Introduction:

This project for cash and carry management system handles various features that help the management procedure in the system replacing all the manual methods. Overall, a digital management system helps to fasten the processes and minimize all the errors, so here we are implementing this system to learn and to understand the backend working and efforts of the developers.

This project was interesting since it has eliminated the challenges faced by cash and carry businesses in managing their operations manually, although our program is very much less developed as compared to their management system but it was important to us as it gave us an overview of building a program by keeping in view the ease of customer.

The project encompasses key functionalities such as Storage management, Sales management, customer management, billing and invoicing, and Administration management. These features enable the users to effectively monitor stock levels, track sales and profits, manage customer information, generate invoices.

Overall, by all the features we added this project can handle a vast area of cash and carry business resulting in optimization of their management processes, improve productivity, and achieve sustainable growth. It offers a user-friendly solution that simplifies complex tasks and improves operational efficiency.

Features and Functionality:

The cash and carry management project include a program solution that is designed to streamline and optimize the operations of cash and carry businesses. This project encompasses various key features, including administration management, sales management, and billing and invoicing. By integrating these features into a single platform, the program will be able to enhance efficiency, accuracy, and productivity for cash and carry businesses.

Administration Management:

The administration management feature provides comprehensive tools for effectively managing various administrative tasks within the cash and carry business. It includes functionalities such as employee management, user roles and permissions, inventory management, and supplier management. With this feature, administrators can easily assign roles and responsibilities to employees, monitor inventory levels, and efficiently manage supplier relationships. This ensures smooth day-to-day operations and effective coordination within the business.

- **Editing**

The editing feature includes the features such as adding the product removing the product and increasing or decreasing quantities. Thus, our program is able to manage the products so that the information ma be easily tracked regarding sales and storage and this feature also enables the user to manage the products according to the inflow of new products on daily basis.

- **Sales Management**

The sales management feature enables cash and carry businesses to efficiently handle their sales processes and check out profits. It includes functionalities such as separating daily and monthly sales and profit tracking. Through this feature, our program can easily manage Sales information, track sales orders, generate sales reports, and analyze sales performance. It provides a centralized system to streamline the entire sales workflow, ensuring accurate and timely order processing and improving customer satisfaction.

Storage Management:

Storage management is a critical aspect of cash and carry management systems. It involves efficiently managing and organizing inventory and stock within the cash and carry store. The main goal of storage management is to ensure optimal utilization of available space, easy accessibility of products, accurate tracking of stock levels, and timely replenishment. So, the program automatically removes the products that are sold so that the stock can be easily managed and when next time when another customer shows up he/she can view the updated stock, it is necessary in case a product is finished the customer may know.

Customer's Purchase:

This program examines the shopping of a customer and adds it up so that the cashier and the customer can tally the products in case the customer has forget to buy something, this program then adds up all the products of different quantities and calculates the total amount and displays a separate amount that includes the GST tax, in case of over payment the program suggests the required cash back to the customer.

Billing and Invoicing:

The billing and invoicing feature simplifies the process of generating invoices and managing billing activities. We have put in a feature of visa payment along with cash payments option, the visa option allows the user to use virtual money and easily buy stuff in case of unavailability of cash. It allows businesses to create professional invoices and track payments. This feature automates the billing process, reducing manual errors and saving time.

Conclusion

By incorporating these features into the cash and carry management system, the businesses can benefit from enhanced operational efficiency, improved sales tracking and analysis, and streamlined billing processes. The project offers a user-friendly

interface, making it easy for users to navigate and utilize these features effectively. Overall, this project serves as a valuable tool for cash and carry businesses, empowering them to manage their administration, sales, and billing activities with ease and efficiency.

For a business, profits is really much concerning. By this type of system man power can be replaced resulting in more saving of the company and as a by product all the operational activities will be enhanced and the management will be done in a faster way.

