Net Programming

Final course report on

“Material Inventory Management System”

Created by

AHMED MD JUBAIR

ID: 1911521211

Class: 19lq

Under The Guidance of

Chen Qiong

Associate professor

Department of Computer Science & Technology

Hubei University of Technology

Wuhan, Hubei, P. R. China

May 31, 2023

Contents

Acknowledgment

I would like to extend my heartfelt appreciation to my project instructor, Assoc. Prof. **Chen Qiong**, for her invaluable guidance and unwavering support throughout the development of my project. Her expertise in net programming has been instrumental in helping me overcome challenges, and her patience and encouragement have kept me motivated throughout the project's duration. I consider myself fortunate to have had such a dedicated and supportive instructor, and I am truly grateful for everything she has done for me. Thank you, Assoc. Prof. Chen Qiong, for your exceptional mentorship, which has contributed greatly to the success of this project.

Abstract

The Material Inventory Management System (MIMS) is a comprehensive software solution designed to optimize and streamline the management of materials within an organization. Efficient inventory management plays a critical role in ensuring uninterrupted production, reducing costs, and improving overall operational efficiency. The MIMS aims to address the challenges faced by businesses in tracking, controlling, and analyzing their material inventory.

The system provides a centralized platform for managing material inventory across multiple locations, enabling real-time visibility into stock levels, usage, and replenishment needs. It offers features such as automated stock tracking, barcode scanning, and integration with suppliers and procurement systems. With these functionalities, MIMS facilitates accurate forecasting, timely reordering, and efficient allocation of materials, thereby reducing the risk of stockouts or excess inventory.

Moreover, MIMS incorporates advanced analytics capabilities to generate insightful reports and forecasts, enabling data-driven decision-making. It provides key performance indicators (KPIs) related to inventory turnover, carrying costs, lead times, and supplier performance, helping organizations optimize their inventory management strategies.

Additionally, the MIMS offers integration with other enterprise systems such as Enterprise Resource Planning (ERP) software, enabling seamless data synchronization and eliminating data silos. The system can be customized to accommodate the unique requirements of different industries and businesses, ensuring a tailored solution for specific inventory management needs.

The implementation of the Material Inventory Management System offers numerous benefits, including improved inventory accuracy, reduced carrying costs, increased productivity, and enhanced customer satisfaction. Organizations can achieve better control, efficiency, and profitability by automating and optimizing material inventory management processes.

Keywords: material inventory management, inventory optimization, real-time visibility, supply chain management, analytics, enterprise integration.

Introduction

1.1 General Purpose

The Material Inventory Management System (MIMS) is a robust software solution developed to address the complexities and challenges associated with managing material inventory within organizations. Efficient inventory management is crucial for businesses to optimize their operations, minimize costs, and ensure smooth production processes. The MIMS serves as a centralized platform that streamlines and enhances the tracking, control, and analysis of material inventory across multiple locations.

Managing material inventory effectively is a multifaceted task, particularly for organizations operating in diverse industries or dealing with large volumes of materials. Manual inventory management processes are often time-consuming, prone to errors, and lack real-time visibility, resulting in stockouts, excess inventory, and inefficient resource allocation. Furthermore, traditional inventory management methods often lack the ability to provide actionable insights and analytics to support data-driven decision-making.

The primary objective of the Material Inventory Management System is to automate and optimize the management of material inventory within organizations. The system aims to provide real-time visibility into stock levels, usage patterns, and replenishment needs, facilitating accurate forecasting and timely procurement. It also seeks to streamline inventory tracking, control, and allocation, reducing the risk of stockouts and excess inventory. Additionally, the MIMS aims to generate comprehensive reports and analytics, enabling organizations to make informed decisions and enhance overall inventory management strategies.

The implementation of the Material Inventory Management System offers numerous benefits to organizations. These include improved inventory accuracy, reduced carrying costs, increased operational efficiency, enhanced customer satisfaction, and streamlined supply chain management. By automating and optimizing material inventory management processes, the MIMS empowers organizations to effectively allocate resources, minimize waste, and improve profitability.

The Material Inventory Management System is designed to be versatile and adaptable, catering to the unique requirements of various industries and businesses. The system can be customized to accommodate specific inventory management needs and seamlessly integrated with other enterprise systems, such as Enterprise Resource Planning (ERP) software. From procurement to consumption, the MIMS provides a comprehensive solution for managing material inventory, ensuring seamless operations and efficient inventory control.

The Material Inventory Management System (MIMS) aims to enhance overall operational efficiency for businesses. By automating inventory management tasks, MIMS enables businesses to reallocate resources to other areas of their operations. Additionally, MIMS streamlines inventory management processes, reducing the time and effort required to maintain optimal inventory levels.

In summary, the general purpose of the Material Inventory Management System is to enable businesses to efficiently manage their inventory, optimize its usage, minimize inventory costs, and ensure the timely availability of materials. Through accurate inventory records, optimized inventory levels, and improved supply chain management, MIMS assists businesses in enhancing their overall operational efficiency and achieving their desired business objectives.

1.2 Significance

The Material Inventory Management System (MIMS) holds significant importance for businesses seeking to optimize their inventory management processes. Efficient inventory management is vital for organizations as it directly impacts operational efficiency, cost reduction, and overall productivity. The introduction highlights the significance of MIMS in addressing the challenges and complexities associated with material inventory management.

Effective inventory management plays a critical role in ensuring uninterrupted production, minimizing costs, and meeting customer demands. By maintaining optimal inventory levels, businesses can avoid stockouts, reduce excess inventory, and improve resource allocation. The introduction

emphasizes the importance of efficient inventory management as a key driver of business success.

Manual inventory management processes are often time-consuming, error-prone, and lack real-time visibility. These challenges result in inaccurate stock records, inefficient utilization of resources, and increased operational costs. The introduction highlights the limitations of traditional inventory management methods and the need for an automated solution like MIMS.

The introduction emphasizes the role of MIMS in addressing the challenges faced by businesses in managing their material inventory. MIMS provides a centralized platform that offers real-time visibility into stock levels, usage patterns, and replenishment needs. It streamlines inventory tracking, control, and allocation, thereby reducing the risk of stockouts, minimizing excess inventory, and optimizing resource utilization.

Implementing the Material Inventory Management System offers several benefits to organizations. The introduction highlights these benefits, including improved inventory accuracy, reduced carrying costs, increased productivity, enhanced customer satisfaction, and streamlined supply chain management. By automating and optimizing inventory management processes, MIMS enables businesses to achieve better control, efficiency, and profitability.

The introduction highlights the scope and adaptability of MIMS, emphasizing its ability to cater to the unique requirements of various industries and businesses. MIMS can be customized to accommodate specific inventory management needs and seamlessly integrated with other enterprise systems, such as Enterprise Resource Planning (ERP) software.

In conclusion, the introduction highlights the significance of the Material Inventory Management System (MIMS) in addressing the challenges faced by businesses in managing their material inventory. MIMS offers streamlined inventory management processes, improved efficiency, and a wide range of benefits, making it a crucial solution for businesses aiming to optimize their operations and achieve their objectives.

1.3 Literature Review

Material inventory management is a fundamental process that involves the careful monitoring and control of inventory levels to ensure the availability of materials when needed while minimizing the associated carrying costs. It is a crucial component of successful supply chain management. In order to automate and streamline these inventory management processes, organizations often implement a material inventory management system (MIMS), which is a software solution specifically designed for this purpose. This literature review aims to provide a comprehensive overview of the existing research and literature on MIMS, focusing on its benefits, challenges, and best practices for implementation.

MIMS encompasses various aspects of material inventory management, including the identification of required materials for production, their procurement, storage, inventory control, and ensuring that the inventory levels align with production needs. The literature review will explore the significance of MIMS in managing a company's assets and ensuring the availability of materials when required. It will delve into the key concepts, principles, and tools commonly used in the implementation of MIMS, shedding light on the importance of accurate forecasting, effective supply chain management, and efficient inventory control.

Accurate forecasting plays a pivotal role in MIMS, enabling organizations to predict their future material requirements and plan accordingly. Effective supply chain management is another crucial aspect, ensuring that the right materials are available at the right time and in the appropriate quantities. Furthermore, efficient inventory control is imperative for optimizing inventory levels, minimizing the risks of stockouts, excess inventory, and waste.

By conducting an in-depth review of the literature on MIMS, this study aims to provide valuable insights into the benefits organizations can derive from implementing this system, the challenges they may encounter during the implementation process, and the best practices that can contribute to its successful adoption. The findings of this literature review will serve as a foundation for organizations seeking to enhance their material inventory management practices by implementing a robust MIMS.

1.4 Methodology

The Material Inventory Management System (MIMS) project requires a well-defined methodology to ensure its successful implementation. This section outlines the proposed methodology for developing and deploying the MIMS, which encompasses the various stages and activities involved in the project.

Project Planning and Scope Definition:

The first step is to establish the project's objectives, scope, and deliverables. This involves identifying the specific inventory management challenges and requirements of the organization, as well as defining the desired outcomes of implementing the MIMS. The project team should collaborate with key stakeholders to gather requirements, set project milestones, and allocate necessary resources.

