основните функции на фирмата и извършването на стриктен контрол, както и на продуктите в обекта, като следене на внесените суми, така и на самите служители, като отчет на извършените поръчки от тяхно име.
- 1.1.3. Бързо действие и комуникация между продавач и клиент. По този начин, чрез използването на улеснени функции и лесен за употреба софтуер, да се намали до минимум работата върху един клиент във обекта за възможно най-кратко време.

1.2. Сигурност

- 1.2.1. Предотвратяване на неоторизиран достъп чрез ограничаването на употребата на софтуера да бъде извършвано само и единствено от служители и управители на обекта или лица, които имат предварително изготвена регистрация от потребител с административен достъп.
- 1.2.2. Разграничаване на потребителски от административни правомощия чрез изготвяне на отделни работни виртуални зони, като административен панел и работна зона.
- 1.2.3. Възможност за даване и отнемане на достъп на потребители с помощта на служители с предварително зададени административни правомощия.

ГЛАВА 2: ИЗСЛЕДОВАТЕЛСКИ ВЪПРОСИ И ТЕОРИТИЧНА РАМКА

2.1. Удобство на потребителя: Повечето продавач-консультанти в тяхната работна среда, решават да използват хартия и химикал. Това сериозно увеличава времето и сложността на дадена поръчка. Целта на проекта е да се ускори изпълнението на дадена задача, без това да затруднява продавача като изчисляването на сбора от дадени продукти, добавяне на отстъпка или въвеждането на стойности.

2.2. Употреба на продукта: Повечето потребители в тази сфера, нямат основна компютърна грамотност. Това може сериозно да затрудни употребата на продукта и да намали качеството на работа и оценката на потребителя. За целта, примерно решение може да бъде автоматизация на програмата. Чрез сканиране на продукта, информацията за него (количество, цена на брой и обща стойност) автоматично да се подреди в списък и да се натисне даден бутон за приключване на дадения касов бон. По този начин проекта ще бъде изготвен по такъв начин, че да може потребителя, без да има някаква компютърна грамотност, да може да използва програмата и без да изпитва никакво неудобство да пречи на работния процес.

2.3. Логистика: Изчисляването и поддържането на оборота може да бъде изключително натоварващо и може да попречи сериозно на работния процес. За целта, програмата ще има възможност да пресмята автоматично дневния, месечния и годишния оборот на магазина. Ще може да също и да коригира сумите, като изважда и въвежда служебно суми от оборота. Функцията може да има отрицателни или нулеви резултати, понеже обекта винаги има шанс да работи на загуба. С цел контрол на разходите, всяка сума трябва задължително да има основание за всяка една транзакция.

2.4. Контрол и сигурност: Оставянето на незаключен касов апарат, може да доведе до неправомерна употреба от неоторизирани лица. За целта, употребата на приложението може да става само от потребители, които разполагат с акаунт в програмата. Акаунтите биват да бъдат разделени на 2 типа: потребителски и административен. Те могат да биват застъпвани през регистрационната форма в програмата. При регистриране на акаунт с потребителски достъп, програмата изпраща потребителя в работната форма и може да извършва функцията на касов апарат, а при акаунт с административен достъп, програмата автоматично го изпраща в административното меню и може да контролира функциите и достъпа до програмата. Понеже регистрацията на потребител не е отворена, за да може да се създаде потребителски или административен акаунт, трябва самата регистрация да бъде извършена лично от потребител с административен достъп. Акаунтите ще имат основни разлики във функционалността си: Потребителите ще имат възможност да използват основните работни функции на програмата, да създават и коригират касовия бон, да въведат връчената стойност и да изчисляват ресто върху нея. Администраторите ще имат възможност да създават дневен, месечен и годишен отчет върху паричния оборот, служебно да въвеждат изваждат парични стойности, да добавят, премахват и редактират потребителски и административни акаунти и да добавят, премахват или редактират артикули.

ГЛАВА 3: ИЗПОЛЗВАНИ ТЕХНОЛОГИИ И ПРИЛОЖЕН СОФТУЕР

3.1. C#



Фиг. 1: Официалното лого на C#

C# (фиг. 1) е силно типизиран, обектно-ориентиран програмен език, създаден през 2000 година от Microsoft. Името на езика произлиза от английската му нотация C Sharp (C#), която е символна нотация за увеличение на нотата на полутонон.

C# е един от най-използваните програмни езици за създаване на приложения за операционната система Windows. Езикът е разработен с цел да бъде лесен за използване и да предостави по-голяма сигурност на програмите чрез проверка на типовете по време на компилацията.

Езикът поддържа много функционалности, включително делегати, събития, анонимни типове, LINQ, асинхронно програмиране и др. C# е също така език за програмиране, който може да бъде използван за създаване на различни видове приложения, включително уеб, настолни, мобилни и игри.

Езикът се използва широко в Microsoft технологиите като .NET Framework, .NET Core, Xamarin и Unity. C# е един от най-популярните езици за програмиране в света и се използва от много програмисти за разработка на софтуер за много от най-големите компании в света.

C# е проектиран да бъде език със стриктни правила за типовете данни, което го прави по-безопасен и сигурен от други езици за програмиране, като например

C++ или JavaScript. Това означава, че програмите, написани на C#, ще бъдат по-стабилни и по-малко склонни към грешки и проблеми.

C# поддържа множество структури от данни и алгоритми, които могат да бъдат използвани за обработка на данни и оптимизация на програмите. Езикът е също така силно интегриран със средата за разработка Visual Studio, която предоставя мощни инструменти за разработка и отстраняване на грешки.

C# е съвместим със много други езици за програмиране и позволява лесна интеграция с различни приложения. Той може да се използва за създаване на приложения за мобилни устройства, компютърни игри, уеб приложения и дори за програмиране на роботи и IoT устройства.

С издаването на .NET Core, C# е станал един от най-популярните езици за програмиране за разработчиците на уеб приложения, като позволява създаването на ефективни уеб приложения.

C# е мощен, универсален и сигурен език за програмиране, който се използва от много програмисти в целия свят. Той е особено полезен за разработка на софтуер за операционната система Windows и за създаване на уеб приложения, мобилни приложения и компютърни игри.

3.2. SQL (MySQL)



Фиг. 2: Официалното лого на SQL

SQL (Structured Query Language) (фиг. 2) е език за управление на релационни бази данни (RDBMS), който се използва за създаване, манипулиране и запитване на данни в тези бази данни. SQL е стандартизиран език, който е приет от много производители на релационни бази данни, като например Oracle, Microsoft SQL Server, MySQL, PostgreSQL и др.

SQL позволява на потребителите да извършват множество операции върху данните в релационните бази данни, включително добавяне, изтриване и обновяване на данни, създаване на нови таблици и структури, извличане на данни от тези таблици и извършване на различни агрегатни функции върху тези данни.

SQL има синтаксис, който е лесен за научаване и използване. Той се състои от заявки, които се изпращат към базата данни и резултатите от тези заявки се връщат към потребителя. Заявките могат да бъдат изградени по много начини, включително чрез използване на оператори за филтриране, сортиране и групиране на данни.

SQL е един от най-широко използваните езици за програмиране в света и е ключов инструмент за мнозина програмисти и администратори на бази данни. Той е особено полезен за компании и организации, които съхраняват големи количества данни и имат нужда от бърз и ефективен начин за манипулиране на тези данни.

SQL е мощен език за управление на релационни бази данни, който се използва в широк спектър от приложения, включително уеб приложения, мобилни приложения, електронна търговия и други. Той е ключов инструмент за множество компании и организации, които съхраняват и манипулират големи количества данни.

3.3. Microsoft SQL Server Management Studio (SSMS)



Фиг. 3: Официалното лого на Microsoft SQL Server

Microsoft SQL Server Management Studio (SSMS) (фиг. 3) е софтуер, който представлява интегрирана среда за управление на SQL Server бази данни. Този инструмент е предназначен да подпомага администраторите и разработчиците

при управлението на SQL Server, като позволява на потребителите да създават, манипулират и управляват обекти в базите данни.