Code Snippets:

The above diagram show a part of code for displaying the front menu. Also, it is the beginning of the program.

```
public static void security(){
     try{
        while (true){
            Scanner input = new Scanner(System.in);
            System.out.print("\t\tEnter ID: ");
            String x = input.nextLine();
            System.out.print("\t\tEnter Password: ");
            String y = input.nextLine();

            if (x.matches(id) && y.matches(password)){
                administration();
                break;
            }
            else if (x.matches(id) && !y.matches(password)){
                System.out.println("\t\tWrong Password. Enter Again.");
            }
            else if (!x.matches(id) && y.matches(password)) {
                System.out.println("\t\tWrong ID. Enter Again.");
            }else{
                System.out.println("\t\tWrong Id and Password. Enter Again.");
            }
        }
    }catch(Exception ex){
        security();
    }

}
```

The above pic show the code for ID and Password which are required to access the Administration area which include different operations such as adding or modifying product or checking sales etc.

```java
public static void sales(){
    while(true){
        try{
            System.out.println("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Sale ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
            Scanner input = new Scanner(System.in);
            System.out.println("\t\tEnter 1- to check Daily sale: ");
            System.out.println("\t\tEnter 2- to check monthly sale: ");
            System.out.println("\t\tEnter 0- to go back: ");
            System.out.print("\t\tWhat you want to do: ");
            int choice = input.nextInt();
            if (choice == 1) {
                daily();
            }else if (choice == 2) {
                monthly();
            }else if (choice == 0) {
                break;
            }
        }catch (Exception ex){
            System.out.println("\t\tInvalid input. Try Again.");
        }
    }
}
```

This pic show the code for accessing the sales area that include different information including the information of the customers.

```java
public static void administration(){
    while(true){
        try{
            Scanner input = new Scanner(System.in);
            System.out.println("\n------------------------------------------ Administration ----------------------------------------");
            System.out.println("\t\tEnter 1- To Modify Products.");
            System.out.println("\t\tEnter 2- To Check Sale.");
            System.out.println("\t\tEnter 3- To change ID or Password.");
            System.out.println("\t\tEnter 4- To Display Stoke.");
            System.out.println("\t\tEnter 5- To Go to Main Menu.");
            System.out.printf("\t\tEnter what you want to do: ");
            int choice = input.nextInt();
            if (choice == 1) {
                modify();
            }
            else if (choice == 2) {
                sales();
            }
            else if (choice == 3) {
                input = new Scanner(System.in);
                System.out.print("\t\tEnter New ID: ");
                String newID = input.nextLine();
                String newPass;
                while(true){
                    input = new Scanner(System.in);
                    System.out.print("\t\tEnter New Password: ");
                    newPass = input.nextLine();
                    if (newPass.matches(password))
                        System.out.println("\t\tNew password cannot be same as previous one.");
                    else{
                        System.out.println("\t\tPassword updated. Successfully.");
                        break;
                    }
                }
                id = newID;
                password = newPass;
            }
            else if (choice == 4) {
                displayMenu();
            }
            else if (choice == 5) {
                break;
            }
            else
                System.out.println("\t\tWrong input. Enter Again.");

        }catch (Exception ex){
            System.out.println("\t\tWrong input. Enter Again.");
        }
    }

}
```

The above pic show the code for administration menu and also for changing the ID and Password for accessing the Administration area. If we select option 1 it allow us to modify products. The 2 options allow us to check the sales. The 3 options allow

us to change the login where 4 allow us to see the stock of products. If we press 5 we return to main menu.

```
public static void daily(){
    try {
        System.out.println("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Daily Sale ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
        File path = new File("D:\\Study Material\\Programming Fundamentals\\Assignments\\Project\\sales.csv");
        BufferedReader reader = new BufferedReader(new FileReader(path));
        int rows = 0;
        String line;
        while ((line = reader.readLine()) != null) {
            rows++;
        }
        reader.close();

        reader = new BufferedReader(new FileReader(path));

        int i = 0;
        String[][] content = new String[rows][6];
        while ((line = reader.readLine()) != null) {
            content[i] = line.split(",");
            i++;
        }
        reader.close();

        int date, month, year;
        while(true){
            try{
                Scanner input = new Scanner(System.in);
                input = new Scanner(System.in);
                System.out.print("\t\tEnter date: ");
                date = input.nextInt();
                System.out.print("\t\tEnter Month: ");
                month = input.nextInt();
                System.out.print("\t\tEnter year: ");
                year = input.nextInt();
                if (checkDate(date, month, year)) {
                    break;
                }else
                    System.out.println("\t\tWrong Date format. Enter Again");
            }catch(Exception ex){
            System.out.println("\t\tWrong date fomat. Integer type only.");
            }
        }
        long totalSale = 0;
        for (i = 0;i < rows ;i++ ) {
            if (content[i][4].matches(String.valueOf(year))) {
                if (content[i][3].matches(String.valueOf(month))) {
                    if (content[i][2].matches(String.valueOf(date))) {
                        totalSale = totalSale + Integer.parseInt(content[i][5]);
                    }
                }
            }
        }
        System.out.println("\t\tTotal Daily sale on " + date + "-" + month + "-" + year + " is " + totalSale + "Rs");
        System.out.println();
    } catch(Exception ex){
        System.out.println("\t\tInvalid input. Try Again." + ex.getMessage());
    }
}
```

