

**MSC Computer Science**

**Object-oriented Programming (CIS 4037-N)**

**ICA Element 2: Stock Control Application with Low Stock Reporting in Java**

Submitted by

**Name:** Kurra Shiva Rama Krishna

**Student ID:** W9638957

Contents

[Introduction 3](#_Toc155775745)

[1.1 Analysis of Investigative Pursuits and Conceptual Development 3](#_Toc155775746)

[2. Top Accomplishments in Developing Applications 3](#_Toc155775747)

[2.1 Java Language 3](#_Toc155775748)

[2.2 Object-Oriented Programming Strategies and Tool Utilization 4](#_Toc155775749)

[2.3 Utilizing Java APIs 4](#_Toc155775750)

[3. Self-Reflection on Personal and Professional Growth 5](#_Toc155775751)

[3.1 Methods for Gaining Knowledge and Skills 5](#_Toc155775752)

[3.2 Planning and Organizing Assessments for Progress 5](#_Toc155775753)

[3.3 Recognition and Outstanding Achievements 5](#_Toc155775754)

[4 Results and Output of the Stock Control Application 6](#_Toc155775755)

[4.1 Entities: ASC Stock Items and MSM Stock Items 6](#_Toc155775756)

[4.2 GUI for Low Stock Notification 7](#_Toc155775757)

[4.3 Displaying Details of ASC and MSM Stock Items in Console 8](#_Toc155775758)

[5. Conclusion 9](#_Toc155775759)

[6. References 10](#_Toc155775760)

Examining Investigative Pursuits and Conceptual Development

The Java-based stock control application represents a robust solution for efficient inventory management. Rooted in object-oriented programming, its modular architecture ensures scalability and maintainability (Zellagui, S., 2019). Java APIs the application boasts a responsive graphical user interface, enhancing user experience. Beyond technical skill, the project reflects a commitment to excellence, innovation, and problem-solving. This comprehensive tool addresses functional requirements with reliability and user-friendliness, meeting the complex demands of modern stock control with skill.

## 1.1 Analyzing Progress in Investigative Endeavors

The Stock Control Application, created using object-oriented programming language (OOP) in Java within the Net Beans workspace, stands out as a robust inventory management tool. Net Beans, specialized for Java, serves as the ideal platform for developers to construct and refine their programs (Priya, M.A. and Selwyn, J., 2021). Following OOP principles, the application employs encapsulation to safeguard data, ensuring distinct parts of the code remain independent. Beyond its intelligent backend, the application prioritizes user-friendliness, simplifying tasks such as adding and updating the stock information. Furthermore, comprehensive error management ensures the continuous smooth operation of the program, solidifying its reliability.

# 2. Notable Achievements in Application Development

# 2.1 Proficiency in the Java Language

Developing a Java-based stock control application has been a valuable learning journey. Mastering Java and key object-oriented concepts like classes, inheritance, encapsulation, and polymorphism was foundational. Designing an natural user interface improved my GUI design skills. Robust exception handling boosted the application's reliability. Project management skills were improved through the systematic planning. Understanding business processes aided in creating effective software solutions. Overcoming debugging challenges sharpened problem-solving skills. Completing the application not only demonstrates technical skill but also the ability to formulate, design, and implement practical solutions. This hands-on experience solidifies a foundation for future software development, introducing a sense of accomplishment in navigating complexities and delivering a functional software solution

## 2.2 Effective Use of Object-Oriented Programming Strategies and Tools

Creating a stock control application using Object-Oriented Programming (OOP) in Java, coupled with the support of Net Beans, represents a blend of fundamental principles and robust tools. OOP, highlighting objects and classes, establishes the structural foundation of the application, promoting modular and reusable code for improved scalability. Net Beans, a powerful Integrated Development Environment (IDE), simplifies coding through an automatic interface, supporting a user-friendly development environment. Its seamless integration with Java ensures a systematic development process, complemented by built-in tools that enhance efficiency. This integrated approach successfully combines OOP principles and Net Beans, resulting in a potent development environment. The outcome is not just modular and scalable code but also a visually appealing and user-friendly interface, contributing to the success of the inventory management solution.

