**Restore Lite**

1. Тема: **Restore Lite** – това е Windows Forms Application, предназначено за издаване на касови бележки, отчитане на дневен оборот, създаване и отчитане на продукти и отчитане на дейностите на служителите.
2. Автори:

* **Йордан Живков Йорданов**, имейл: dani\_yordani@abv.bg, училище: ППМГ

„Добри Чинтулов“, град Сливен

* **Валентин Антониев Хаджиминов**, имейл: valio1810@abv.bg, училище: ППМГ

„Добри Чинтулов“, град Сливен

1. Ръководител: **Димитър Минчев**
2. Резюме на проекта:
   1. Restore Lite е предназначен за управляване ресурсите на даден магазин и отчитане на фактури и продукти.
   2. Основните етапи в реализирането на нашия проект са:

* Създаване на база от данни
* Създаване на форми
* Валидация на формите
* Създаване на логиката по MVC модел.
* Оправяне на конвенциите за писане на код
* Писане на Unit Tests

След определяне на замисъла на проекта задачите са разпределени по следния начин:

**Йордан Йорданов** – участие в разработката на Main Form, разработка на бизнес логиката на приложението, участие в писането на базата данни.

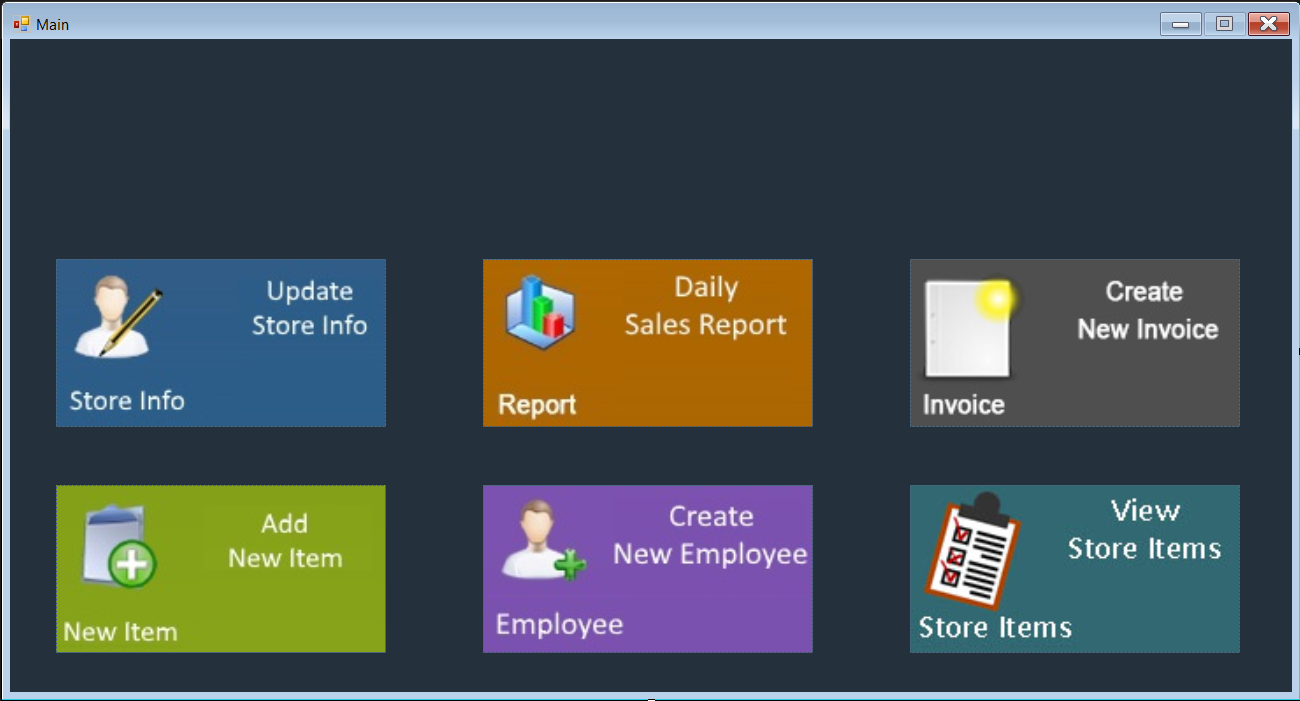
**Валентин Хаджиминов** – съставяне на дизайна на формите, участие в разработката на Main Form, участие в писането на базата данни;

* 1. Логическо и функционално описание: проектът е базиран и изграден на трислойния модел MVC, 3 основни модула: View – Frontend (дизайна на формите); Model – класове за обръщение към базата данни (Entity Framework); Controller – Backend (класове, имплементиращи бизнес логиката и обединяващи дейността на приложението). Взаимодействието между отделните елементи се обуславя на контролера, който е взаимовръзката между View и базата данни.