System Analysis and Design:

The system analysis phase involves conducting a detailed assessment of the organization's current inventory management processes and identifying areas for improvement. This includes analyzing data flow, inventory tracking mechanisms, and integration requirements with other systems, such as ERP or warehouse management systems. Based on the analysis, a comprehensive system design is created, outlining the architecture, modules, and functionalities of the MIMS.

Development and Configuration:

During the development phase, the MIMS is built according to the system design specifications. This may involve customizing off-the-shelf inventory management software or developing a bespoke solution, depending on the organization's requirements. The development team should ensure that the system integrates seamlessly with existing systems and databases, and conduct rigorous testing to verify its functionality and performance.

Data Migration and System Integration:

Once the MIMS is developed, the next step is to migrate existing inventory data to the new system. This involves extracting, cleansing, and transforming data from legacy systems or spreadsheets into a format compatible with the MIMS. Additionally, integration with other systems, such as procurement or

sales systems, should be established to enable real-time data exchange and ensure a holistic view of inventory across the organization.

User Training and Change Management:

User training and change management are critical for the successful adoption of the MIMS. The project team should conduct comprehensive training sessions to familiarize users with the new system, its functionalities, and best practices for inventory management. Change management strategies should also be implemented to address any resistance to change, communicate the benefits of the MIMS, and ensure a smooth transition and user acceptance.

System Testing and Deployment:

Before the MIMS is deployed in a live production environment, thorough testing should be conducted to validate its performance, reliability, and security. This includes functional testing, integration testing, performance testing, and user acceptance testing. Once the system is deemed ready, a carefully planned deployment strategy is executed, ensuring minimal disruption to ongoing operations.

Post-Deployment Support and Continuous Improvement:

After the MIMS is deployed, ongoing support and maintenance are essential to address any issues, provide assistance to users, and make necessary enhancements or modifications. Regular monitoring and evaluation of the system's performance should be conducted, with feedback from users and stakeholders used to drive continuous improvement and optimize the system's effectiveness.

By following this methodology, the Material Inventory Management System project can be executed in a structured and systematic manner, ensuring successful implementation and maximizing the benefits for the organization.

1.5 Organization of the paper

This paper aims to delve into the Material Inventory Management System (MIMS) and its implications for efficient and effective inventory management in businesses. The paper is structured into four main sections: Introduction, Significance, Methodology, and Conclusion. Each section provides valuable

insights and details regarding the importance, implementation, and benefits of MIMS.

The Introduction section serves as a comprehensive overview of MIMS, emphasizing its significance and advantages for businesses. It outlines the core functionalities and features of the MIMS solution while providing a roadmap for the subsequent sections of the paper.

Moving on, the Significance section explores the critical role of MIMS in business operations. It delves into how effective inventory management can profoundly impact a company's operations and profitability. Additionally, it elucidates how MIMS assists businesses in optimizing inventory utilization, mitigating inventory costs, and ensuring the timely availability of materials.

The Methodology section serves as a guide for implementing MIMS within an organization. It presents a detailed framework encompassing various steps, such as conducting a needs assessment, designing the system, migrating data, configuring the system, conducting thorough testing, providing comprehensive user training, and deploying the system.

Lastly, the Conclusion section provides a summary of the main points discussed throughout the paper. It reiterates the significance of MIMS for businesses and underlines its potential benefits, including improved inventory accuracy, enhanced efficiency, reduced inventory costs, and increased customer satisfaction. Furthermore, the section emphasizes the importance of implementing and adopting MIMS to fully capitalize on these advantages.

In conclusion, this paper offers a comprehensive exploration of the Material Inventory Management System, covering its importance, implementation methodology, and resulting benefits. By adhering to the methodology presented herein, businesses can successfully implement MIMS, optimizing their inventory management practices and ultimately enhancing their overall operations and profitability.

Chapter 2: Requirements

2.1 Basic Information Requirement

The Material Inventory Management System is an innovative software solution created to monitor and control the inventory levels of materials utilized within a company's operational processes. This comprehensive system incorporates essential details like material identification, name, specifications, type, and unit of measurement. These features enable precise tracking and seamless reporting of inventory data, ensuring efficient inventory management.

Material number

The unique identifier assigned to each material within the inventory system serves as a distinguishing factor and facilitates the monitoring of material movement within the system. It allows for effective tracking and traceability of materials throughout their lifecycle in the inventory management process.

Material name

The material name serves as a concise description of the item within the inventory system. It plays a crucial role in swiftly identifying the material and differentiating it from other items present in the inventory.

Specification

The specification of a material entails comprehensive information regarding its quality, characteristics, and properties. These details are of utmost importance in guaranteeing the appropriate selection and utilization of the material for specific tasks or purposes.

Type

Materials can be categorized into various groups or types based on specific characteristics or attributes. This classification system aids in effectively organizing the inventory system, streamlining inventory management processes, and simplifying the tracking and control of inventory levels.

Measurement unit

The measurement unit is employed to quantify the quantity of material present in the inventory system. This crucial information is essential for various purposes, including calculating the required amount of material for a specific task or project and determining when to reorder materials in case inventory levels reach a minimum threshold.

In summary, the Material Inventory Management System leverages fundamental data elements such as material number, material name, specification, type, and measurement unit to enable precise monitoring and reporting of inventory data. By utilizing this information, the system enhances inventory management practices, streamlines operational processes, and maximizes the efficient utilization of materials within the organization.

2.2 Implemented functions Requirement

The Material Inventory Management System is a software application developed in C# that aids in the efficient management of material inventory for businesses and organizations. This system encompasses various functionalities to streamline inventory control and tracking processes.

2.2.1 Input of basic information about materials

This feature enables users to enter and store essential material information, including name, description, unit of measure, supplier details, and other pertinent data. Additionally, the system allows users to assign a distinct identifier (such as an SKU or part number) to each material, facilitating efficient tracking and management of inventory.

2.2.2 Query and modification of basic material information

The system should provide users with the capability to search for materials using different criteria, including name, description, supplier information, and

other pertinent details. Users should have the ability to access and review all the fundamental information related to a material, as well as make necessary modifications. Additionally, the system should maintain a comprehensive record of all changes made to the basic material information for the purpose of auditing.

2.2.3 Input of inbound material information

This function enables users to enter details regarding the arrival of materials, including the date of receipt, received quantity, supplier information, and any other pertinent information. Additionally, the system provides the ability for users to link inbound materials with a purchase order or another receiving document for the purpose of tracking.

2.2.4 Query and modification of inbound material information

The system should provide users with the capability to search for information about inbound materials using different criteria, such as the receiving date, supplier details, and other pertinent information. Users should have the ability to access and review all the relevant inbound information associated with a particular material, and if necessary, make modifications. Additionally, the system must maintain a comprehensive record of all changes made to the inbound material information to facilitate auditing.

2.2.5 Input of outbound material information

This function enables users to enter details regarding the departure of materials, including the date of shipment, quantity shipped, customer information, and any other pertinent information. Additionally, the system provides the ability for users to link outbound materials with a sales order or another shipping document for the purpose of tracking.

2.2.6 Query and modification of outbound material information

The system should provide users with the capability to search for information about outbound materials using different criteria, such as the shipping date, customer details, and other pertinent information. Users should have the

ability to access and review all the relevant outbound information associated with a particular material, and if necessary, make modifications. Additionally, the system must maintain a comprehensive record of all changes made to the outbound material information to facilitate auditing.

2.2.7 Query of material balance information

This function enables users to retrieve up-to-date information on material balances, including the current quantity of materials in stock, quantity received, quantity shipped, and other relevant details. The system is designed to provide real-time data on material balances, assisting users in inventory management and planning tasks.

2.2.8 Browsing of material balance information

This function provides users with the ability to easily navigate and explore material balance information for all stored materials in the system. The information is presented in a user-friendly format that allows users to efficiently view the quantity of each material in stock, the quantity received, and the quantity shipped. Moreover, the system enables users to filter and sort the material balance information based on various criteria, enhancing their ability to analyze and manage inventory effectively.

2.2.9 Exit from the system

This function enables users to securely exit the system, ensuring that unauthorized individuals do not have access to sensitive information. Additionally, the system includes an automatic exit feature that activates after a period of user inactivity. This measure helps to prevent unauthorized access and protect the system and its data.

It is essential to acknowledge that a Material Inventory Management System can encompass additional functionalities tailored to the specific requirements of the organization implementing it. These functionalities may include features such as setting reorder points to manage inventory levels effectively, generating purchase orders to facilitate the procurement process, and generating reports to provide insights and analytics on material inventory. The system should be designed to accommodate and address the unique needs and

goals of the organization, ensuring its effectiveness and efficiency in managing materials and supporting overall operations.