The above pic shows the code for daily sales. In this we have specified the path where we have created a file in which the sales will be saved. Also, it has the code for checking the validity of the date. Also we have used the try catch formula to stop the program from crashing and also so that a user doesn't enter an invalid input.

```
public static void monthly(){
    try {
        System.out.println("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Monthly Sale ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
        File path = new File("D:\\Study Material\\Programming Fundamentals\\Assignments\\Project\\sales.csv");
        BufferedReader reader = new BufferedReader(new FileReader(path));
        int rows = 0;
        String line;
        while ((line = reader.readLine()) != null) {
            rows++;
        }
        reader.close();

        reader = new BufferedReader(new FileReader(path));

        int i = 0;
        String[][] content = new String[rows][6];
        while ((line = reader.readLine()) != null) {
            content[i] = line.split(",");
            i++;
        }
        reader.close();
        int date = 1;
        int month, year;
        while(true){
            try{
                Scanner input = new Scanner(System.in);
                input = new Scanner(System.in);
                System.out.print("\t\tEnter Month: ");
                month = input.nextInt();
                System.out.print("\t\tEnter year: ");
                year = input.nextInt();
                if (checkDate(date, month, year)) {
                    break;
                }else
                    System.out.println("\t\tWrong Date format. Enter Again");
            }catch(Exception ex){
            System.out.println("\t\tWrong date fomat. Integer type only.");
            }
        }
        long totalSale = 0;
        for (i = 0;i < rows ;i++ ) {
            if (content[i][4].matches(String.valueOf(year))) {
                if (content[i][3].matches(String.valueOf(month))) {
                        totalSale = totalSale + Integer.parseInt(content[i][5]);
                }
            }
        }
        System.out.println("\t\tTotal monthly sale on " + month + "-" + year + " is " + totalSale + "Rs");
        System.out.println();
    } catch(Exception ex){
        System.out.println("\t\tInvalid input. Try Again." + ex.getMessage());
    }
}
```