При влизане в приложението, потребителят има възможност да си направи регистрация и да промени информацията на магазина (Store Name, Store Address), да следи своя Daily Sales Report и да следи и да променя информацията на продуктите.

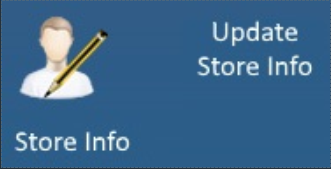
Реализация: MVC модела дават възможност да бъде разделен потребителския интерфейс от бизнес логиката в самостоятелно обособени модули, като едновременно с това дава възможност всички те да бъдат разработвани в една обща програмна среда Microsoft Visual Studio. Като програмен език е използван C#. NET Core Framework, който позволява приложението да бъде пуснато на различни платформи (операционни системи).

Всяка снимка е експортната като png с резолюция подходяща за екрана на потребителя. Работено е в RGB цветови профил, а цветовете във фронтенда са вкарани благодарение на HEX кода на всеки цвят. Използвани са няколко логота – едно за Daily Sales Report, и едно за главната форма.

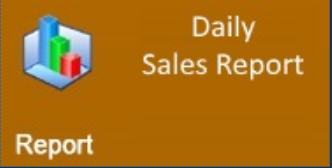
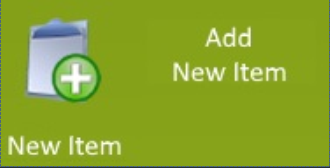


5.1. Главната форма ни препраща към всички второстепенни форми.

5.2. Този бутон ни препраща към Update Store Info формата. Тази форма има полета Store Name и Store Address. Чрез тези полета можем да обновим информацията за магазина.

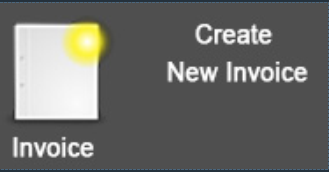


5.3. Този бутон ни препраща към Add New Item формата. Тази форма ни позволява да създадем и да добавим нов продукт в базата данни на магазина. Тази форма има полета Barcode, Product Name, Original Price, Sales Price и Quantity.



5.4. Този бутон препраща към Daily Sales Report формата. Тази форма ни позволява да следим нашия доход за деня и ни показва всички закупени от нашите клиенти артикули.

5.5. Този бутон препраща към Create New Employee формата. Чрез тази форма ние добавяме нов служител и запазваме неговите данни в базата данни.