2.3 Hardware and Software Requirements

2.3.1 Hardware Requirement

- Processor: Intel Core i5 or higher
- RAM: 4 GB or higher
- Hard Disk Space: 100 GB or higher

2.3.2 Software Requirement

- Operating System: Windows 10 or higher
- Microsoft Visual Studio 2022 with C# programming language
- .NET Framework 4.7.2 or higher

Chapter 3: Code Implementation

3.1 Input of basic information about materials

```
static void InputMaterial()
    {
        Console.WriteLine("Enter Material Number:");
        if (!int.TryParse(Console.ReadLine(), out int materialNumber))
        {
            Console.WriteLine("Invalid material number. Please enter a
valid number.");
            return;
        }

        Console.WriteLine("Enter Material Name:");
        string materialName = Console.ReadLine();

        Console.WriteLine("Enter Specification:");
        string specification = Console.ReadLine();

        Console.WriteLine("Enter Type:");
        string type = Console.ReadLine();

        Console.WriteLine("Enter Measurement Unit:");
        string measurementUnit = Console.ReadLine();

        Material material = new Material()
        {
            MaterialNumber = materialNumber,
            MaterialName = materialName,
            Specification = specification,
            Type = type,
            MeasurementUnit = measurementUnit
        };

        materials.Add(material);

        Console.WriteLine("Material added successfully");
    }8
```

3.2 Query and modification of basic material information

```
static void QueryAndModifyMaterial()
    {
        Console.WriteLine("Enter Material Number or Material Name:");
        string searchKey = Console.ReadLine();

        List<Material> foundMaterials = materials
            .Where(m => m.MaterialNumber.ToString() == searchKey ||
m.MaterialName == searchKey)
            .ToList();
```

```
        if (foundMaterials.Count == 0)
        {
            Console.WriteLine("No materials found");
            return;
        }

        Console.WriteLine("Found Materials:");

        foreach (Material material in foundMaterials)
        {
            Console.WriteLine("Material Number: " +
material.MaterialNumber);
            Console.WriteLine("Material Name: " + material.MaterialName);
            Console.WriteLine("Specification: " + material.Specification);
            Console.WriteLine("Type: " + material.Type);
            Console.WriteLine("Measurement Unit: " +
material.MeasurementUnit);
            Console.WriteLine();
        }

        Console.WriteLine("Enter the Material Number of the material you
want to modify:");
        if (!int.TryParse(Console.ReadLine(), out int materialNumber))
        {
            Console.WriteLine("Invalid material number. Please enter a
valid number.");
            return;
        }

        Material foundMaterial = foundMaterials.FirstOrDefault(m =>
m.MaterialNumber == materialNumber);

        if (foundMaterial == null)
        {
            Console.WriteLine("Material not found");
            return;
        }

        Console.WriteLine("Enter the new Material Name:");
        string newMaterialName = Console.ReadLine();

        Console.WriteLine("Enter the new Specification:");
        string newSpecification = Console.ReadLine();

        Console.WriteLine("Enter the new Type:");
        string newType = Console.ReadLine();

        Console.WriteLine("Enter the new Measurement Unit:");
        string newMeasurementUnit = Console.ReadLine();

        foundMaterial.MaterialName = newMaterialName;
        foundMaterial.Specification = newSpecification;
        foundMaterial.Type = newType;
        foundMaterial.MeasurementUnit = newMeasurementUnit;

        Console.WriteLine("Material modified successfully");
    }
```

3.3 Input of inbound material information

```
static void InputInboundMaterial()
    {
        Console.WriteLine("Enter Material Number:");
        if (!int.TryParse(Console.ReadLine(), out int materialNumber))
        {
            Console.WriteLine("Invalid material number. Please enter a
valid number.");
            return;
        }

        Console.WriteLine("Enter Quantity:");
        if (!int.TryParse(Console.ReadLine(), out int quantity))
        {
            Console.WriteLine("Invalid quantity. Please enter a valid
number.");
            return;
        }

        Console.WriteLine("Enter Date (YYYY-MM-DD):");
        if (!DateTime.TryParse(Console.ReadLine(), out DateTime date))
        {
            Console.WriteLine("Invalid date format. Please enter a valid
date (YYYY-MM-DD).");
            return;
        }

        Console.WriteLine("Enter Supplier Name:");
        string supplierName = Console.ReadLine();

        InboundMaterial inboundMaterial = new InboundMaterial()
        {
            MaterialNumber = materialNumber,
            Quantity = quantity,
            Date = date,
            SupplierName = supplierName
        };

        inboundMaterials.Add(inboundMaterial);

        Console.WriteLine("Inbound material added successfully");
    }
```

3.4 Query and modification of inbound material information

```
static void QueryAndModifyInboundMaterial()
    {
        Console.WriteLine("Enter Material Number:");
        if (!int.TryParse(Console.ReadLine(), out int materialNumber))
        {
```

```
            Console.WriteLine("Invalid material number. Please enter a
valid number.");
            return;
        }

        List<InboundMaterial> foundInboundMaterials = inboundMaterials
            .Where(im => im.MaterialNumber == materialNumber)
            .ToList();

        if (foundInboundMaterials.Count == 0)
        {
            Console.WriteLine("No inbound materials found for the specified
material number");
            return;
        }

        Console.WriteLine("Found Inbound Materials:");

        foreach (InboundMaterial inboundMaterial in foundInboundMaterials)
        {
            Console.WriteLine("Material Number: " +
inboundMaterial.MaterialNumber);
            Console.WriteLine("Quantity: " + inboundMaterial.Quantity);
            Console.WriteLine("Date: " +
inboundMaterial.Date.ToString("yyyy-MM-dd"));
            Console.WriteLine("Supplier Name: " +
inboundMaterial.SupplierName);
            Console.WriteLine();
        }

        Console.WriteLine("Enter the Date (YYYY-MM-DD) of the inbound
material you want to modify:");
        if (!DateTime.TryParse(Console.ReadLine(), out DateTime date))
        {
            Console.WriteLine("Invalid date format. Please enter a valid
date (YYYY-MM-DD).");
            return;
        }

        InboundMaterial foundInboundMaterial =
foundInboundMaterials.FirstOrDefault(im => im.Date == date);

        if (foundInboundMaterial == null)
        {
            Console.WriteLine("Inbound material not found");
            return;
        }

        Console.WriteLine("Enter the new Quantity:");
        if (!int.TryParse(Console.ReadLine(), out int newQuantity))
        {
            Console.WriteLine("Invalid quantity. Please enter a valid
number.");
            return;
        }

        Console.WriteLine("Enter the new Date (YYYY-MM-DD):");
        if (!DateTime.TryParse(Console.ReadLine(), out DateTime newDate))
```

```
        {
            Console.WriteLine("Invalid date format. Please enter a valid
date (YYYY-MM-DD).");
            return;
        }

        Console.WriteLine("Enter the new Supplier Name:");
        string newSupplierName = Console.ReadLine();

        foundInboundMaterial.Quantity = newQuantity;
        foundInboundMaterial.Date = newDate;
        foundInboundMaterial.SupplierName = newSupplierName;

        Console.WriteLine("Inbound material modified successfully");
    }
```

3.5 Input of outbound material information

```
static void InputOutboundMaterial()
    {
        Console.WriteLine("Enter Material Number:");
        if (!int.TryParse(Console.ReadLine(), out int materialNumber))
        {
            Console.WriteLine("Invalid material number. Please enter a
valid number.");
            return;
        }

        Console.WriteLine("Enter Quantity:");
        if (!int.TryParse(Console.ReadLine(), out int quantity))
        {
            Console.WriteLine("Invalid quantity. Please enter a valid
number.");
            return;
        }

        Console.WriteLine("Enter Date (YYYY-MM-DD):");
        if (!DateTime.TryParse(Console.ReadLine(), out DateTime date))
        {
            Console.WriteLine("Invalid date format. Please enter a valid
date (YYYY-MM-DD).");
            return;
        }

        Console.WriteLine("Enter Customer Name:");
        string customerName = Console.ReadLine();

        OutboundMaterial outboundMaterial = new OutboundMaterial()
        {
            MaterialNumber = materialNumber,
            Quantity = quantity,
            Date = date,
            CustomerName = customerName
        };

        outboundMaterials.Add(outboundMaterial);
```

```
        Console.WriteLine("Outbound material added successfully");
    }
```