As defined above that was for daily sales and this code is for monthly sales storing. In this scenario we check for valid date then store the record so that the record is valid and is useful in future.

```java
public static void customer(){
    Scanner input = new Scanner(System.in);
    String cnic = "";

    try {
        System.out.print("\t\tEnter your name: ");
        String name = input.nextLine();

        String phone;
        while (true) {
            System.out.print("\t\tEnter your phone number (without dash): +92");
            phone = input.nextLine();
            if (phone.matches("\\d{10}")) {
                if (phone.charAt(0) == '3') {
                    break;
                } else {
                    System.out.println("\t\tInvalid phone number.");
                }
            } else {
                System.out.println("\t\tInvalid phone number.");
            }
        }

        while (true) {
            System.out.print("\t\tEnter your CNIC (xxxxx-xxxxxxx-x): ");
            cnic = input.nextLine();
            if (cnic.matches("\\d{5}-\\d{7}-\\d{1}")) {
                break;
            } else {
                System.out.println("\t\tInvalid CNIC. Try again");
            }
        }
        int date, month, year;
        while(true){
            try{
                input = new Scanner(System.in);
                System.out.print("\t\tEnter date: ");
                date = input.nextInt();
                System.out.print("\t\tEnter Month: ");
                month = input.nextInt();
                System.out.print("\t\tEnter year: ");
                year = input.nextInt();
                if (checkDate(date, month, year)) {
                    break;
                }else
                    System.out.println("\t\tWrong Date format. Enter Again");
            }catch(Exception ex){
            System.out.println("\t\tWrong date fomat. Integer type only.");
            }
        }
        System.out.println("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Welcome To Our Menu ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

        File path = new File("D:\\Study Material\\Programming Fundamentals\\Assignments\\Project\\Data.csv");
        Scanner read = new Scanner(path);
        int row = 0;
        while (read.hasNextLine()) {
            String line = read.nextLine();
            row++;
        }
        String[] content = new String[row];

        int menuSize = 0;

        read = new Scanner(path);

        int productNumber = 0;

        while (read.hasNextLine()) {
            String line = read.nextLine();
            content[menuSize] = line;
            menuSize++;
```

```
        String[] itemData = line.split(",");

        for (int i = 0; i < itemData.length; i++) {
            System.out.printf("\t%-16s", itemData[i]);
        }
        System.out.println();

        productNumber++;
    }

    long bill = 0;

    String[][] purchasedProducts = new String[row][3];
    int purchasedProductsCount = 0;

    while (true) {
        try {
            System.out.print("\t\tEnter product number and 0 to exit: ");
            int x = input.nextInt();
            if (x == 0) {
                break;
            } else {
                if (x < 1 || x >= productNumber) {
                    System.out.println("\t\tInvalid product number.");
                    continue;
                }
                String[] selectedProduct = content[x].split(",");
                System.out.println("\t\tProduct Name is: " + selectedProduct[1]);
                if (isOutOfStock(selectedProduct)) {
                    System.out.println("\t\tOut of Stock");
                    continue;
                }
                while (true) {
                    System.out.print("\t\tEnter total number of quantities you want: ");
                    int y = input.nextInt();
                    if (y > Integer.parseInt(selectedProduct[3])) {
                        System.out.println("\t\tWe don't have that much in stock.");
                    } else if (y < 1) {
                        System.out.println("\t\tInvalid quantity.");
                    } else {
                        int price = Integer.parseInt(selectedProduct[2]);
                        bill = bill + (price * y);

                        String[] purchasedProduct = new String[3];
                        purchasedProduct[0] = selectedProduct[1];
                        purchasedProduct[1] = String.valueOf(y);
                        purchasedProduct[2] = selectedProduct[2];

                        purchasedProducts[purchasedProductsCount] = purchasedProduct;
                        purchasedProductsCount++;

                        int availableQuantity = Integer.parseInt(selectedProduct[3]);
                        int updatedQuantity = availableQuantity - y;
                        selectedProduct[3] = String.valueOf(updatedQuantity);

                        content[x] = String.join(",", selectedProduct);

                        break;
                    }
                }
            }
        } catch (Exception ex) {
            System.out.println("\t\tInvalid choice. Try again.");
            input.next();
        }
    }

    System.out.println("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Order Details ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
    System.out.println("\t\tName: " + name);
    System.out.println("\t\tPhone: +92" + phone);
    System.out.println("\t\tCNIC: " + cnic);
```

```
System.out.println("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Purchased Products ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
for (int i = 0; i < purchasedProductsCount; i++) {
    String[] productData = purchasedProducts[i];
    System.out.println("\t\tProduct Name: " + productData[0]);
    System.out.println("\t\tQuantity: " + productData[1]);
    System.out.println("\t\tPrice per unit: " + productData[2]);
}

System.out.println("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Total Bill ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
System.out.println("\t\tTotal bill: " + bill);

// Payment logic
boolean paymentCompleted = false;
while (!paymentCompleted) {
    try {
        System.out.print("\t\tPress 1 for Visa card or 2 for cash payment: ");
        int payment = input.nextInt();
        if (payment == 1) {

            boolean validCardEntered = false;


            while (!validCardEntered) {
                Scanner in = new Scanner(System.in);
                System.out.print("\t\tEnter your 16-digit visa card number (xxxx-xxxx-xxxx-xxxx): ");
                String card = in.nextLine();

                if (card.matches("\\d{4}-\\d{4}-\\d{4}-\\d{4}")) {
                    while (true) {
                        System.out.print("\t\tEnter your 4-digit PIN: ");
                        String pass = in.nextLine();

                        if (pass.matches("\\d{4}")) {
                            System.out.println("\t\tPayment completed.");
                            transfer(name, cnic, date, month, year, bill);
                            validCardEntered = true;
                            paymentCompleted = true;
                            break;
                        } else {
                            System.out.println("\t\tWrong password. Enter again.");
                        }
                    }
                } else {
                    System.out.println("\t\tWrong visa card format.");

                }
            }
        } else if (payment == 2) {
            System.out.print("\t\tEnter total cash given: ");
            if (input.hasNextDouble()) {
                double cashGiven = input.nextDouble();
                System.out.println("\t\tCash Deposit is: " + cashGiven);
                System.out.println("\t\tTotal bill is: " + bill);
                System.out.println("\t\tCash change is: " + (cashGiven - bill));
                System.out.println("\t\tPayment completed");
                transfer(name, cnic, date, month, year, bill);
                break;
            } else {
                System.out.println("\t\tInvalid input. Please enter a valid cash amount.");
                input.next();
            }
        } else {
            System.out.println("\t\tInvalid choice. Try again.");
        }
    } catch (InputMismatchException ex) {
        System.out.println("\t\tInvalid choice. Try again.");
        input.next();
    }
}
```

```
            try {
                FileWriter writer = new FileWriter(path);
                for (int i = 0; i < content.length; i++) {
                    writer.write(content[i] + "\n");
                }
                writer.close();
            } catch (IOException e) {
                System.out.println("\t\tAn error occurred while updating the file: " + e.getMessage());
            }

        } catch (Exception e) {
            System.out.println("\t\tAn error occurred: " + e.getMessage());
        }

    }
```