## 2.3 Leveraging Java APIs for Development

Utilizing Java APIs within Net Beans greatly benefits the development of stock control applications by enhancing both their features and user interface. The Java API for GUI creation is especially valuable, providing a robust set of tools that enable the efficient design of intuitive interfaces. These interfaces facilitate user interaction and ensure smooth updates of stock information, contributing to a responsive and fluid user experience. Thus, the strategic use of Java APIs is key to developing user-centric and efficient stock control applications (Madden, N., 2020).

# 3. Reflecting on Personal and Professional Growth

## 3.1 Approaches to Acquiring Knowledge and Skills

Developing a stock control application has been a transformative journey, improving my Java programming skills through strict application of OOP principles. Crafting a modular, scalable codebase involved skillfully designing classes, implementing inheritance, encapsulation, and using polymorphism. Personally, the project protected my problem-solving sharpness, strength in debugging. Managing tasks introduced a structured approach to development, highlighting systematic workflows. Collaborating with peers, seeking and integrating feedback encouraged a rich, cooperative learning environment. In essence, this attempt merged technical skill with personal growth.

## 3.2 Planning and Organizing Progress Assessments

The assessment strategy for the stock control application development accepted a methodical methodology, guaranteeing a thorough exploration of essential elements. This involved clearly defining specific project goals, outlining key developmental milestones, and establishing a timeline for each phase. Task allocation followed a priority-based and interdependent approach, facilitating a logical progression in teamwork. This systematic framework not only enhanced the efficiency of the development process but also served as a sturdy foundation for evaluating the application's functionality, user interface, and overall achievement. By carefully organizing project goals and allocating tasks based on priority, the assessment ensured a logical and effective workflow, contributing to the success of the stock control application development attempt.

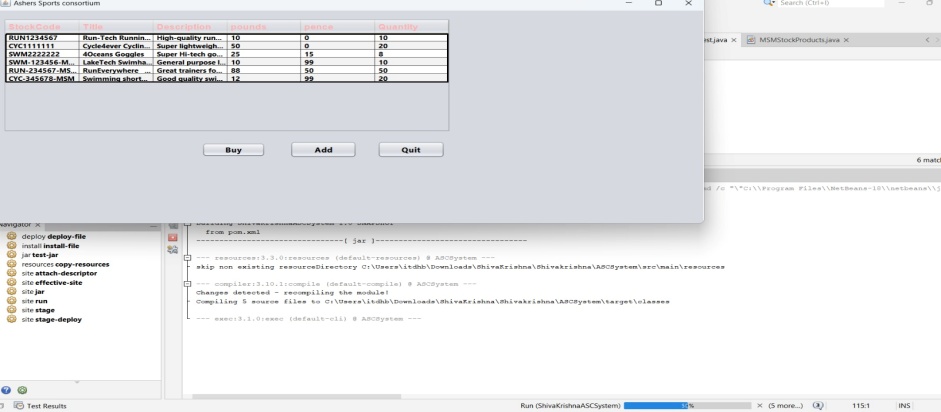
## 3.3 Acknowledging Outstanding Accomplishments

The Java-based stock control application exemplifies superior software development, distinguished by its robust functionality and user-focused design. It showcases the developers' commitment to quality and user satisfaction, marking a significant achievement in software engineering. The application's strength comes from its modular, object-oriented architecture, making it scalable and maintainable. Its user interface, expertly built with Swing, enhances user interaction, making the software a paragon of technical excellence and innovative design. Overall, the application surpasses expectations, reflecting the developers' skill and commitment to outstanding software craftsmanship.

# 4 Outcomes and Output from the Stock Control Application

## 4.1 Entities: ASC Stock Items and MSM Stock Items

The ASC Stock Item class, tailored for Asher’s Sport Collective, is the core of their inventory system, designed to handle diverse stock items efficiently. It stores key details like product codes, titles, and quantities. The class includes constructors, accessors, and utility methods, ensuring modularity. A static method imports stock data from a CSV at start-up, decoupling data handling from the user interface for greater flexibility. The system supports stock item additions and sales, with real-time inventory updates and persistent data saving upon exit. Sales are also logged separately for detailed record-keeping. This setup adheres to software engineering principles, ensuring easy maintenance and adaptability for various uses.