3.6 Query and modification of outbound material information

```
static void QueryAndModifyOutboundMaterial()
    {
        Console.WriteLine("Enter Material Number:");
        if (!int.TryParse(Console.ReadLine(), out int materialNumber))
        {
            Console.WriteLine("Invalid material number. Please enter a
valid number.");
            return;
        }

        List<OutboundMaterial> foundOutboundMaterials = outboundMaterials
            .Where(om => om.MaterialNumber == materialNumber)
            .ToList();

        if (foundOutboundMaterials.Count == 0)
        {
            Console.WriteLine("No outbound materials found for the
specified material number");
            return;
        }

        Console.WriteLine("Found Outbound Materials:");

        foreach (OutboundMaterial outboundMaterial in
foundOutboundMaterials)
        {
            Console.WriteLine("Material Number: " +
outboundMaterial.MaterialNumber);
            Console.WriteLine("Quantity: " + outboundMaterial.Quantity);
            Console.WriteLine("Date: " +
outboundMaterial.Date.ToString("yyyy-MM-dd"));
            Console.WriteLine("Customer Name: " +
outboundMaterial.CustomerName);
            Console.WriteLine();
        }

        Console.WriteLine("Enter the Date (YYYY-MM-DD) of the outbound
material you want to modify:");
        if (!DateTime.TryParse(Console.ReadLine(), out DateTime date))
        {
            Console.WriteLine("Invalid date format. Please enter a valid
date (YYYY-MM-DD).");
            return;
        }

        OutboundMaterial foundOutboundMaterial =
foundOutboundMaterials.FirstOrDefault(om => om.Date == date);

        if (foundOutboundMaterial == null)
        {
```

```
            Console.WriteLine("Outbound material not found");
            return;
        }

        Console.WriteLine("Enter the new Quantity:");
        if (!int.TryParse(Console.ReadLine(), out int newQuantity))
        {
            Console.WriteLine("Invalid quantity. Please enter a valid
number.");
            return;
        }

        Console.WriteLine("Enter the new Date (YYYY-MM-DD):");
        if (!DateTime.TryParse(Console.ReadLine(), out DateTime newDate))
        {
            Console.WriteLine("Invalid date format. Please enter a valid
date (YYYY-MM-DD).");
            return;
        }

        Console.WriteLine("Enter the new Customer Name:");
        string newCustomerName = Console.ReadLine();

        foundOutboundMaterial.Quantity = newQuantity;
        foundOutboundMaterial.Date = newDate;
        foundOutboundMaterial.CustomerName = newCustomerName;

        Console.WriteLine("Outbound material modified successfully");
    }
```

3.7 Query of material balance information

```
static void QueryMaterialBalance()
    {
        Console.WriteLine("Enter Material Number:");
        if (!int.TryParse(Console.ReadLine(), out int materialNumber))
        {
            Console.WriteLine("Invalid material number. Please enter a
valid number.");
            return;
        }

        int inboundQuantity = inboundMaterials
            .Where(im => im.MaterialNumber == materialNumber)
            .Sum(im => im.Quantity);

        int outboundQuantity = outboundMaterials
            .Where(om => om.MaterialNumber == materialNumber)
            .Sum(om => om.Quantity);

        int balance = inboundQuantity - outboundQuantity;

        Console.WriteLine("Material Balance for Material Number " +
materialNumber + ":");
        Console.WriteLine("Inbound Quantity: " + inboundQuantity);
        Console.WriteLine("Outbound Quantity: " + outboundQuantity);
        Console.WriteLine("Balance: " + balance);
```

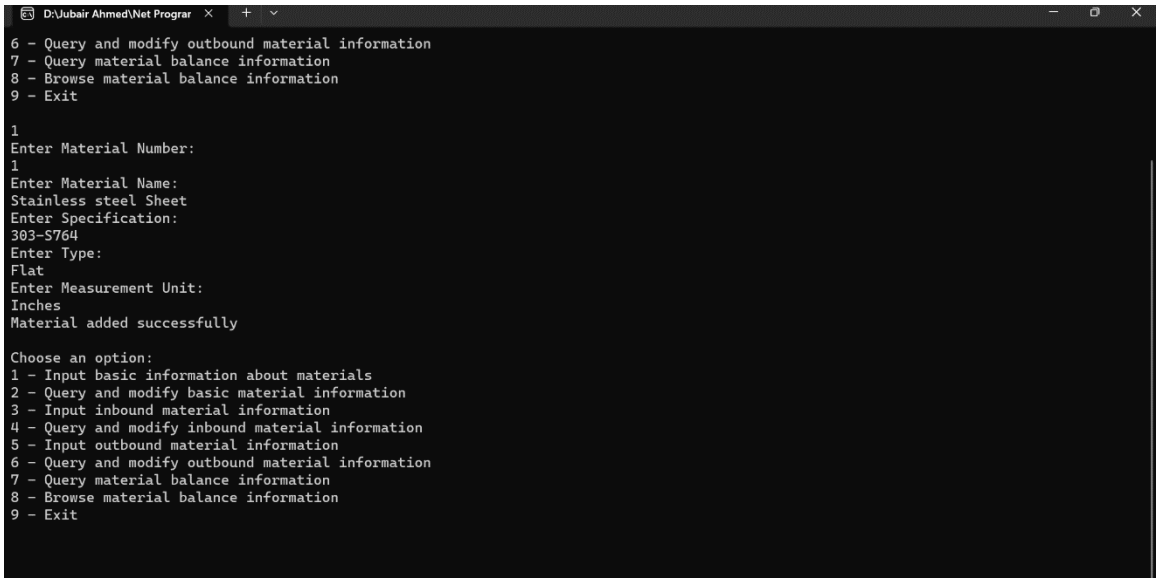
```
    }
```

3.8 Browsing of material balance information

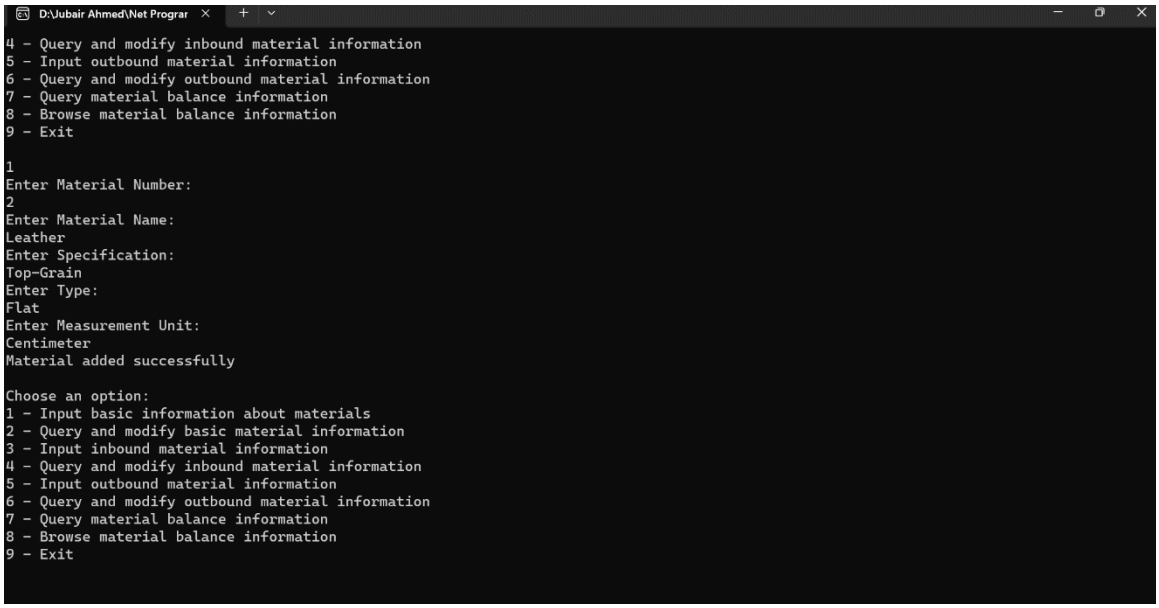
```
static void BrowseMaterialBalance()
    {
        Console.WriteLine("Material Balance Information:");

        foreach (Material material in materials)
        {
            int inboundQuantity = inboundMaterials
                .Where(im => im.MaterialNumber == material.MaterialNumber)
                .Sum(im => im.Quantity);

            int outboundQuantity = outboundMaterials
                .Where(om => om.MaterialNumber == material.MaterialNumber)
                .Sum(om => om.Quantity);

            int balance = inboundQuantity - outboundQuantity;

            Console.WriteLine("Material Number: " +
material.MaterialNumber);
            Console.WriteLine("Material Name: " + material.MaterialName);
            Console.WriteLine("Inbound Quantity: " + inboundQuantity);
            Console.WriteLine("Outbound Quantity: " + outboundQuantity);
            Console.WriteLine("Balance: " + balance);
            Console.WriteLine();
        }
```

Chapter 4: Code Testing

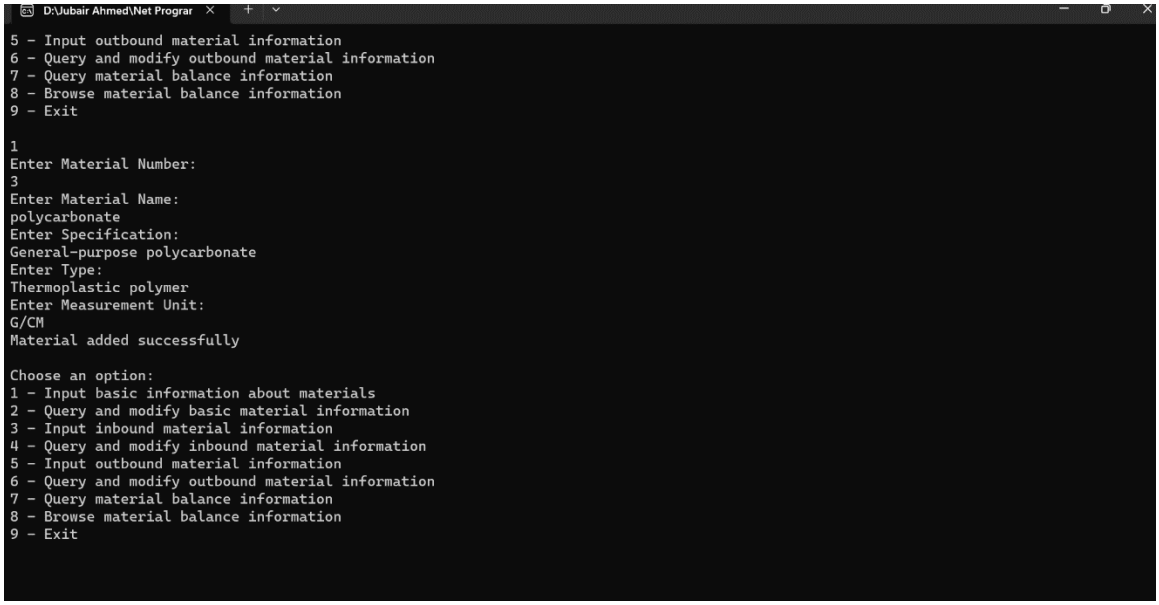
4.1 Input of basic information about materials