SSMS е графичен инструмент, който предлага множество функционалности и опции, които могат да бъдат използвани от потребителите. Сред тези функционалности са включени обектова навигация, редактор на T-SQL, инструмент за управление на версиите, инструмент за анализ на производителността и инструмент за създаване на резервни копия.

Обектовата навигация представлява система, която позволява на потребителите да разглеждат и управляват обекти в базите данни. Тези обекти включват таблиците, индексите, процедурите, функциите, изгледите и много други.

Редакторът на T-SQL позволява на потребителите да пишат и изпълняват заявки в T-SQL. Редакторът предоставя много полезни функции, като автоматично допълване на кода, форматиране на кода и синтаксис на цветове.

Инструментът за управление на версиите позволява на потребителите да контролират версиите на обектите в базите данни.

Инструментът за анализ на производителността позволява на потребителите да анализират производителността на базите данни и да идентифицират проблеми в тях.

Инструментът за създаване на резервни копия позволява на потребителите да създават резервни копия на базите данни и да ги възстановяват при необходимост.

SSMS е мощен инструмент, който може да бъде използван за управление на SQL Server бази данни от много различни типове потребители.

3.4. Visual Studio



Фиг. 4: Официалното лого на Visual Studio

Visual Studio (фиг. 4) е интегрирана среда за разработка на софтуер, която позволява на програмистите да създават и поддържат различни видове приложения, включително десктоп, уеб, мобилни и игри. Тази среда е разработена от Microsoft и е налична за множество операционни системи, включително Windows и MacOS.

Visual Studio предлага множество инструменти и функции, които позволяват на програмистите да работят по-ефективно и да създават по-качествен софтуер. Тези инструменти включват интегрирано отстраняване на грешки, средства за тестване на кода, инструменти за управление на източниците, графични редактори за създаване на потребителски интерфейси и много други.

Относно езиците, с които работи Visual Studio, те включват C++, C#, Visual Basic и други. Сред тези езици, C# е един от най-широко използваните и се използва за създаване на множество видове приложения.

Visual Studio е разработена за да подпомага процеса на разработка на софтуер, като предоставя на програмистите средства за бързо и лесно създаване на код, тестване на приложения и разпространение на готовия продукт. Тази среда може да бъде използвана от програмисти на всяко ниво на опит и могат да работят в различни отрасли като софтуерни компании, образователни институции, държавни институции и други.

Visual Studio е интегрирана среда за разработка на софтуер, която предлага на програмистите много инструменти и функции за бързо и ефективно създаване на качествен софтуер. Тази среда е създадена от Microsoft и се използва от много програмисти по света.

3.5. .NET Framework



Фиг. 5: Официалното лого на Visual Studio

.NET Framework (фиг. 5) е софтуерен фреймуърк, разработен от Microsoft, който предоставя среда за изпълнение на софтуерни приложения на различни програмни езици, включително C#, VB.NET, F#, и други. Фреймуъркът включва голяма библиотека от класове, която предоставя много готови функционалности, като работа с файлове, манипулация на XML данни, работа с мрежи и интернет протоколи, сигурност и криптиране, работа с бази данни и други. Класовете от тази библиотека могат да бъдат използвани от разработчиците, за да създават софтуерни приложения по-бързо и по-лесно.

.NET Framework предоставя Common Language Runtime (CLR), която е среда за изпълнение на .NET приложения. Тя управлява изпълнението на кода и осигурява автоматично управление на паметта, управление на изключенията и други функционалности. Фреймуъркът също предоставя компилатори за няколко програмни езика, включително C#, VB.NET, F#, и други.

Visual Studio е основната среда за разработка на приложения на .NET платформата. Тя осигурява интуитивен интерфейс за създаване, тестване и деплоиране на приложения. Освен това, .NET приложения могат да бъдат създадени за различни операционни системи, включително Windows, macOS и Linux, което осигурява многоплатформеност.

.NET Framework е мощен инструмент за разработка на софтуерни приложения, който осигурява богата библиотека от класове, среда за изпълнение на кода, интеграция със Visual Studio, и многоплатформеност. Той е изключително полезен за разработчиците, които искат да създават мощни, сигурни и бързи приложения.

ГЛАВА 4: КОНСТРУКЦИЯ НА ПРИЛОЖЕНИЕТО

4.1. Регистрационна форма (фиг. 6.1): Представява малка форма, поставена върху под форма, която извършва функцията на заден фон. Основната формата проверява дали даден потребител е регистриран в базата данни. Тя проверява потребителското име, паролата и нивото на достъп на потребителя. Също така системата проверява, дали има празни текстови полета. Самата регистрационна форма е разработена с помощта на пакет за графична обработка GunaUI. По този начин, програмата предлага приятен и динамичен дизайн. Всички компоненти във формата са интерактивни и се променят с всяко изключение.

Ако поради някаква причина програмата няма възможност да се свърже със базата данни, тогава ще се появи най-отгоре на екрана голямо червено балонче което ще уведоми потребителя, че системата няма достъп до базата данни и няма да има възможност да се свърже със административната или работната форма.

Когато потребителското име или парола не са въведени в съответните полета, но потребителя натисне бутона Login, текстовите полета ще се маркират в червено, интерактивната икона ще се преобрази в предупредителна и текста под нея ще се промени на „Please enter username and password“.

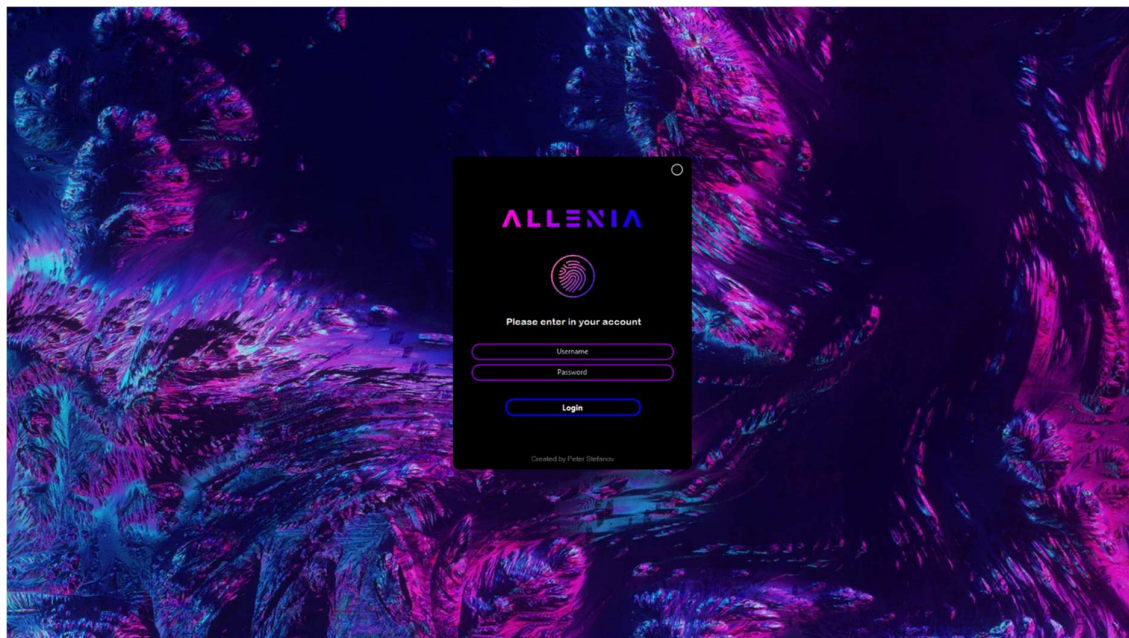
Когато потребителското име или парола не съвпадат с тези в базата данни, но потребителят натисне бутона Login, текстовите полета ще се маркират в червено и ще се изчистят, интерактивната икона ще се преобрази в предупредителна и текста под нея ще се промени на „Invalid username or password“.

Когато всички условия са изпълнени и в базата данни има регистриран потребител с въведените данни, то програмата започва да проверява нивото на достъп на потребителя.

