

Invoice Management



Creater: Parthiv Vekaria

Roll No: 23CED035

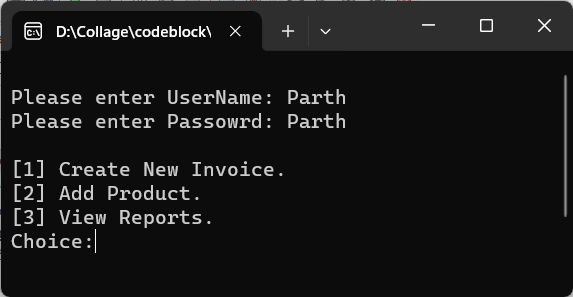
**INDEX**

|  |  |
| --- | --- |
|  | Pg. No. |
|  |  |
| Introduction | 2 |
| How to Create Invoice | 2 |
| How to Add/Update Product | 4 |
| How to View Report | 5 |
| Technical Information | 6 |
| Link Of Original Source Code | 10 |

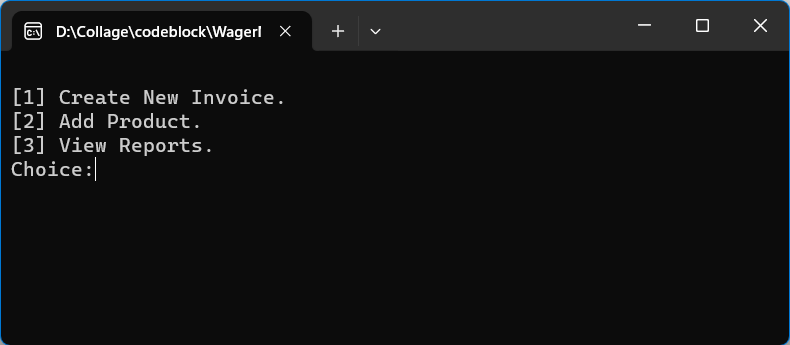
Purposer.

It provides support to manage Invoice at Point Of Sale (POS). It helps to manage data redundancy there it provides better space management, due to all Entity is stored separately and all the attributes are in atomic form in brief all the data is store in 3 Normalized Form without any introduction of structured database, using Directory Management technique. As data is stored digitally, it reduces the physical space required to store paper-based data and increase the confidentiality, who can access the data.

Introduction to WaGe Manager.

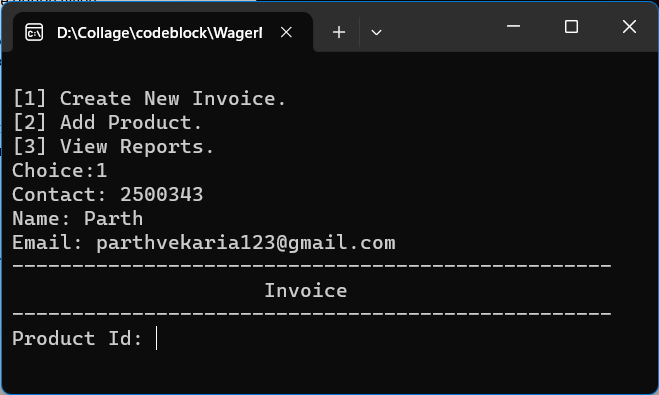
At first user will input User name and then password. After inputting the authentation info, if either of info is info is incorrect user want have access and it will mention which info is incorrect.

User will have “3” attempt to try before end of program.

The Fig shows the initial view of application after authentation. There are three option given:

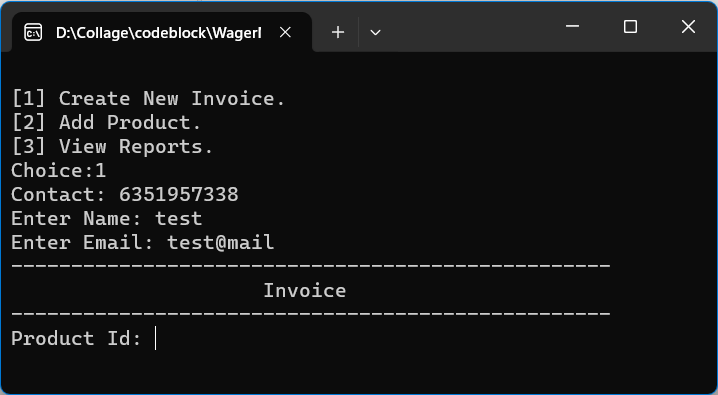
1. Create New Invoice.
2. Add Product.
3. View Report.