Material added successfully 1

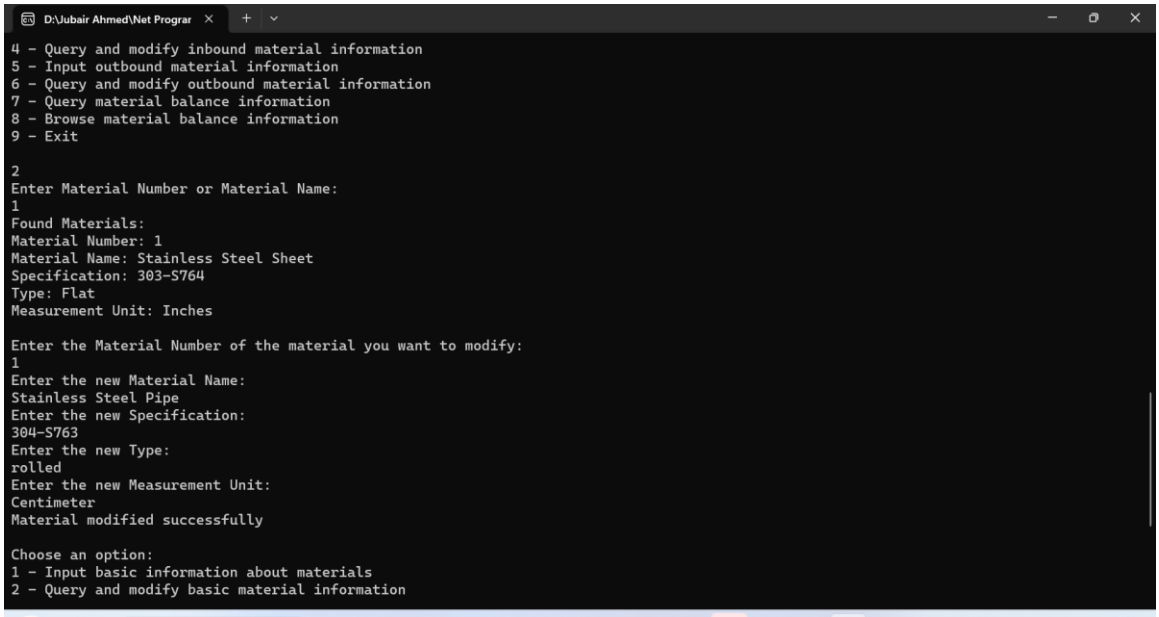


Material added successfully 2



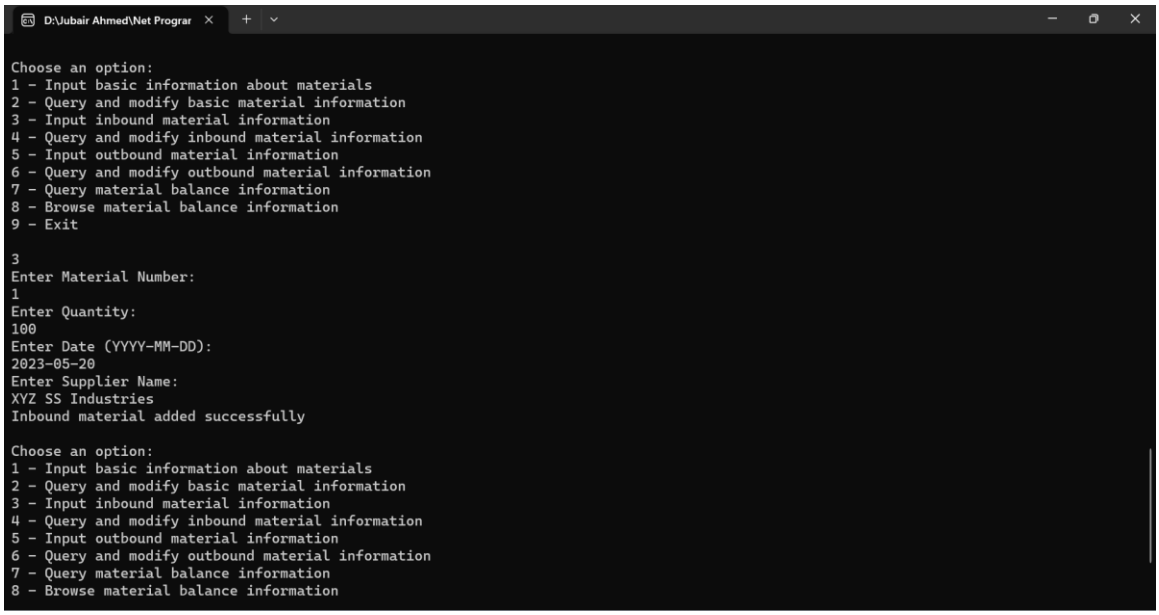
Material added successfully 3

4.2 Query and modification of basic material information

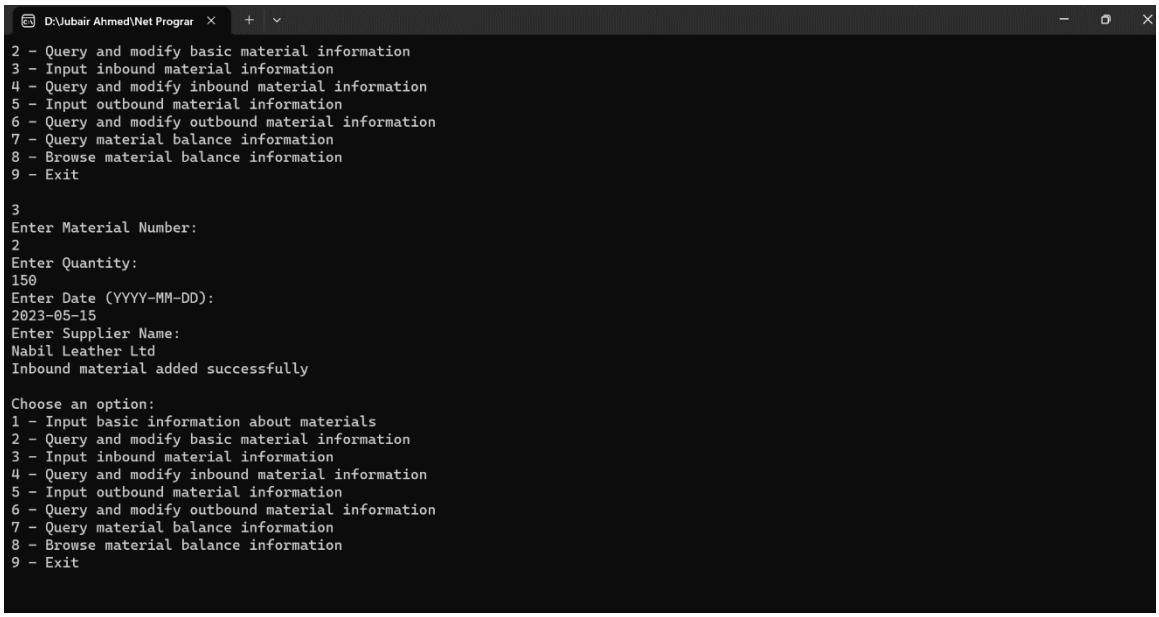


Material queried and modified successfully

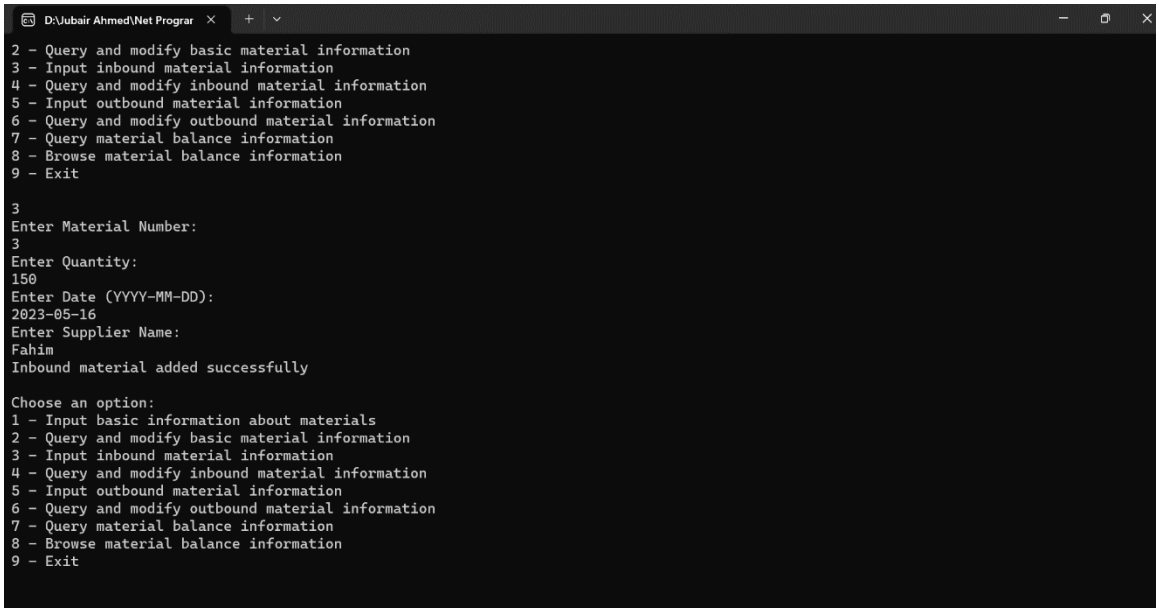
4.3 Input of inbound material information