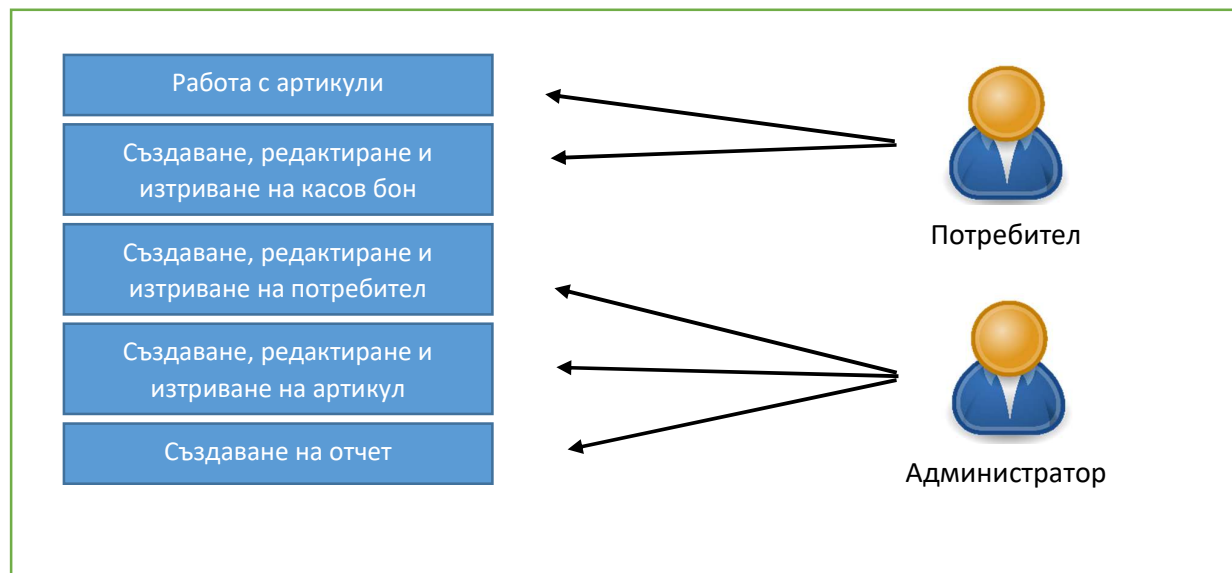
Ако потребителя разполага с потребителски достъп, програмата ще го преведе към работната форма.

Ако потребителя разполага с административен достъп, то програмата ще затвори малката форма и на нейно място върху същата под форма ще излезе самата административна форма.

Потребителите ще имат възможност да използват основните работни функции на програмата, да създават и коригират касовия бон, да въведат връчената стойност и да изчисляват ресто върху нея. Администраторите ще имат възможност да създават дневен, месечен и годишен отчет върху паричния оборот, служебно да въвеждат изваждат парични стойности, да добавят, премахват и редактират потребителски и административни акаунти и да добавят, премахват или редактират артикули. (Фиг. 6.1)



Фиг. 6.1: Визуална графика на регистрационна форма



Фиг. 6.2: Функции на потребителите и администраторите

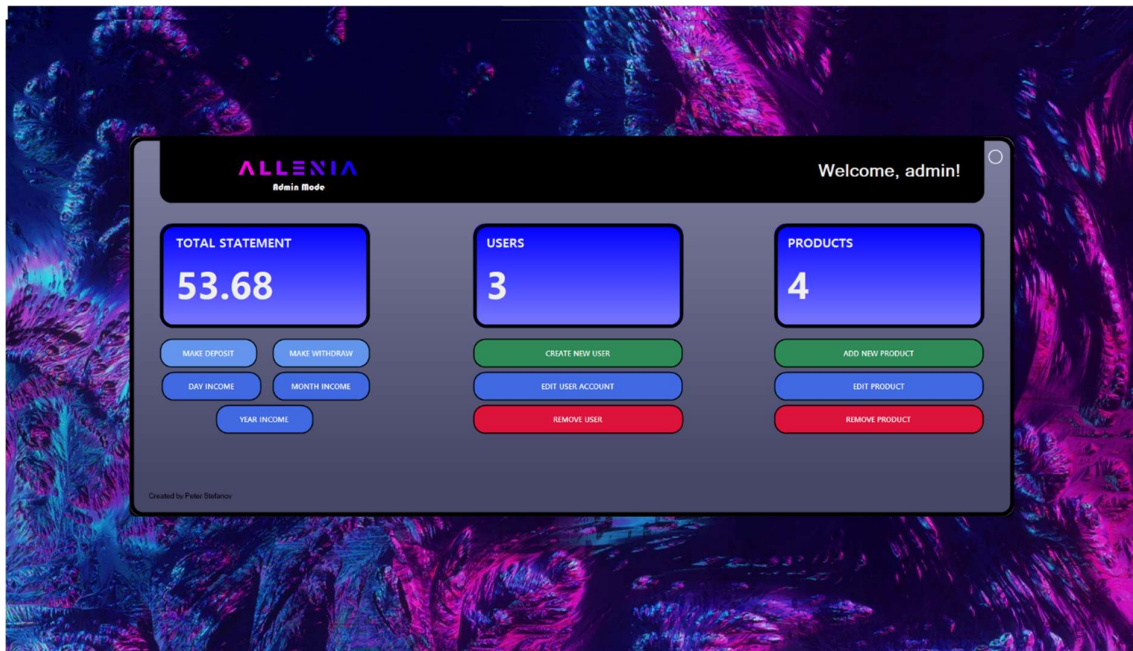
4.2. Административна форма (фиг. 7): Представява малка форма, поставена върху под форма, в която може да се влезе само, ако потребителят е въвел в регистрационната форма, акаунт с административен достъп. Формата представлява таблично подредени опции, представящи различните опции с които потребителя разполага.

Административната форма разполага със следните опции:

- издаване на дневен отчет
- издаване на месечен отчет
- издаване на годишен отчет
- добавяне на потребител
- редактиране на потребител
- изтриване на потребител
- добавяне на артикул
- редактиране на артикул
- изтриване на артикул.

Също така са поставени 3 кутийки, които да показват общия оборот на магазина, общия брой потребители и общия брой продукти.

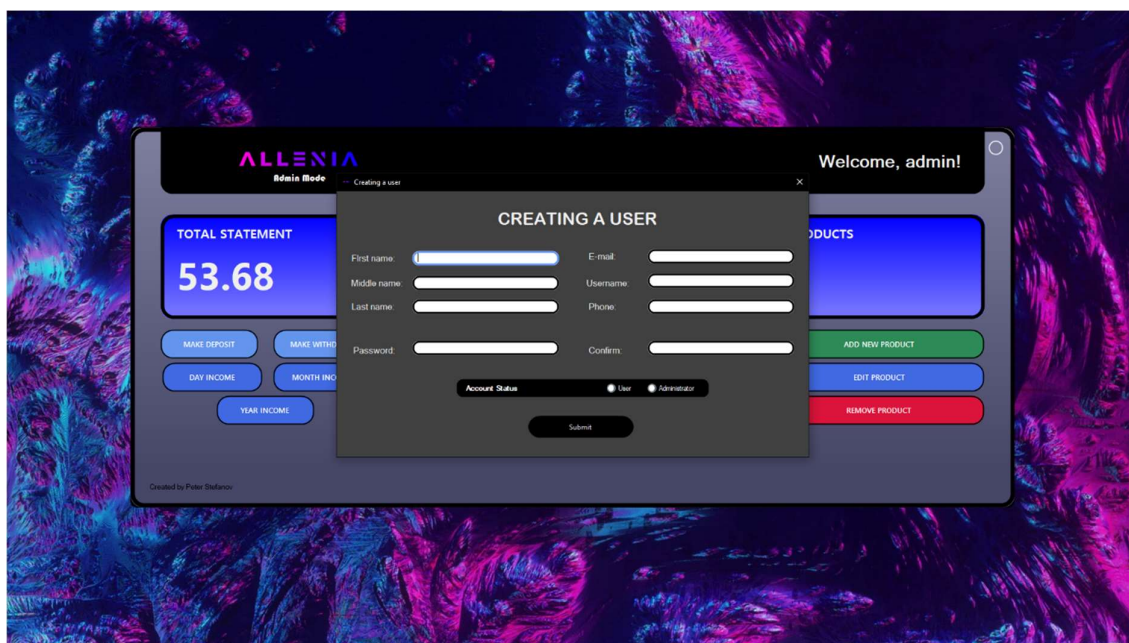
При издаването на дневен, годишен или месечен отчет (фиг. 7.1), ще се отвори допълнителна малка форма на която да се напише времеви период за който ще се отнася отчета. След въвеждането на датата, се изписва на екрана директно колко клиенти са пазарували през този срок и какъв оборот е бил направен през него.