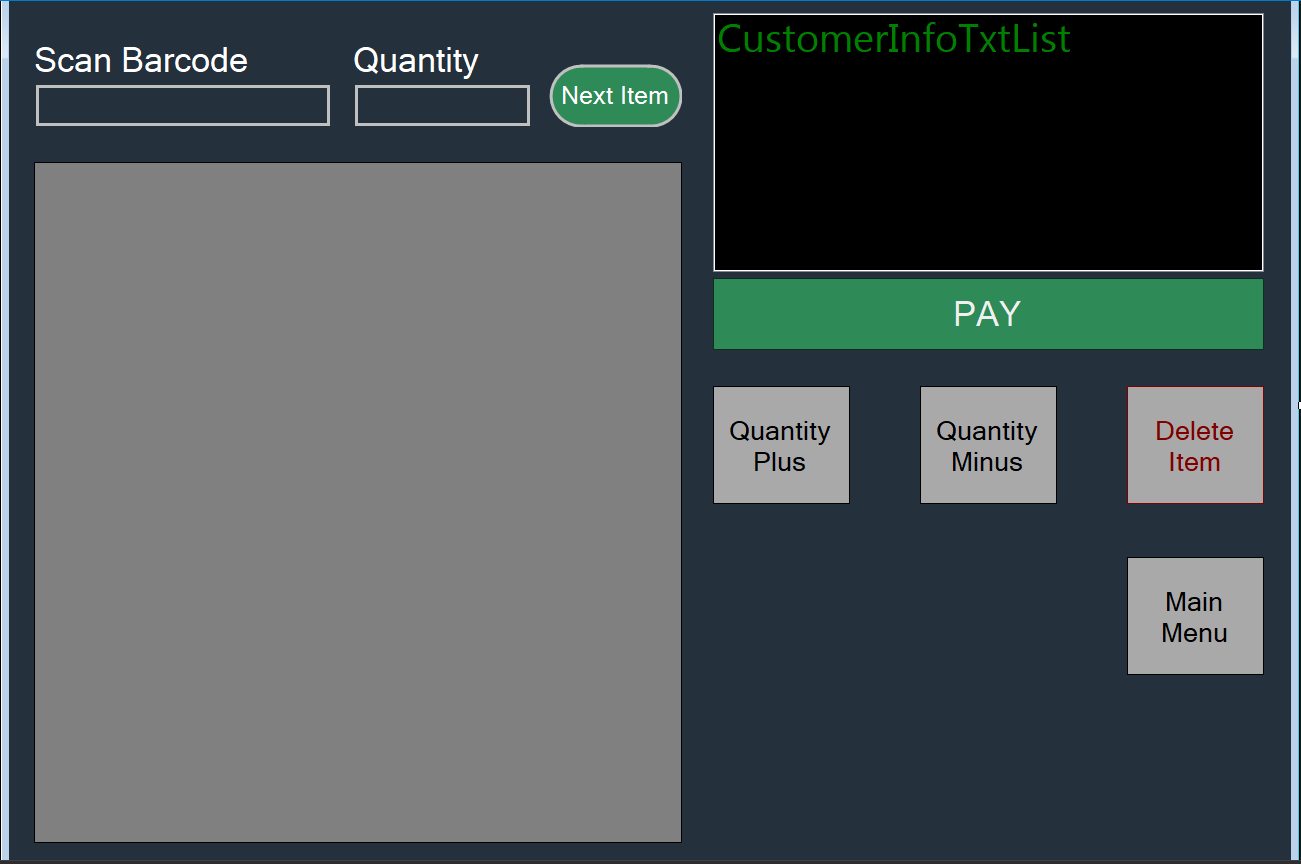


5.6. Този бутон препраща към Create New Invoice формата. Тази форма създава продуктите за една фактура и ги препраща към Pay формата.

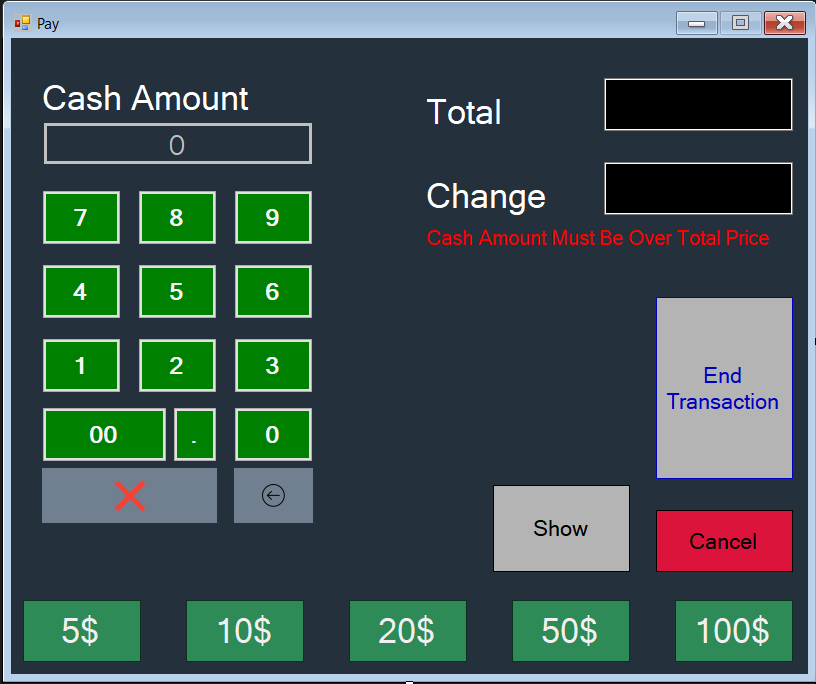


5.7. Този бутон препраща към View Store Items формата. Чрез тази форма можем да видим с какви артикули разполага магазина, да обновяваме информацията за тях или да ги изтриваме от базата данни.

6. Това е дизайнът на Invoice формата. Той се състои от следните елементи:



* две текстови полета, като в първото се въвежда баркода на желания от клиента артикул, а във второто – количеството, което клиента иска да закупи.
* DataGrid, който ни показва всички избрани от клиента артикули, с техния баркод, име на продукта, единична цена и крайна цена
* Next Item бутон, който добавя нов елемент в DataGrid-а и занулява текстовите полета за баркода и за количеството
* Pay бутон, който препраща към Pay формата
* Quantity Plus и Quantity Minus бутоните, които съответно увеличават или намалят количеството на селектирания от продукт, Delete Item бутон, който изтрива селектирания продукт и Main Menu бутон, който препраща към главната форма



7. Това е дизайнът на Pay формата. Тя се състои от следните елементи:

* Cash Amount текстово поле, в което клиентът въвежда сумата, която ще плати
* Total текстово поле, което показва общата сума за плащане
* Change текстово поле, което показва колко е рестото, което клиента ще получи
* End Transaction бутон, който приключва цялата транзакция
* Show бутон, който показва разписката, която ще бъде издадена на клиента
* Cancel бутон, който отказва плащането и затваря формата

Заключение: Продуктът има за цел да създава фактури и да показва информация за магазина. Опростения и лесен за разбиране дизайн го прави лесен за използване за всички служители. Съществуват много възможности за развитие, като популяризирането му в големи вериги магазини. Съществува възможност да бъдат предоставени настройки за шрифта, езика и резолюцията. Също така, проектът може да бъде популяризиран чрез социалните мрежи.