The above defined snips are of customer module. In this we have first asked the name of the customer followed by his phone number and CNIC. Then the customer is allowed to purchase the product after he is finished with shopping the bill is generated along with GST tax. Then the customer is asked for payment method that whether he wants to pay by Visa card or cash. In any case the payment is accepted and the bill is provided to the customer.

```
public static void displayMenu(){
    try{
        File path = new File("D:\\Study Material\\Programming Fundamentals\\Assignments\\Project\\Data.csv");
        Scanner read = new Scanner(path);

        while(read.hasNextLine()){
            String[] line = read.nextLine().split(",");
            for (String k: line)
                System.out.printf("\t%-16s", k);
            System.out.println();
        }
        read.close();

    }
    catch(Exception ex){
        System.out.println(ex.getMessage());
    }
}
```

This above code is for the displaying of the stock menu which include the product number , product name , its price and its quantity which help the user to see thye product he/she is looking for is there or not.

```java
public static void modify(){
    while(true){
        try{
            Scanner in = new Scanner(System.in);
            System.out.println("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Modify Products ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
            System.out.println("\t\tEnter 1- To Add New Products.");
            System.out.println("\t\tEnter 2- To Delete Products.");
            System.out.println("\t\tEnter 3- To Change in Existing Products.");
            System.out.println("\t\tEnter 4- To Go back to Administration.");
            System.out.printf("\t\tWhat do you want to do: ");
            int choice= in.nextInt();
            if (choice==1){
                addProduct();
            }
            else if (choice==2){
                deleteProduct();
            }
            else if (choice==3){
                changeProduct();
            }
            else if (choice==4) {
                administration();
                break;
            }
            else{
                System.out.println("\t\tinvalid Input.");
            }
        }catch (Exception ex){
            System.out.println("\t\tWrong input. Enter Again");
            modify();
        }
    }
}
```

The above code is for modifying a product in administration area. The different options are given for the user from which he can easily perform this task.

```java
public static void addProduct(){
    try{
        Scanner in = new Scanner(System.in);
        File path = new File("D:\\Study Material\\Programming Fundamentals\\Assignments\\Project\\Data.csv");
        Scanner read = new Scanner(path);
        String[] line = read.nextLine().split(",");
        System.out.println("\t\tEnter new values according to the given column seperated by a comma:");
        for (String k : line)
            System.out.printf("\t\t%-15s", k);
        System.out.println();
        System.out.print("\t\t");
        String addition = in.nextLine();
        String[] value = addition.split(",");
        boolean productExist = false;
        while(read.hasNextLine()){
            String[] rowData = read.nextLine().split(",");
            if (rowData[0].equals(value[0])) {
                productExist = true;
                break;
            }
        }
        if (productExist) {
            System.out.println("\t\tProduct Number already exist. Enter another.");
            addProduct();
        }else{
            BufferedWriter write = new BufferedWriter(new FileWriter(path, true));
            write.write(addition);
            write.newLine();
            write.close();
            System.out.println("\t\tProduct Added Successfully.");
        }

    }catch(Exception ex){
        System.out.println(ex.getMessage());
        addProduct();
    }
}
```