Фиг. 7.1: Визуална графика на издаване на отчет

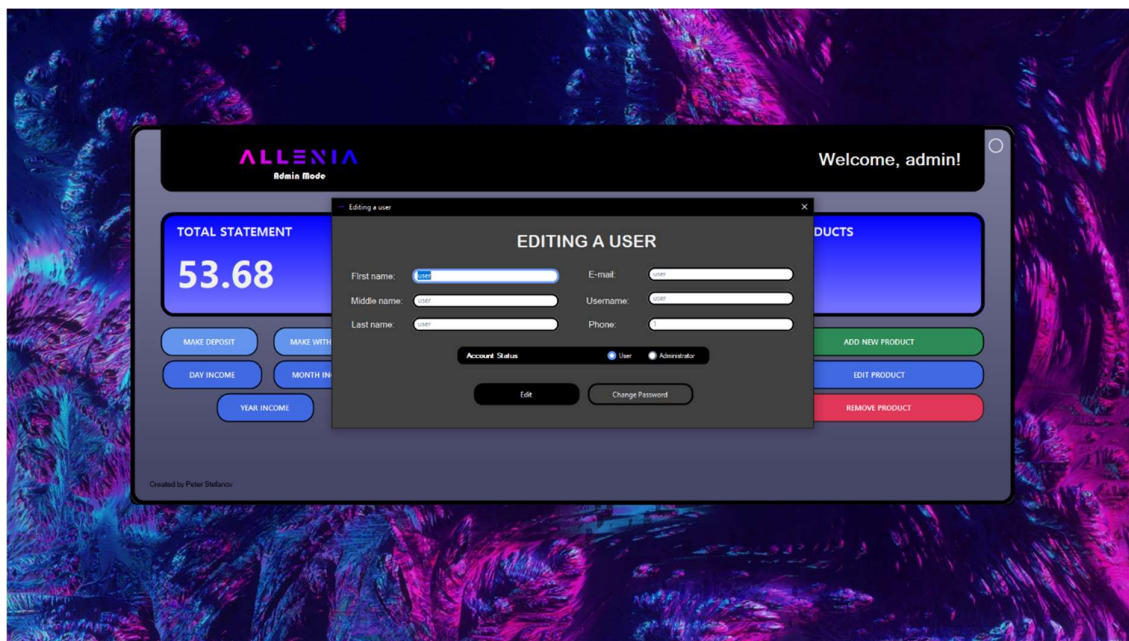
При служебното внасяне и изваждане на стойности в системата, ще се отвори допълнителна малка форма в която да се напише конкретната стойност, която потребителя желае да въведе. След въвеждането на стойността, програмата изпраща съобщение, че сумата е записана успешно.

При добавяне на потребител (фиг. 7.2) ще се отвори нова форма, наподобяваща на регистрационна анкета. В нея трябва да се попълни име, презиме, фамилия, емайл, телефон, парола и потвърждение на паролата. След попълването на необходимите данни, информацията от формата се изпраща в базата данни и вече може да бъде валидно въведена от регистрационната форма.



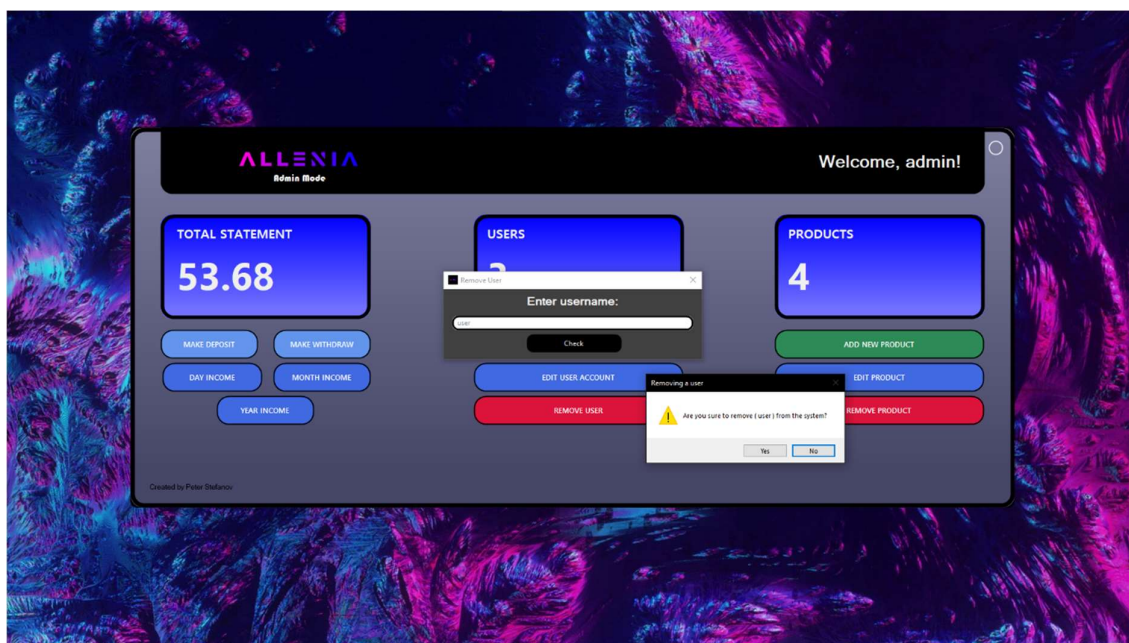
Фиг. 7.2: Визуална графика на създаване на потребител

При редактиране на потребител (фиг. 7.3) ще се отвори малка форма, съдържаща текстово поле и бутон. След въвеждане на избрания потребител ще се отвори нова форма, наподобяваща на регистрационната анкета, но разликата, че данните ще бъдат вече въведени в текстовите полета. След промяната на данните, редактираната информация ще замени старите данни с новите в базата данни и ще могат да бъдат въведени от регистрационната форма.



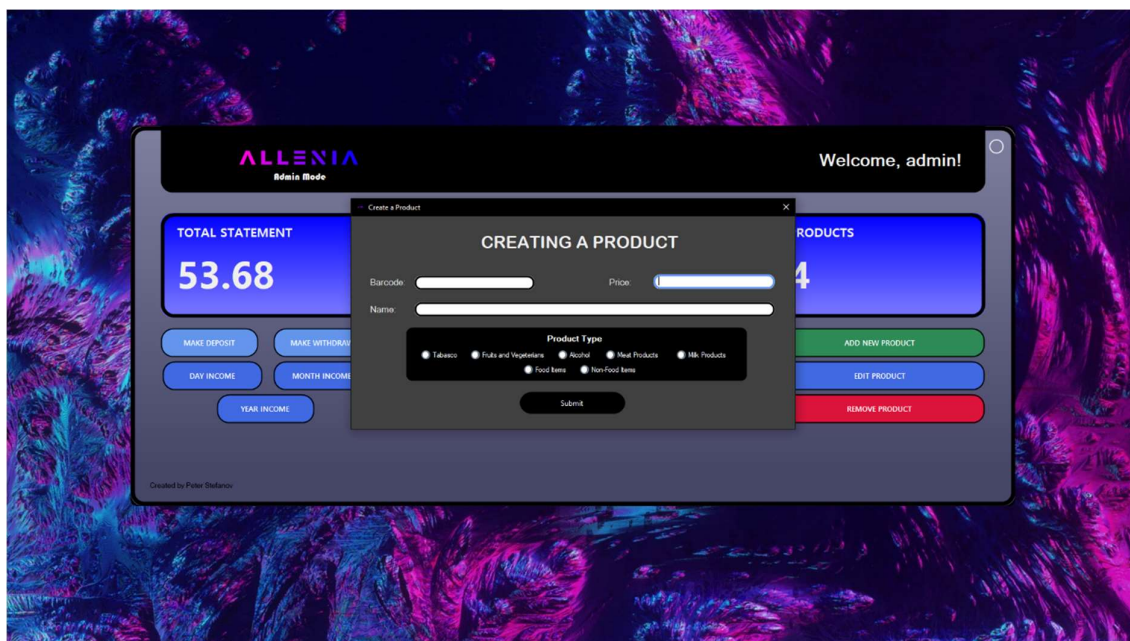
Фиг. 7.3: Визуална графика на редактиране на потребител