User is asked to choose one of the option by inputting the integer.

Create New Invoice

After choosing the first option it will ask user to input the contact which mandatory.

Scenario 1:

After Inputting the contact if the contact already exist in the system it will automatically list its Name and Email, then it will initiate to produce the invoice, where it will ask user to enter product code to identify the product.

Scenario 2:

After Inputting the contact if the contact does not exist it as the user to enter the name and Email, after inputting the details it will lead to initiate to create invoice.

Introduction to products in invoice.

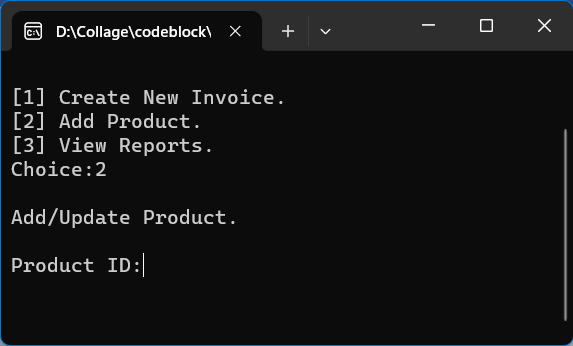
Next phase is to enter the product code which can be (barcode number, skew code or any code) that represent the product.

As per the figure the first Product ID: PLY is enter it outputs product details (Name, Price). Then user will be asked to enter Quantity required, next Subtotal is listed.

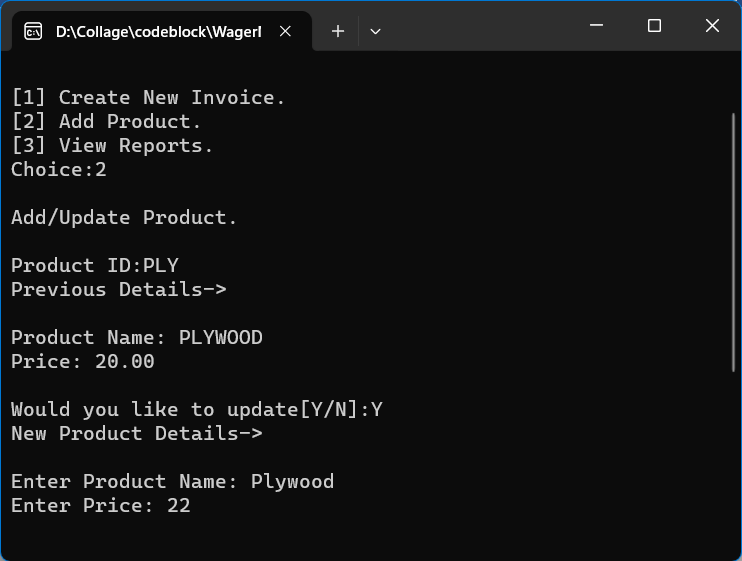
If Product code entered dose not exist in product list it will output “Product Not Found”.

After that user will be asked if it require to add more product if yes then “Y” in any case. If No then “N” or any char to terminate the invoice.

At the End of the invoice Total is given. The process of choosing option is initiated again.

Add Product

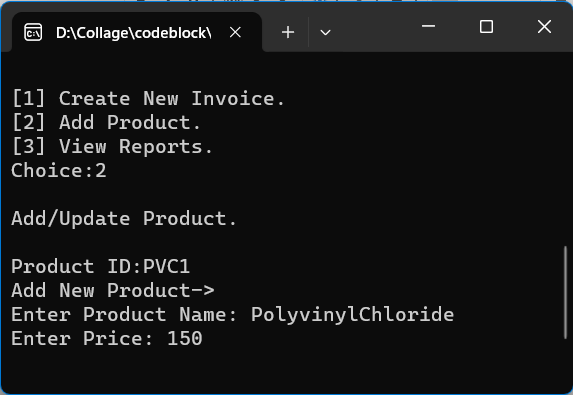
The Second chooise “Add Product”, the following screen on right will be displayed it will ask to Enter Product ID, To Update or Add New Product.