Similarly, this code is for adding the product in the stock. If that product already exist, the message will be provided elsewise the product will be added to the stock.

```java
public static void deleteProduct() {
    try {
        File path = new File("D:\\Study Material\\Programming Fundamentals\\Assignments\\Project\\Data.csv");
        BufferedReader reader = new BufferedReader(new FileReader(path));
        int rows = 0;
        String line;
        while ((line = reader.readLine()) != null) {
            rows++;
        }
        reader.close();

        reader = new BufferedReader(new FileReader(path));

        int i = 0;
        String[][] content = new String[rows][4];
        while ((line = reader.readLine()) != null) {
            content[i] = line.split(",");
            i++;
        }
        reader.close();

        for (i = 0; i < content.length; i++) {
            for (int j = 0; j < content[i].length; j++) {
                System.out.printf("\t%-16s", content[i][j]);
            }
            System.out.println();
        }
        while(true){
            try{
                System.out.print("\t\tEnter product number you want to delete: ");
                Scanner in = new Scanner(System.in);
                int delNum = in.nextInt();
                BufferedWriter writer = new BufferedWriter(new FileWriter(path));
                for (i = 0; i < content.length; i++) {
                    if (content[i][0].equals(String.valueOf(delNum))) {
                        continue;
                    }
                    for (int j = 0; j < content[i].length; j++) {
                        writer.write(content[i][j] + ",");
                    }
                    writer.newLine();
                }
                System.out.println("\t\tProduct deleted successfully.");
                writer.close();
                break;
            }catch(Exception ex){
                System.out.println("\t\tWrong Product Number. Try Again.");

            }
        }
    } catch (Exception ex) {
        System.out.println("\t\tWrong input. Try Again.");
        deleteProduct();
    }
}
```

This code allow us to delete a product from storage or stock. It is useful if there is a product that is expired, and we want to remove it. Or if a product is in abundance that it is not selling but has taken up the storage so that we can delete it and make some room for the new product. This program ask for the product number that you want to delete than it goes to the stock file and remove the content under that product number.

```
public static void changeProduct(){
    try{
        String path = "D:\\Study Material\\Programming Fundamentals\\Assignments\\Project\\Data.csv";
        BufferedReader reader = new BufferedReader(new FileReader(path));
        int rows = 0, columns = 0;
        String line;
        while((line = reader.readLine()) != null)
            rows++;
        reader.close();
        reader = new BufferedReader(new FileReader(path));
        String firstLine;
        if ((firstLine = reader.readLine()) !=  null)
            columns = firstLine.split(",").length;
        reader.close();

        reader = new BufferedReader(new FileReader(path));
        String[][] content = new String[rows][columns];
        int i = 0;
        while((line = reader.readLine()) != null){
            content[i] = line.split(",");
            i++;
        }

        for (int j = 0;j < content.length; j++){
            for (int k = 0;k < 2 ;k++ ) {
                System.out.printf("\t\t%-15s",content[j][k]);
            }
            System.out.println();
        }
        Scanner input = new Scanner(System.in);
        System.out.print("\t\tEnter product number you want to change: ");
        int editRow = input.nextInt();
        int x = 0;



        while ((line = reader.readLine()) != null) {
            content[x] = line.split(",");
            x++;
        }
        reader.close();

        for (i = 0; i < content.length; i++) {
            if (content[i][0].equals(String.valueOf(editRow))) {
                break;
            }
        }
        int rowNum = i;


        for (int j = 0; j < content[0].length; j++)
            System.out.println("\t\t" + (j + 1) + ": " + content[0][j] + "  ");

        System.out.println();

        System.out.print("\t\tWhich column do you want to edit: ");
        int editCol = input.nextInt() - 1;

        System.out.println("\t\tOld " + content[0][editCol] + " is: " + content[rowNum][editCol]);
        System.out.print("\t\tEnter new " + content[0][editCol] + ": ");
        input = new Scanner( System.in);
        String newValue = input.nextLine();
        content[rowNum][editCol] = newValue;

        BufferedWriter writer = new BufferedWriter(new FileWriter(path));

        for (int j = 0; j < content.length; j++){
            for (int k = 0; k < content[0].length; k++){
                writer.write(content[j][k] + ",");
            }
            writer.newLine();
```

```
            }
            writer.close();
            System.out.println("\t\tSuccessfully Changed.\n\t\tEnter 0 to edit again.\n\t\tEnter any other key to go back. "
            System.out.print("\t\t");
            String choice = input.next();
            if (choice.equals("0"))
                changeProduct();
            else
                modify();

        }
        catch(Exception ex){
            System.out.println("\t\tInvalid input. Enter again.");
            System.out.println("\t\t" + ex.getMessage());
        }
    }
```