При премахване на потребител (фиг. 7.4) ще се отвори нова форма, съдържаща текстова кутия и бутон. след въвеждане на избрания потребител и натискане на бутона, програмата ще поиска потвърждение дали е сигурен, че иска да изтрие този потребител. След потвърждението, информацията за избрания потребител ще бъде изтрита от базата данни с потребителите, но няма да променя останалите бази данни, където потребителя е извършвал интеракции с програмата.



Фиг. 7.4: Визуална графика на изтриване на потребител

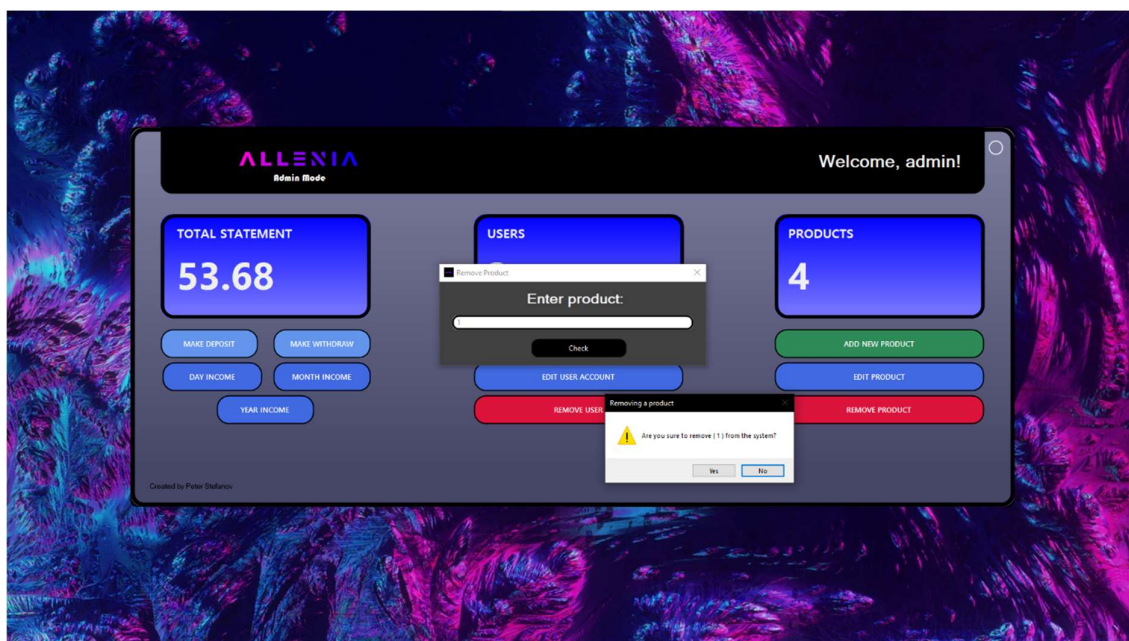
При добавяне на артикул (фиг. 7.5), ще се отвори нова форма, наподобяваща на регистрационна анкета. В нея да се попълни име на артикул, цена, баркод и категория на артикула. След попълването на необходимите данни за артикула, програмата ще ги въведе в базата данни с артикули и ще бъдат готови за сканиране в работната форма.



Фиг. 7.5: Визуална графика на създаване/редактиране на артикул.

При редактиране на артикул, ще се отвори малка форма, която ще изисква да се въведе баркода на артикула. След сканирането, ще се отвори друга форма, наподобяваща на регистрационна анкета, но с разликата, че данните в нея вече ще бъдат добавени в текстовите полета. След промяната на данните, редактираната информация ще замени старите данни с новите в базата данни и ще могат да бъдат избирани от работната форма.

При изтриване на артикул (фиг. 7.6), че се отвори малка форма, която ще изисква да се въведе баркода на артикула. След сканирането, програмата ще пита администратора дали е сигурен, че иска да изтрие артикула с даденото име. След потвърждението, артикула ще бъде изтрит от базата данни.



Фиг. 7.6: Визуална графика на изтриване на артикул.

4.3. Работна форма (фиг. 8) – Представява форма, заемаща целия екран, в която може да се влезе само, ако потребителя е въвел в регистрационната форма, акаунт с потребителски достъп. Формата представлява няколко сортирани панели, разполагащи с различни функции, необходими за работата на един продавач в хранителен магазин. Формата разполага с панел със списък с продукти, клавиатурен панел, панел за изготвяне на касов бон и панел за информацията на касиера.

Панела със списъка с продукти (фиг. 8А) , представлява списък тип DataGridView в която ще се извеждат всички продукти от базата данни, с цел да се помогне на потребителя да намери по-бързо даден продукт. Също така с цел, по-удобно търсене, има създадени допълнителни бутони за сортиране с които да се избере какъв продукт точно потребителят търси. Когато продукта бъде успешно намерен, може да бъде избран с помощта на мишката и той ще бъде изведен при клавиатурния панел с цел коригиране на баркода, преди да бъде изведен в касовия бон.

Клавиатурния панел (фиг. 8В), представлява множество бутони, образувачи клавиатура и текстово поле, което потребителят да използва за въвеждане на баркодове и изчисляване на количество. Клавиатурата разполага с бутон за утвърждаване на десетични стойности, бутон за изчисление на количеството,

бутон за изчистване на текстовото поле и бутон за извеждане на информацията от текстовото поле в панела за изготвяне на касовия бон.

За да се изведе една информация за даден продукт в касовия бон, трябва да се въведе баркода в текстовото поле с помощта на клавиатура на компютъра или на приложението, или да се избере от списъка с продукти, след което да се натисне бутона за извеждане (изписан в приложението с инициалите TTL).

Ако потребителят не въведе нищо в текстовото поле, или информацията в него не е валидна, приложението ще изпише съобщение, че информацията не е валидна и да се коригира.

Ако потребителят желае да се въведе нетно количество на даден продукт, трябва първо да се въведе баркода на артикула, след това да се натисне на клавиатурата бутона за изчисление на количество (изписан в приложението със знака X) и на финала да се въведе количеството на артикула. Ако количеството се води нетно, може да се използва бутона за утвърждаване на десетични стойности (изписан в приложението с точка).

Не е задължително за потребителя да въвежда за всеки артикул количество, ако той се води само един. Функцията се използва само ако потребителя желае да закупи повече от един продукт или ако количеството му е нетно.

Ако потребителят желае да изчисти текстовото поле поради грешка, може да използва бутона за изчистване на текстовото поле (изписан в приложението с инициалите C) и по този начин да коригира поръчката, без да я извежда в касовия бон.

Панела за изготвяне на касовия бон (фиг. 8С), представлява списък тип DataGridView в която да се изведат всички селектирани артикули от потребителя, заедно с тяхното количество, цена на продукта и неговата умножена изчислена стойност. Към списъка има разпечатващи текстови полета за общата стойност на поръчката, връчената сума от клиента и нужното ресто, което трябва да се върне на клиента. Към тях, панела разполага с бутони за корекция, бутон за изчистване на касовия бон, бутони за отстъпка и бутон за приключване на касовия бон.