 Scenario 1: Existing ID

If user inputs the existing Product Id, It will display existing Product Info and it will be asked if User want to update product info if yes(Y) in any case.

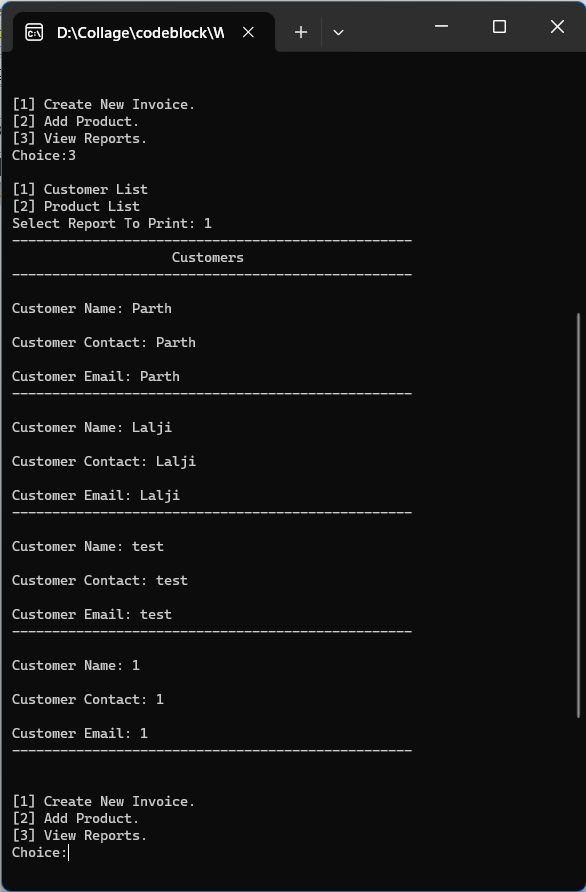
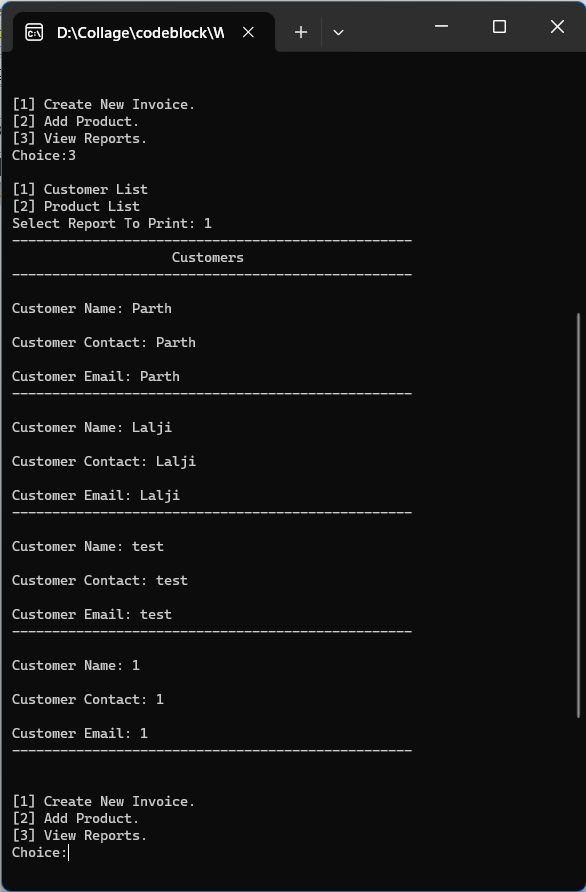
Entre Product Name, Product Price. On Enter product info will be updated.

If No(N) or anyother char it will back to Choices.

Scenario 2: New Product ID

If enetred product Id dose not exist it will ask to enetr product name and product price. The new product is created in database.

show report



The Third chooise “View Report”, the following screen on right will be displayed it will ask to select report waqnt to View.

Currently Existing Report:

* Customers Info
* Products Info

**Technical Info**

There are list of object used in order to create mentioned product:

1. User define Function
2. Structure
3. Pointer
4. Custom Preprocessor directory file
5. File Handling
6. Directory Management
7. Cryptography (Custom) Designee.

Custom Preprocessor directory file

The Brief info to each Module.