Inbound material added successfully 1

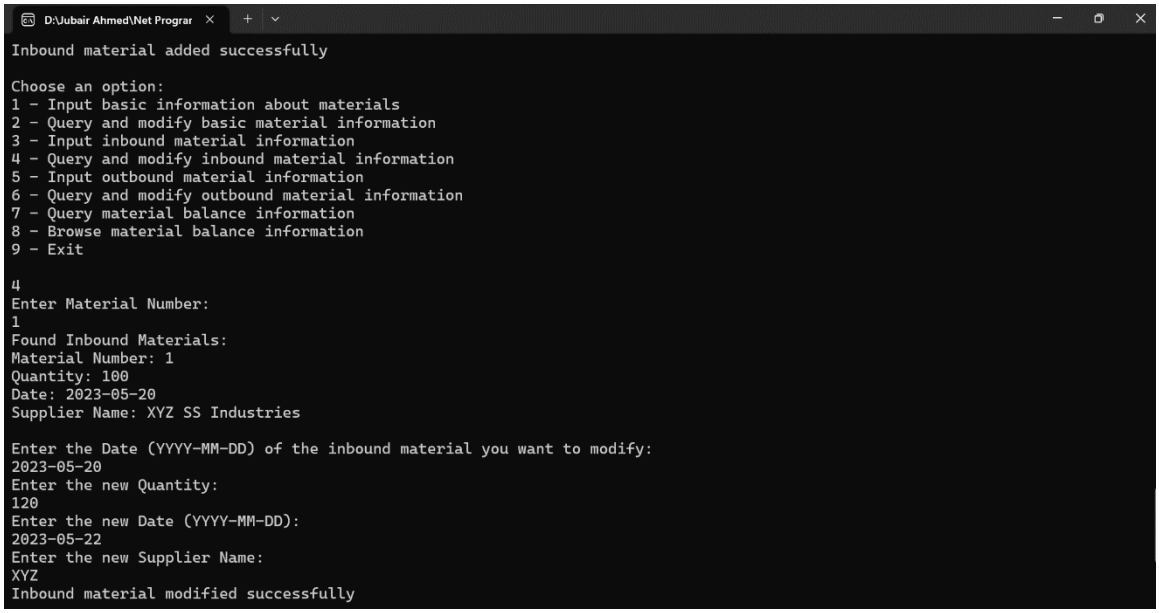


Inbound material added successfully 2



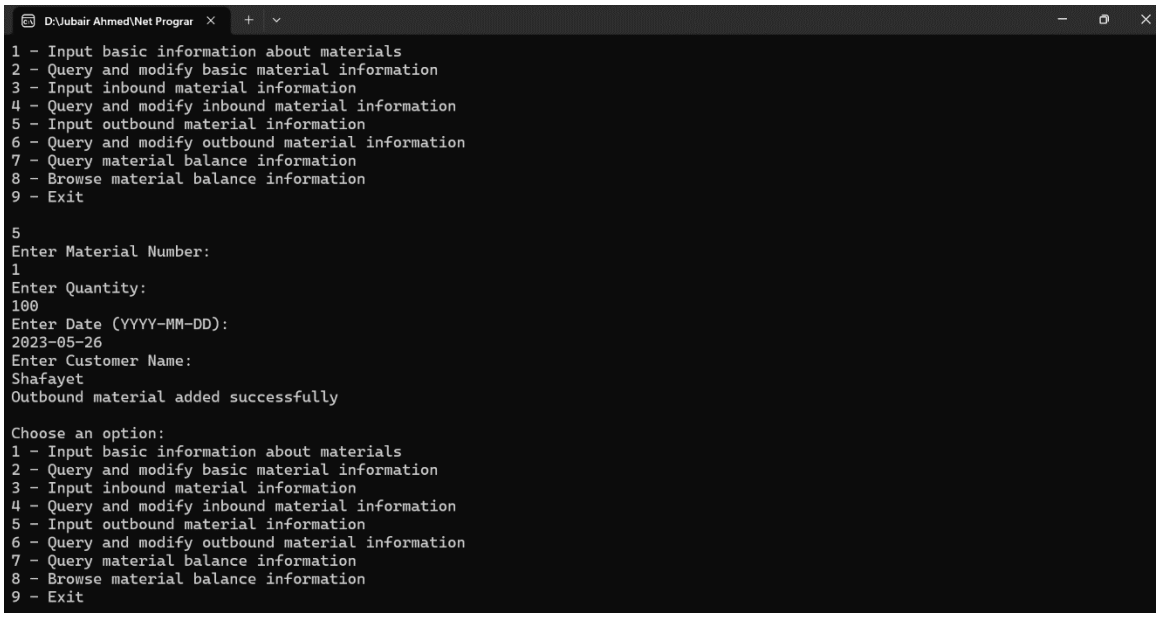
Inbound material added successfully 3

4.4 Query and modification of inbound material information

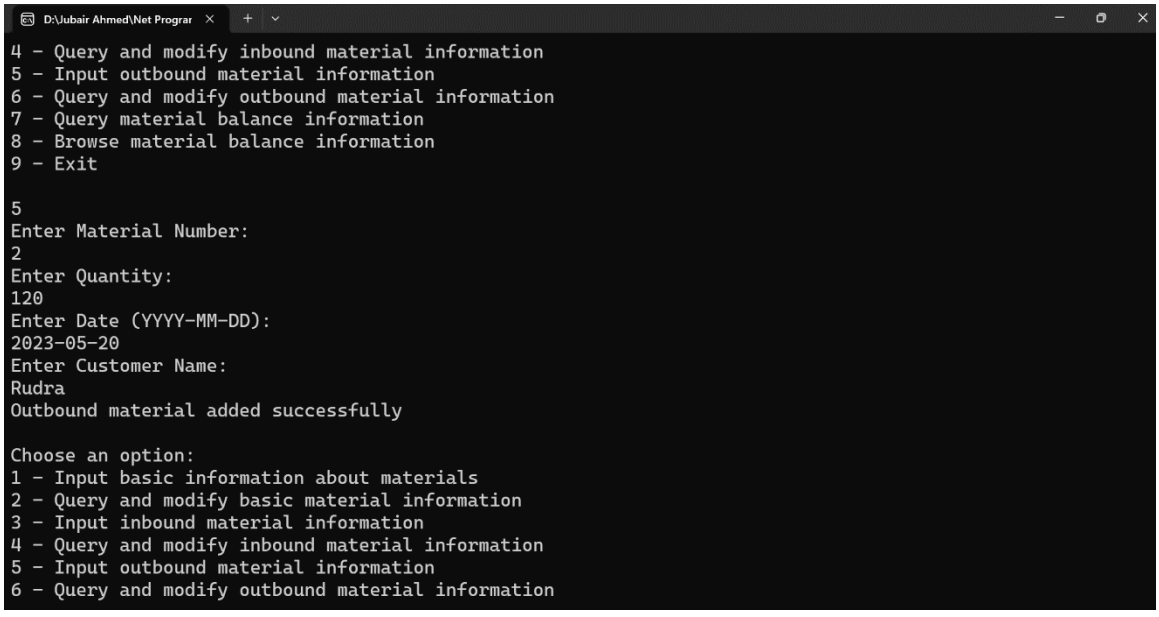


Inbound material queried and modified successfully

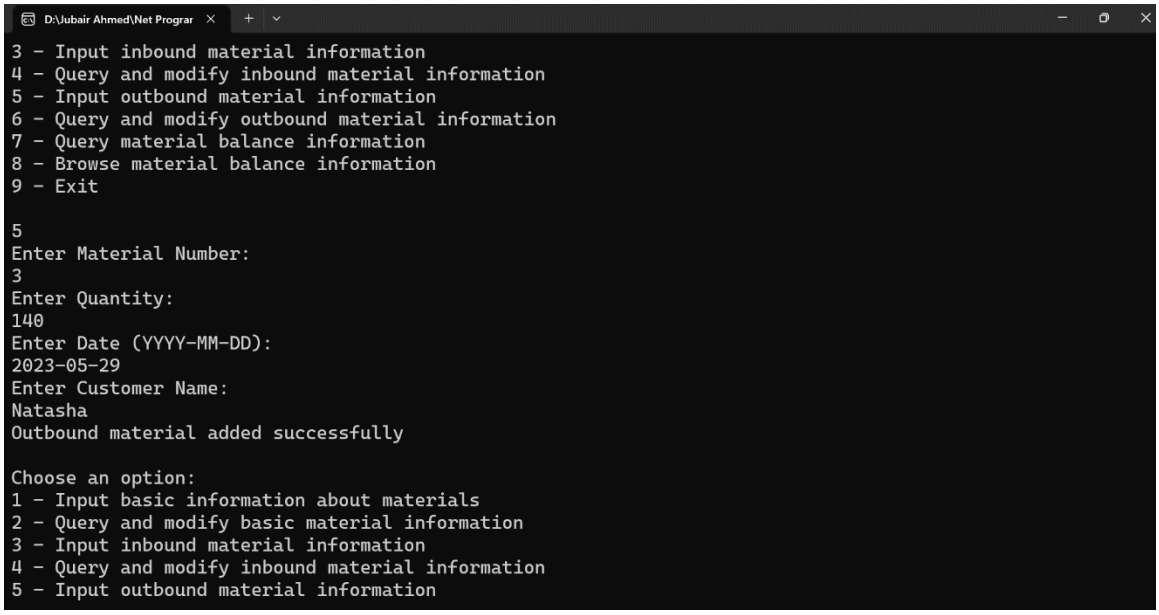
4.5 Input of outbound material information