След като са въведени серия от продукти в списъка с поръчката, въведени от клавиатурния панел, потребителят трябва да въведе сумата, връчена от клиента в

текстовото поле, надписано с надпис „Cash“. Програмата ще изчисли рестото, базирано на общата сума на поръчката и сумата, връчена от клиента.

Ако в програмата не е въведена връчена сума от клиента, или е по-малка от общата стойност на продуктите, на потребителя няма да бъде разрешено да приключи касовия бон.

Ако потребителят желае да начисли отстъпка върху поръчката на клиента, може да се възползва от бутоните за отстъпка. Отстъпките варират от 10 и 20 процента. За да се начисли отстъпката, потребителя трябва първо да въведе всички продукти в касовия бутон, преди да се избере опцията (в противен случай програмата няма да изчисли правилно крайните стойности). Отстъпката може да се използва само веднъж на поръчка, при което след въвеждането ѝ, програмата заключва бутоните до започването на нова поръчка.

Ако потребителят желае да коригира списъка с поръчката, може да използва бутона за корекция. При натискането му, програмата иска потвърждение, дали да се изпълни операцията. След приемането, програмата премахва последния артикул от списъка и премахва неговата стойност от изчисленията. Бутонът може да бъде използван неограничен брой пъти. Ако бутонът бъде натиснат, когато няма никакви поръчки в списъка, програмата ще извести, че няма никакви артикули в списъка.

Ако потребителят желае да прекрати поръчката и да занули всички стойности във формата, може да използва бутона за зануляване. При натискането му, програмата иска потвърждение, дали да се изпълни операцията. След приемането, програмата рестартира формата и по този начин всичко се връща в първоначален вид.

След като потребителят въведе всички артикули, при желание въведе отстъпка и бъде въведена сумата, връчена от клиента, може вече да се приключи поръчката с бутона за приключване на касовия бон (надписан в програмата с надпис Checkout). При натискането му, програмата прави необходимите проверки на стойностите, въвежда всичката информация в базата данни, разпечатва разписката със закупените артикули и занулява формата за да може да бъде готова за следващата поръчка.

Панела за информацията на касиера (фиг. 8D), представлява текстово поле, съдържащо името на касиера, работещо на програмата, бутон за излизане от профила и бутон за излизане от приложението.

Ако потребителя желае да излезе от работната форма и да се върне на регистрационната форма, може да използва бутона за излизане от профила (изписан в програмата с надпис “Log out”).

Ако потребителя желае да излезе от работната форма и да напусне приложението, може да използва бутона за излизане от приложението (изписан в програмата с надпис „Exit“).



Фиг. 8: Визуална графика на административната форма

- A. Списък с продукти B. клавиатурен панел C. Панел за изготвяне на касов бон
D. Панел с информацията на касиера

4.4. Базы данни – Базата данни на програмата съдържа 3 таблици (фиг. 9) – Users, Products и Statements.

Таблица „Users”, съдържа информацията за всички служители, разполагащи с профил в приложението. Таблицата се състои от:

- userID – поле от целочислен тип, което съдържа уникалния номер на потребител.

- username – поле от низов тип, съдържащо потребителското име на потребителя.
- firstName – поле от низов тип, съдържащо първото име на потребителя.
- fatherName – поле от низов тип, съдържащо бащиното име на потребителя.
- lastName - поле от низов тип, съдържащо фамилното име на потребителя.
- phone - поле от низов тип, съдържащо телефонния номер на потребителя.
- email - поле от низов тип, съдържащо електронната поща на потребителя.
- passwd - поле от низов тип, съдържащо паролата на потребителя.
- userStatus – поле от целочислен тип, съдържащо информация за статуса на потребителя (администратор или потребител)

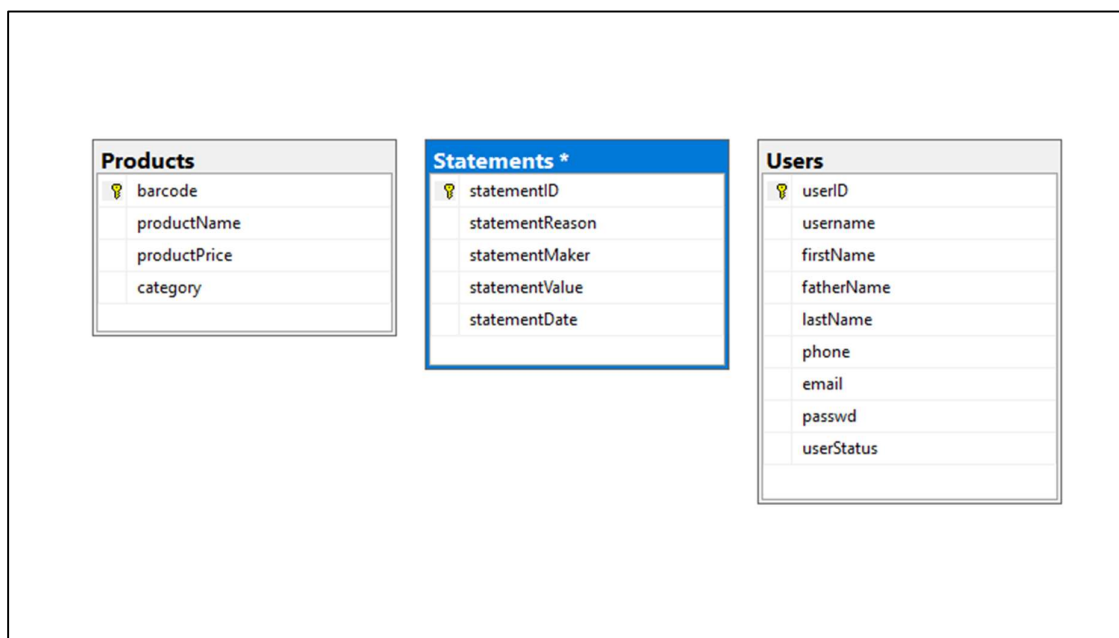
Таблицата „Products”, съдържа информацията за всички продукти с които програмата разполага и могат да се използват в приложението. Таблицата се състои от:

- barcode - поле от целочислен тип, което съдържа уникалния barcode на продукта
- productName поле от низов тип, съдържащо името на продукта.
- productPrice – поле от реален тип, съдържащо цената на продукта.
- category – поле от целочислен тип, съдържащо информация за категорията на продукта.

Таблица „Statements” съдържа информацията за всички направени поръчки от работната форма. Таблицата се състои от:

- statementID – поле от целочислен тип, съдържащо уникалния номер на поръчката.
- statementReason – поле от низов тип, съдържащо причината за направената поръчка.
- statementMaker – поле от низов тип, съдържащо името на създателя на поръчката.
- statementValue - поле от реален тип, съдържащо стойността на поръчката.

- statementDate – поле за дата, съдържащо датата на която е направена поръчката.



Фиг. 9: Схема на базата данни

ГЛАВА 5: СИСТЕМНИ ИЗИСКВАНИЯ

5.1. Хардуерни изисквания:

- Процесор: 1 GHz или по-бърз;
- Памет: 4 GB;
- Място на твърдия диск: 4 GB свободно място;
- Разделителна способност на екрана: 1920 x 1080 пиксела с мащабиране на 100% (ако не бъде спазена тази резолюция, програмата няма да може да се визуализира правилно);
- Графична карта: поддръжка на DirectX 9 или по-нова версия с WDDM 1.0 драйвер;

5.2. Софтуерни изисквания:

- Операционна система: Windows 10;
- .NET Framework: версия 4.7.2 или по-нова;
- Visual Studio: версия 2022 или по-нова (ако използвате Visual Studio за разработка на приложението).
- SSMS: версия 19 или по-нова със сървър версия 16.0.1000.6 или по-нова.
- Прикачена база данни към SSMS - dpDatabase

ГЛАВА 6: РЪКОВОДСТВО НА ПОТРЕБИТЕЛЯ

6.1. Влизане в системата:

- При стартиране на приложението, данните на потребителя се въвеждат във полетата за потребителско име и парола и се натиска бутона „Login”. Ако разполага с административен акаунт, се появява административна форма (т. 6.3). В противен случай, ако разполага с потребителски акаунт, ще се появи работна форма (т. 6.2).