1. Product

There are there User define function to Save Product, Get Product   
Information (Product Name, Price) and to check if product exist.

* **SaveProduct** Function take “ProductInvo” structure by reference for better memory management. And create a text File Name of Product id, since it’s unique attribute (Primary Key) to the Product Entity.
* A Setter Procedure Name **GetProductData** Is take “ProductInvo” structure parameter as reference and based on its Product ID Other Detail Of The Product is updated in pointer structure.
* Third Function ProductExist Take Array Of char (Product ID), the function returns “1” if Product (File name of Product ID) Exist else return “0”.

1. Customer

It include functions, to check customer existence, Get customer data, Save Customer Data. Same Operation performed as Product Module just it uses the “Customer” Structure and the Primary key in this case is Contact.

1. CustomerInvoice

A module to that include functions to create Invoice, add product in invoice, get Invoice Number.

* **createInvoice** it takes invoice number and customer contact as primary key and foreign key respectively. The file name is concatenated as InvoiceNumber\_CustomerContact.txt and created then the file name is retuned at end Before that Invoice number is Updated.
* **AddProductToInvoice** it take Filename and product Structures pointer it saves that product info in given file.
* **getInvoiceNum** it is a function that open a “Invoice.ini” file and read new invoice number and returns.

1. Report

Currently it include only two function(report) Customer List, Product List.

1. Format

It helps to alligen text in heading automatical. It include three Function, Border, Margin, Title.

* Function **Border** takes two parameter the length and Char, means length takes number of time the charater want to print in single line.
* **Margin** it takes just length and print the only blank space need in given line.
* **Title** it takes string as charater array of title, style as char of border, then length of border as integer (it should be at list greater than the length than Title To alligne Title other wise title will display in left aligned ), then it take integer of required margine title then, it take an enumerated data type “r”, “l” and “c” which means Right, Left, Center aligment of title.

1. LoginSys

It has four functions, to create user, to check user, to check password, Encrypt Password(Custom Encryption Method is used). This Encryption method is **ONE WAY ENCRYPTION** means given text is converted in to a Encrypted text that cannot be convert back to normal text (one Way Function), it can only be compared. Hence it can only be cracked using Brute Force Attack \Dictionary attack to find original text. The Pseudocode written on Next Line.

Algorithms of **LoginSys.c** File in Pseudocode:

FUNCTION INT \* EncryptPassword(STRING Password)

DECLARE INT strLength = strlen(Password)

DECLARE INT EncryptedPassword[strLength +1]

DECLARE INT \*Pointer

DECLRE I

FOR(I=0; \*(Password + i) != ‘\0’; i++)

EncryptedPassword[i] = Password[i] + strLength

END FOR

EncryptedPassword[i] = -1;

Pointer = &EncryptedPassword;

END FUNCTION

PROSEDURE createUser(STRING USERNAME,STRING PASSWORD)

Fp =OpenFile(“relative\_Data{ USERNAME }.txt”, “W”)

FOR (i=0; \*( EncryptPassword(PASSWORD)+1) != -1; i++)

WRITE(Fp , \*(EncryptPassword(pass) + i))

END FOR

CLOSEFILE(Fp)

END PROSEDURE

FUNCTION INT checkUser(char User[])

Fp = OpenFile(“relative\_Data{ User }.txt”, “R”)

Is\_exist = 0

IF (Fp != NULL)

Is\_exist = 1

End IF

CLOSEFILE(Fp)

RETURN Is\_exist

END FUNCTION

FUNCTION INT checkPassword(CHAR User[], CHAR pass[])

CHAR IntStrPass[100];

CHAR temp[5];

CHAR FileData[100]

Fp = OpenFile(“relative\_Data{ User }.txt”, “R”)

READ(FileData, Fp)

CLOSEFILE(Fp)

FOR(I=0; \*(EncryptPassword(pass) + i) != -1; i++)

NUM\_TO\_STR (temp, \*(EncryptPassword(pass) + i))

CONCATINATE(IntStrPass, temp)

END FOR

IF ( filePath = IntStrPass)

THEN

RETURN 1

ELSE

RETURN 0

END IF

END FUNCTION

Project Link on Github

<https://github.com/Parthiv2749/C_Project_fstYear/tree/main/WagerManager>