This code help us to change the information of the product. Such as if we want to change the price or name of a product we can use this facility.

```
    private static boolean isOutOfStock(String[] productData) {
        int quantity = Integer.parseInt(productData[3]);
        return quantity == 0;
    }


    public static boolean checkDate(int day, int month, int year){
        int maxdays;
        if (month == 2) {
            if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
                maxdays = 29;
            else
                maxdays = 28;
        }
        else if (month == 4 || month == 6 || month == 9 || month == 11)
            maxdays = 30;
        else
            maxdays = 31;

        if ((day >= 1 && day <= maxdays) && (month >= 1 && month <= 12) && (year > 1))
            return true;
        else
            return false;
    }

    public static void transfer(String name, String cnic, int date, int month, int year, long bill){
        try{
            String path = "D:\\Study Material\\Programming Fundamentals\\Assignments\\Project\\sales.csv";
            BufferedWriter writer = new BufferedWriter(new FileWriter(path, true));
            writer.write(name + "," + cnic + "," + date + "," + month + "," + year + "," + bill);
            writer.newLine();
            writer.close();
        }catch(Exception ex){
            System.out.println("\t\t Error occured.");
        }
    }
}
```

The above code is for checking a product whether it is in stock or out of stock. The second module is for checking the validity of the date that whether the date entered anywhere in the program is valid if it is invalid it provides with the invalid message so that one can enter the correct date.

Output:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~ Programing Fundamental ~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~ Final Project BDS 2A ~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~ Muhammad Saad Ali (FA22-BDS-030) ~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~ Ahmed Nawaz Abbasi (FA22-BDS-005) ~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~ Syed Mohammad Kumeyal Rizvi (FA22-BDS-039) ~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~ Muhammad Zeeshan (FA22-BDS-032) ~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~ Muhammad Nouman (FA22-BDS-028) ~~~~~~~~~~~~~~~~~~~~
~~~~ Submitted To: ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Sir Rizwan Rashid ~~~~

~~~~~~~~~~~~~~~~~~~~~~ Cash and Carry Managment System ~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Main Menu ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                1- Customer
                2- Administration
                3- TO display Menu
                4- To Exit
                Enter your choice between 1 and 4: 2
```

```
Enter ID: ass
Enter Password: sd
Wrong Id and Password. Enter Again.
Enter ID: Khamaja
Enter Password: fds
Wrong Password. Enter Again.
Enter ID: saad
Enter Password: saad
Wrong ID. Enter Again.
Enter ID: Khamaja
Enter Password: saad
~~~~~~~~~~ Administration ~~~~~~~~~~
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~ Administration ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                Enter 1- To Modify Products.
                Enter 2- To Check Sale.
                Enter 3- To change ID or Password.
                Enter 4- To Display Stoke.
                Enter 5- To Go to Main Menu.
                Enter what you want to do: 1
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Modify Products ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                Enter 1- To Add New Products.
                Enter 2- To Delete Products.
                Enter 3- To Change in Existing Products.
                Enter 4- To Go back to Administration.
                What do you want to do: 1
```

```
Enter new values according to the given column seperated by a comma:
Product Number          Product Name            Product Price           Quantity
101,New Product,5000,60
Product Number already exist. Enter another.
Enter new values according to the given column seperated by a comma:
Product Number          Product Name            Product Price           Quantity
102,Product,600,50
Product Added Successfully.
```

101	100	Hair Oil	860	1000
102	101	new PRouct	60	40
103	102	Product	600	50

```
Enter product number you want to delete: 101
Product deleted successfully.
```

100	99	Cookies	60	1000
101	100	Hair Oil	860	1000
102	102	Product	600	50

```
Enter product number you want to change: 101
1: Product Number
2: Product Name
3: Product Price
4: Quantity

Which column do you want to edit: 1
Old Product Number is: 101
Enter new Product Number: 105
Successfully Changed.
```

```
Which column do you want to edit: 2
Old Product Name is: Product
Enter new Product Name: New Product
Successfully Changed.
```

```
Which column do you want to edit: 3
Old Product Price is: 600
Enter new Product Price: 1000
Successfully Changed.
```

```
Which column do you want to edit: 4
Old Quantity is: 50
Enter new Quantity: 900
Successfully Changed.
```