6.2. Работа с работна форма:

А. Въвежда се баркода на продукта в текстовото поле (фиг. 10А) с помощта на клавиатурата на екрана (Фиг. 10В) (или на компютърното устройство) или с натискане на някой от изложените артикули в списъка с продукти (Фиг. 10С) вдясно с помощта на мишката.

- За по-удобно търсене в списъка с продукти, се използват бутоните за сортиране (Фиг. 10D) под него.
- За да се въведе количеството на даден продукт, първо се въвежда баркода в лентата, след това се натиска бутона X (Фиг. 10E) за умножение на количеството (ако количеството е нетно, трябва да се използва точката за разделение (Фиг. 10F)) и зад знака количеството на продукта.
- За да се изчисти лентата, поради направена грешка в изписването, се натиска червения бутон C (Фиг. 10G).

Б. След въвеждане на баркода (и количеството) се натиска синия бутон TTL (Фиг. 10H) за да се въведе информацията в касовия бон (Фиг. 10I).

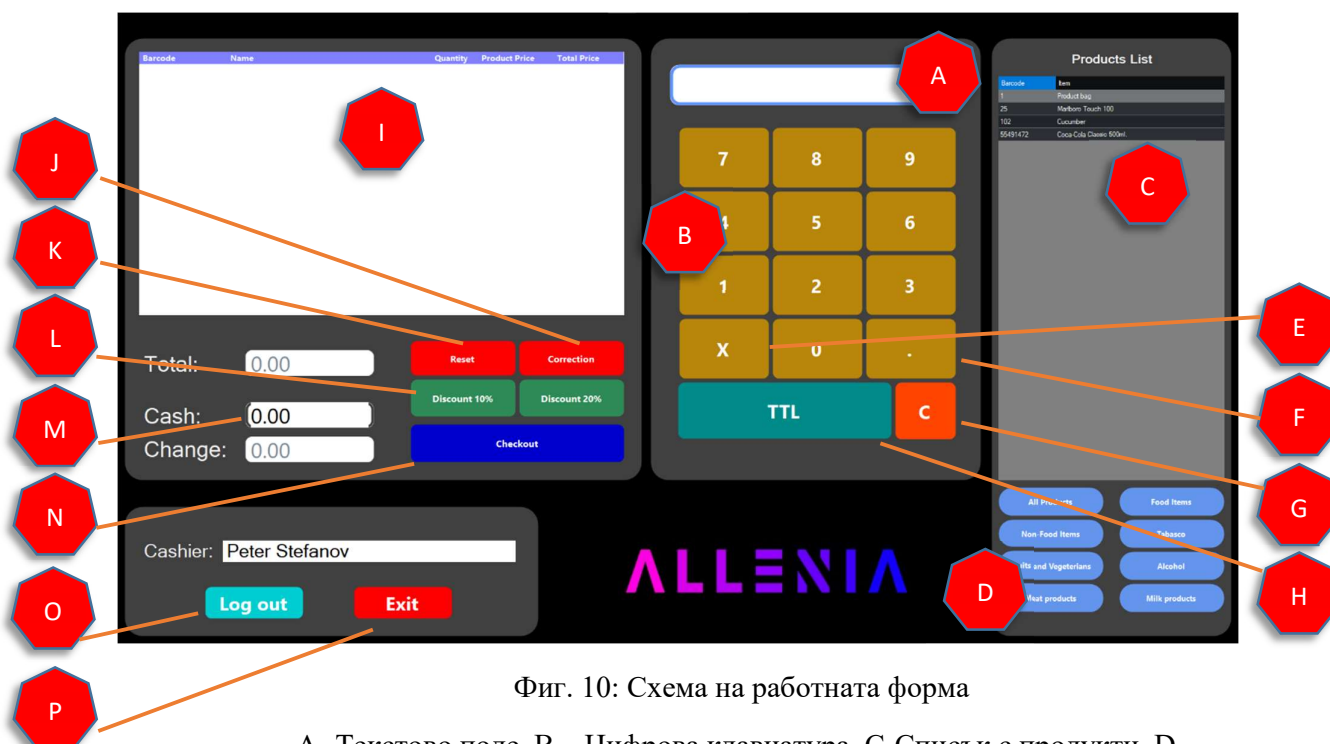
- За да се изчисти даден продукт от касовия бон, поради грешка, се натиска червения бутон с надпис “Correction” (Фиг. 10J). По този начин ще се изтрие последния продукт от касовия бон.
- За да се занули касовия бон и да се започне нова поръчка, се натиска червения бутон с надпис „Reset” (Фиг. 10K).
- За да се въведе отстъпка на поръчката, може да използват зелените бутони за отстъпки от 10 или 20 процента (Фиг. 10L) от общата сума

на касовия бон (отстъпката се въвежда само ако след като всички продукти са въведени в касовия бон).

В. След въвеждането на продуктите, се натиска с мишката върху кутийката с надпис „Cash” (Фиг. 10М) и се въвежда сумата, връчена от клиента. Програмата ще изчисли рестото, което е нужно да се върне въз основа на общата сума и сумата, връчена от клиента.

Г. Натиска се тъмносиния бутон “Checkout” (Фиг. 10N) за да се разпечата касовия бон и да се започне нова поръчка.

- За да се излезе от профила на потребителя, без да се затваря програмата, се натиска светлосиния бутон „Log out” (Фиг. 10O) за да се върне в менюто за регистрация.
- За да се излезе от програмата, натиснете червения бутон „Exit” (Фиг. 10P).



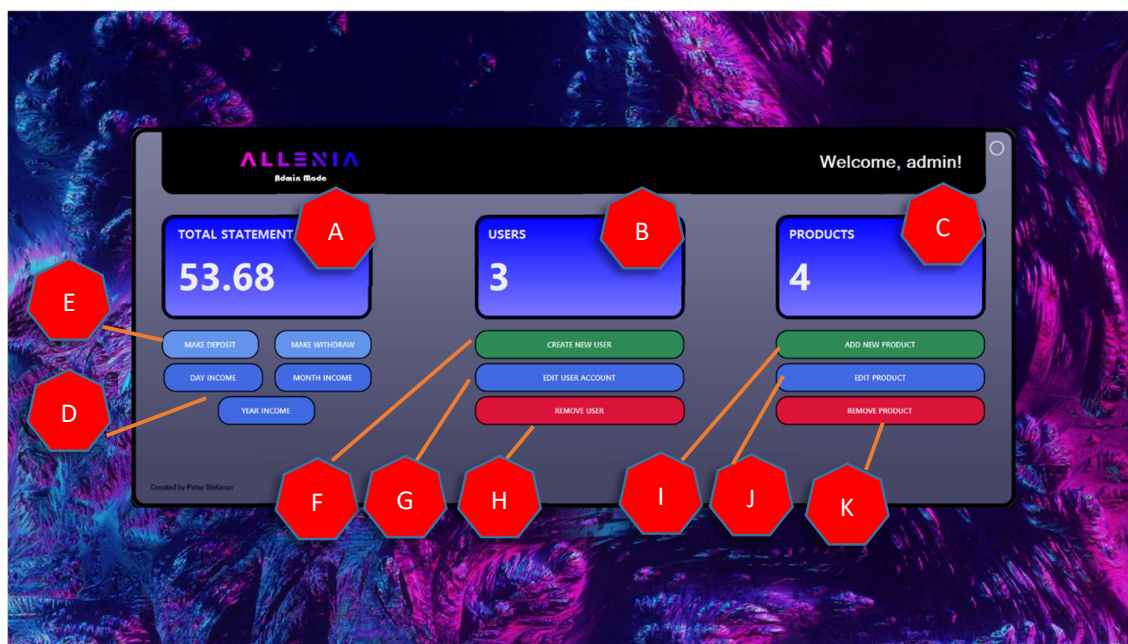
Фиг. 10: Схема на работната форма

А- Текстово поле, В – Цифрова клавиатура, С-Списък с продукти, D- бутони за сортиране, Е- символ за умножение, F- точка за разделяне, G- бутон за изчистване, H- бутон за въвеждане, I- списък за касов бон, J- бутон за корекция, K- бутон за нулиране, L- бутони за отстъпка, M- поле за получена сума, N – бутон за приключване на поръчката, O – бутон за излизане от профила, P – бутон за излизане от приложението