Outbound material added successfully 1

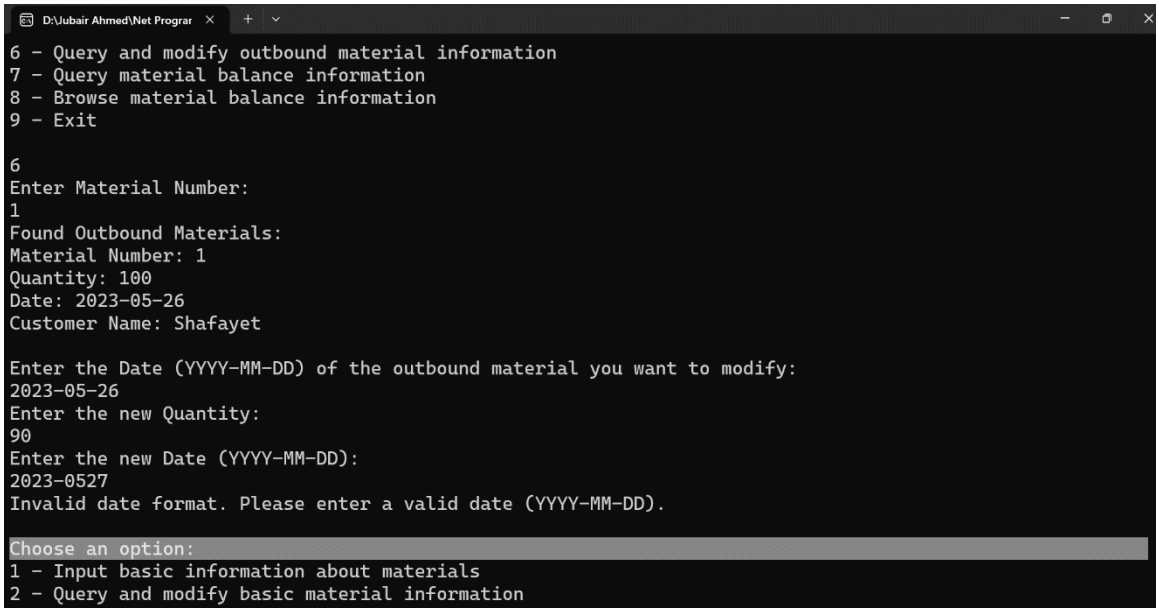


Outbound material added successfully 2



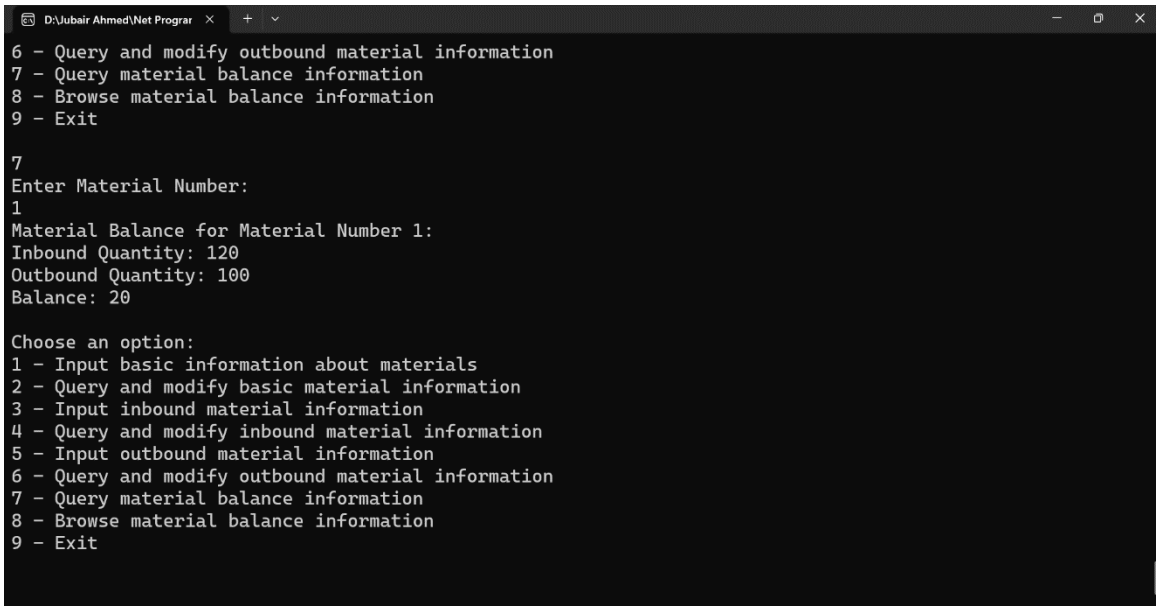
Outbound material added successfully 3

4.6 Query and modification of outbound material information

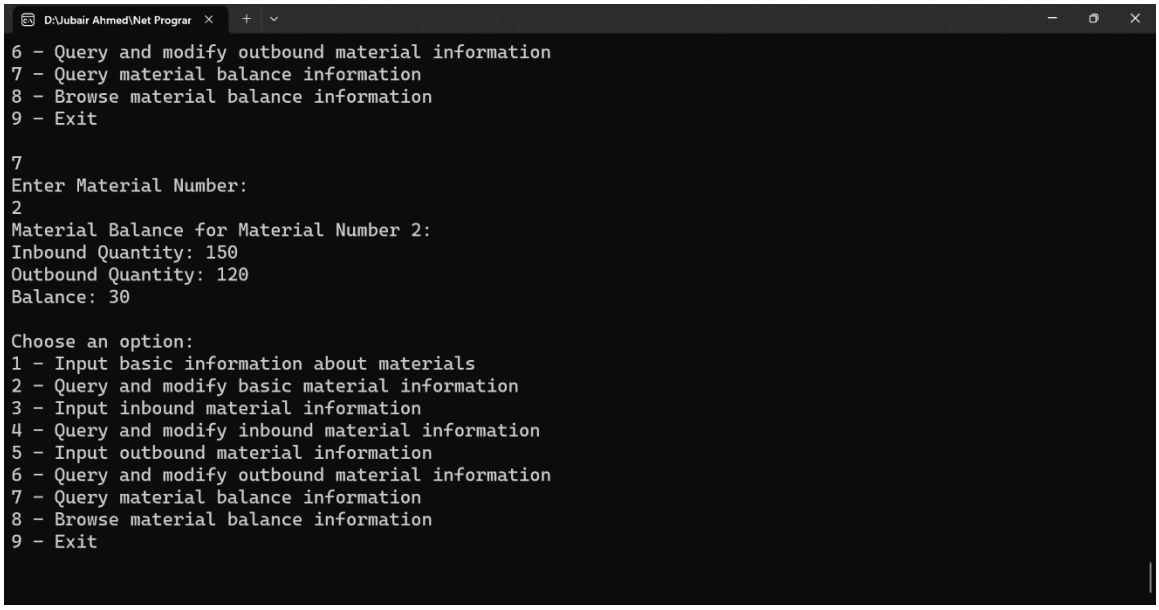


Outbound material queried and modified successfully

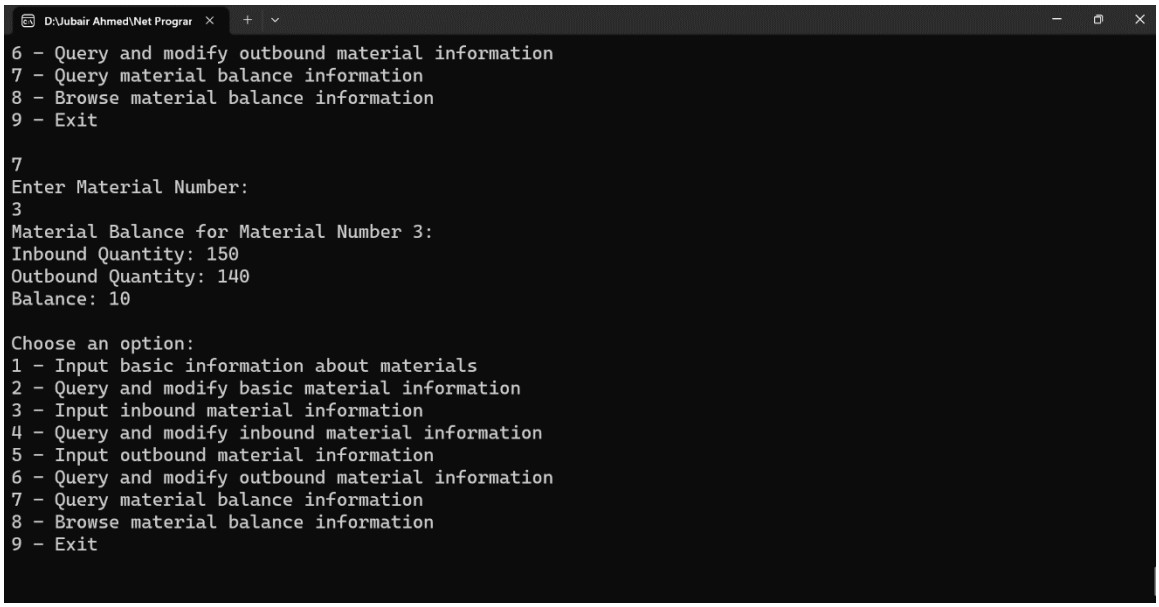
4.7 Query of material balance information



Material balance queried successfully 1

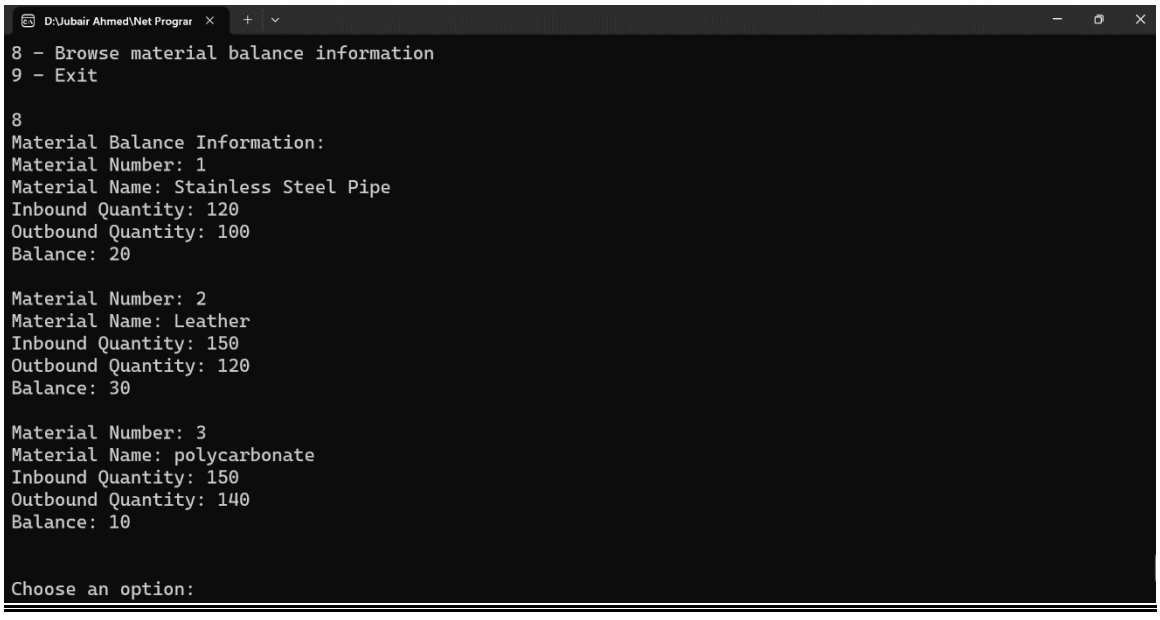


Material balance queried successfully 2

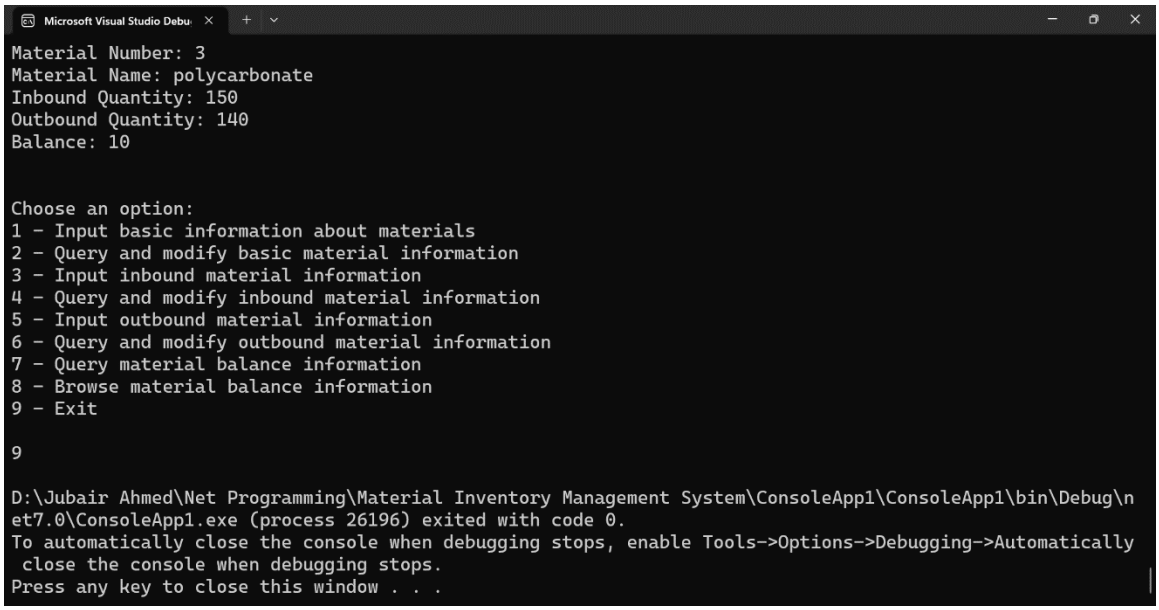


Material balance queried successfully 3

4.8 Browsing of material balance information



4.9 Exit from the system



Exit successful

Chapter 5: Conclusion

Inventory management is a vital component of any business dealing with physical products. It ensures that the right amount of inventory is available when needed, while minimizing expenses and maximizing profits. In our research, we have delved into the concept of Material Inventory Management System and its significance in contemporary business operations.

The key discoveries in this study are as follows:

- Material Inventory Management System involves the utilization of software and hardware tools to oversee inventory levels, track products, and monitor supply chain activities.
- The system automates various inventory-related processes such as ordering, receiving, storing, and shipping, thus enhancing efficiency.
- It helps businesses reduce costs associated with excess inventory, prevent stockouts, and improve customer satisfaction.
- Material Inventory Management System can be integrated with other business software systems like Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) to streamline operations.

This paper also provides valuable insights into the distinctive values and alternative perspectives concerning Material Inventory Management System. For instance, it highlights that the system is not a one-size-fits-all solution and its effectiveness depends on factors such as business type, product characteristics, and supply chain complexity. Moreover, the paper emphasizes the significance of data accuracy and the need for continuous improvement and adaptation to changing market conditions.

Furthermore, the paper underscores the relevance of Material Inventory Management System in the current business environment and its future prospects. With rising competition and market uncertainty, businesses must adopt efficient inventory management practices to remain competitive and

profitable. Material Inventory Management System presents a viable solution to achieve this objective. Additionally, advancements in technologies such as Artificial Intelligence (AI), Internet of Things (IoT), and Blockchain are expected to revolutionize inventory management and enhance the capabilities of Material Inventory Management System in the future.

In conclusion, Material Inventory Management System is a crucial tool for businesses aiming to optimize inventory levels, reduce costs, and enhance customer satisfaction. This paper offers a comprehensive overview of the concept's importance in contemporary business operations. The main findings, valuable insights, and relevance of the system to the current and future business environment have been highlighted. It is recommended that businesses explore and adopt Material Inventory Management System to enhance their competitiveness and achieve their strategic goals.