99	Cookies	60	1000
100	Hair Oil	860	1000
105	New Product	1000	900

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Administration ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                Enter 1- To Modify Products.
                Enter 2- To Check Sale.
                Enter 3- To change ID or Password.
                Enter 4- To Display Stoke.
                Enter 5- To Go to Main Menu.
                Enter what you want to do: 2
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Sale ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                Enter 1- to check Daily sale:
                Enter 2- to check monthly sale:
                Enter 0- to go back:
                What you want to do: 1
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Daily Sale ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                Enter date: 10
                Enter Month: 2
                Enter year: 2022
                Total Daily sale on 10-2-2022 is 1360Rs
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Monthly Sale ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                Enter Month: 2
                Enter year: 2022
                Total monthly sale on 2-2022 is 2860Rs
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~ Administration ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                Enter 1- To Modify Products.
                Enter 2- To Check Sale.
                Enter 3- To change ID or Password.
                Enter 4- To Display Stoke.
                Enter 5- To Go to Main Menu.
                Enter what you want to do: 3
                Enter New ID: Project
                Enter New Password: pf
                Password updated. Successfully.
```

```
 Enter ID: Project
 Enter Password: pf
~~~~~~~~~~~~ Administration ~~~~~~~~~
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Main Menu ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
            1- Customer
            2- Administration
            3- TO display Menu
            4- To Exit
            Enter your choice between 1 and 4: 1

            Enter your name: Ahmed
            Enter your phone number (without dash): +923125311759
            Enter your CNIC (xxxxx-xxxxxxx-x): 37405-0841069-1
            Enter date: 11
            Enter Month: 6
            Enter year: 2023

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Welcome To Our Menu ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      Product Number          Product Name          Product Quantity        Price
      1                       Salt                  1000                    15
      2                       Onion                 1000                    60
      3                       Rice                  1000                    300
      4                       Flour                 1000                    346
      5                       Meat                  1000                    780
      6                       Fish                  1000                    550
      7                       Dairy                 1000                    150
      8                       Socks                 1000                    140
      9                       Fruits                1000                    560
      10                      Sweets                1000                    130
      11                      Snacks                1000                    50
      12                      Toilet Cleaner        1000                    250
      13                      Potato                1000                    70
      14                      Eggs                  1000                    250
      15                      Tomato                1000                    50
      16                      Tea                   1000                    1200
      17                      Paneer                1000                    402
      18                      Carrot                1000                    80
      19                      Radish                1000                    40
      20                      Lady Finger           1000                    100
      21                      Noodles               1000                    75
      22                      House Utilities       1000                    1000
      23                      Lux Soap              1000                    60
      24                      Ariel                 1000                    50
      25                      Surf Excel            1000                    50
      26                      Crockery              1000                    2000
      27                      Cat Food              1000                    1150
      28                      Dog Food              1000                    1150
      29                      Utensils              1000                    1000
      30                      Makeup Products       1000                    1600
```

```
Enter product number and 0 to exit: 15
Product Name is: Tomato
Enter total number of quantities you want: 10
Enter product number and 0 to exit: 23
Product Name is: Lux Soap
Enter total number of quantities you want: 23
Enter product number and 0 to exit: 55
Product Name is: Sugar
Enter total number of quantities you want: 45
Enter product number and 0 to exit: 0

~~~~~~~~~~~~~~~~~ Order Details ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Name: Ahmed
Phone: +923125311759
CNIC: 37405-0841069-1

~~~~~~~~~~~~~ Purchased Products ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Product Name: Tomato
Quantity: 10
Price per unit: 1000
Product Name: Lux Soap
Quantity: 23
Price per unit: 1000
Product Name: Sugar
Quantity: 45
Price per unit: 1000
```


Total Bill
Total bill: 78000
Press 1 for Visa card or 2 for cash payment: 1
Enter your 16-digit visa card number (xxxx-xxxx-xxxx-xxxx): 1111-1111-1111-1111
Enter your 4-digit PIN: 1234
Payment completed.
Thanks For Shopping!!