6.3. Работа с административна форма:

- За да се прегледа фирмения оборот на обекта, трябва да се прегледа резултата в кутийката с надпис „Total statement” (Фиг. 11А).
- За да се прегледа броя регистрирани потребители в програмата, трябва да се прегледа резултата в кутийката с надпис „Users” (Фиг. 11В).
- За да се прегледа броя регистрирани продукти в програмата, трябва да се прегледа в кутийката с надпис „Products” (Фиг. 11С).
- За да се направи дневен, месечен или годишен отчет за точно конкретна дата, се натиска върху бутона с надпис „Day income”, „Month Income” или „Year Income” (Фиг. 11D) в зависимост от самия вид отчет. В прозореца, се въвежда датата за която потребителя желае да направите отчет и натиска бутона „Check”. Резултатите ще се изпишат отдолу.
- За служебно въвеждане и изваждане на стойност във и от отчета, се натиска бутона „Make deposit” или „Make withdraw” (Фиг. 11Е) в зависимост от операцията. В прозореца се въвежда сумата, която да внесете или изтеглите и натиснете бутона „Check”. Ако всичко е вярно, изписва съобщение, че операцията е успешна.
- За да се създаде нов потребител, се натиска върху бутона „Create new user” (Фиг. 11F). В прозореца се въвежда необходимата информация, изписана във формуляра и се натиска бутона „Submit”. Ако всичко е вярно, програмата изписва съобщение, че операцията е успешна.
- За да се редактира даден потребител, се натиска върху бутона „Edit user account” (Фиг. 11G). В прозореца се въвежда потребителското име на потребителя и се натиснете бутона „Check”. След въвеждането, програмата извежда цялата информация за потребителя. След редакцията се натиска бутона „Submit”. Ако всичко е вярно, програмата изписва съобщение, че операцията е успешна.
- За да се изтрие даден потребител, се натиска върху бутона „Delete user” (Фиг. 11H). В прозореца се въвежда потребителското име на потребителя и се натиска бутона „Check”. Програмата ще попита за потвърждение за изтриване на потребителя. След потвърждение, програмата изписва съобщение, че операцията е успешна.

- За да се добави нов артикул, се натиска върху бутона „Add new product”. (Фиг. 11I). В прозореца се въвежда необходимата информация, изписана във формуляра и се натиска бутона „Submit”. Ако всичко е вярно, програмата изписва съобщение, че операцията е успешна.
- За да се редактира даден артикул, се натиска върху бутона „Edit product”(Фиг. 11J). В прозореца се въвежда баркода на продукта и се натиска бутона „Check”. След въвеждането програмата извежда цялата информация за продукта. След редакцията се натиска бутона “Submit”. Ако всичко е вярно, програмата изписва съобщение, че операцията е успешна.
- За да се изтрие даден артикул, се натиска върху бутона „Delete product”(Фиг. 11K). В прозореца се въвежда баркода на продукта и се натиска бутона „Check”. Програмата ще попита за потвърждение за изтриване на артикула. След потвърждение, програмата изписва съобщение, че операцията е успешна.



Фиг. 11: Схема на административната форма

А-Баланс на фирмата, Б-Брой потребители, С-брой артикули, Д-бутони за отчет, Е-бутони за внасяне и изтегляне на суми, F- Създаване на потребител, G- Редактиране на потребител , Н- изтриване на потребител, I- създаване на артикул, J- редактиране на артикул, К- изтриване на артикул

ГЛАВА 7: БЪДЕЩО РАЗВИТИЕ НА ПРИЛОЖЕНИЕТО

7.1. Опция за регистриране на потребители с помощта на баркод – с цел по-експедитивна работа с приложението, в бъдеще може да се избегне писането на потребителски имена и пароли и на тяхно място да се сканира карта с баркод, съдържащ вътре потребителското име и да се потвърждава идентичността с цифрен код.

7.2. Усъвършенстване на разпечатването на отчетите – с цел повече извличане на информация за състоянието на фирмата, може да се разпечатва хартиена разписка в която да се съдържа по-подробна информация за събитията през въведения период. Също така може да се изведе електронна разписка като служебен бон, който да показва колко голяма стойност бива служебно извадена или въведена.

7.3. Деактивиране на акаунти – в случай, че даден потребител, временно няма да извършва функциите си в програмата, да се въведе допълнителни правомощия, които да не позволяват да се влезе в неговия профил.

7.4. Добавяне на бутони за нерегистрирани артикули – ако даден продукт по време на създаване на поръчка, не бъде засечен в системата, вместо да се прекратява поръчката, да се влиза от администратор, да се създаде артикула и тогава да се влезе като потребител и да се сканира всичко на ново, да се създаде допълнителен фиксиран бутон, който чрез въвеждане на сума и натискането му, да се изпише в касовия бон вместо името на артикула, да се изпише „food item”, “tabasco” и т.н.

7.5. Употребата на преки пътища, чрез клавиатурата – използване на клавишни команди с цел по-бърза работа с приложението. Например, да се използва бутона Enter за въвеждане на артикул в касовия бон.

7.6. Актуализация на базата данни – използване на по-подробни данни в таблиците и употреба на повече релации.

ГЛАВА 8: ТЕСТОВЕ И ЗАКЛЮЧЕНИЕ

Поради причината, че тестовете са най-важния етап от разработката на едно приложение, изпълнението на този проект се извършват ръчно в различни работни среди с различни устройства с цел най-добре функционалност.

Програмата работи успешно на тестваните устройства след всички извършени тестове.

Тестван на настолен компютър със спецификации:

- Процесор: Intel Core i5-8400
- Видео карта: AMD Radeon 5700
- RAM: 16GB

Тестван на лаптоп със спецификации:

- Процесор: Intel Core i7-11370H
- Видео карта: NVIDIA GeForce RTX 3060
- RAM: 24GB

Тестван на лаптоп със спецификации:

- Процесор: Intel Core i7 9750H
- Видео карта: NVIDIA GeForce GTX 1650
- RAM: 8GB

Направените тестове доказват, че текущият проект предоставя готово приложение, което разполага със следните секции: „Влизане“, „Админ“ и „Служители“. Служителите имат възможност да създават касов бон, да отчитат продажби от артикули и коригират касов бон. Програмата съдържа административно меню, което може да създава, коригира и изтрива потребители и продукти и да преглежда дневен, месечен и годишен отчет.

ИЗПОЛЗВАНА ЛИТЕРАТУРА

- **Република България – Национален статистически институт -**
[https://wwwv.nsi.bg/bg/content/1_9914/npeCCb06ueW1e/V13nO"3BaHe-Ha-hH\\$OPMaLVIOHHh/-KOMYHVIKaUhOHH/-TeXHonorMM-B-aovaKVIHCTeaTa-M-OT-nøgaTa-npe3-2022-rOAVIHa](https://wwwv.nsi.bg/bg/content/1_9914/npeCCb06ueW1e/V13nO)
- **Безплатна книга „Въведение в програмирането C#“ -**
<https://www.introprogramming.info/wp-content/uploads/2015/10/Intro-CSsharp-Book-v2015.pdf>
- **Практикум по бази от данни – част 1 -**
https://www.researchgate.net/publication/303920226_Praktikum_po_bazi_ot_danni_-_I_cast
- **Microsoft Learn – SQL Server Management Studio -**
<https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16>
- **Wikipedia – Microsoft Visual Studio –**
https://bg.wikipedia.org/wiki/Microsoft_Visual_Studio
- **Microsoft Learn - Инструментът за поправка на Microsoft .NET Framework –**
https://support.microsoft.com/bg-bg/topic/MHapYMeHTET-3a-nonpaBG-Ha-micrasoft-net-framework-nna@opMa-e-HamweH-942a01_e3-5b8b-7abb-c166-c34a2f4b612a1