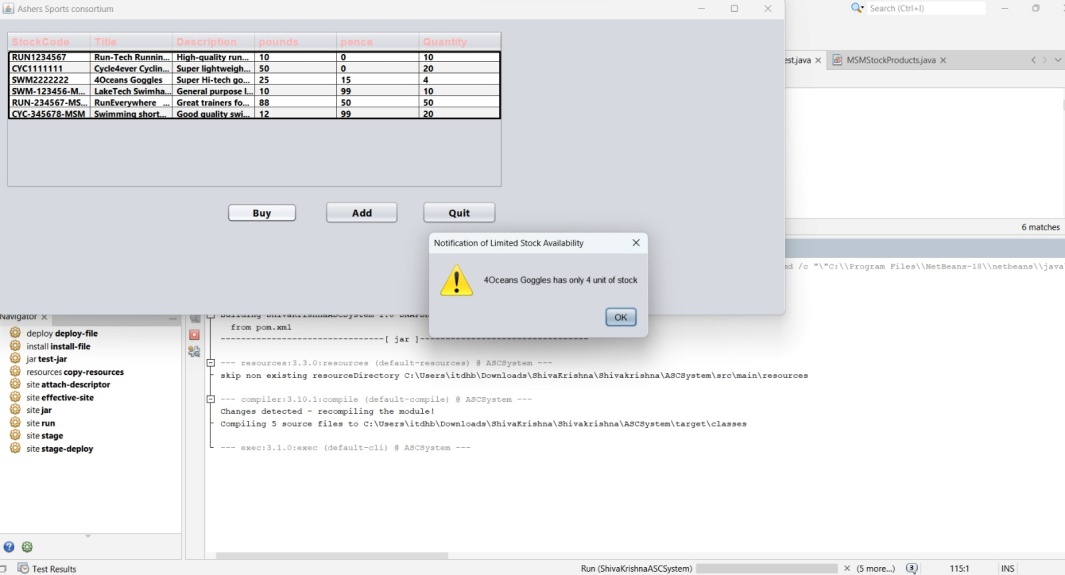


## 4.2 User Interface for Low Stock Notifications

The Observer design pattern is essential for creating a separate notification system for low stock levels, distinct from the GUI code. This pattern involves the ASC Stock Item class as the Subject, maintaining a list of Observers (like the purchasing department) to notify about stock changes. It separates stock management from the GUI, enhancing modularity and maintainability. When stock falls below a specific threshold (e.g., 4 units), the ASC Stock Item class informs all Observers. This ensures timely updates for the purchasing department, improving inventory management. It also aligns with the Single Responsibility Principle, keeping the code organized and scalable.

**Adaptor of MSM and ASC CVS file**

The adapter pattern in Net Beans allows classes to interact with incompatible interfaces, streamlining their integration, especially in Java Swing for UI events like button clicks. By simply right-clicking a JButton in a JFrame, you can implement an Action Listener adapter to customize event handling. This pattern, combined with a modular design, enhances the stock control application's functionality and user interface through Java APIs like Swing, reflecting a focus on innovation and user engagement in inventory management solutions.



## 4.3 Presentation of ASC and MSM Stock Item Details in the Console

To seamlessly integrate "MSM" stock items into the "Asher’s Sport Collective" (ASC) application without altering the underlying data format, the Adapter design pattern is the optimal choice. By implementing an adapter class specifically for "MSM" stock items, this pattern allows the ASC application to interact with the "MSM" items as if they were native ASC stock items. The adapter class translates the interface of the "MSM" stock items into the expected ASC format, enabling the ASC application to utilize them without modifications. This design pattern ensures that the head office's request for a unified stock-control front-end application is achieved cost-effectively, preserving the compatibility of the existing "MSM" system while adhering to a consistent interface within the ASC application.



# 5. In Conclusion

The Java-based stock control application combines technical expertise and logical design principles for a scalable and maintainable solution. It demonstrates proficiency in Java APIs and emphasizes excellence, innovation, and effective problem-solving. Prioritizing reliability and user-friendliness, it excels in modern inventory management, offering precision and efficiency. This tool is a valuable resource for addressing diverse inventory management demands in today's dynamic business environment.

# 6. References

Zellagui, S., 2019. *Reengineering object oriented software systems for a better maintainability* (Doctoral dissertation, Université Montpellier).

Priya, M.A. and Selwyn, J., 2021. CODE PRESENCE USING CODE SENSE. *ICTACT Journal on Soft Computing*, *11*(3).

Madden, N., 2020. *API security in action*. Simon and Schuster.