



MSc Computer Science
School of Computing, Engineering and Digital technologies
Object-oriented Programming
CIS 4037-N

Academic Year: 2023/24

ICA Element 2: Stock Control Application with low stock reporting

10 January 2024

Table of Contents

1. [Introduction](#)
2. [Project Aim: Stock Control Application with Low Stock Reporting](#)
 - [2.1 Planning](#)
 - [2.2 Evaluation](#)
 - [2.3 Development](#)
 - [2.4 Testing](#)
 - [2.5 User-Friendly Interface](#)
 - [2.6 Sales Transaction Recording](#)
 - [2.7 Low Stock Reporting](#)
3. [Research & Methodology](#)
4. [Results and Discussion](#)
5. [Self-Reflection](#)
6. [Conclusion](#)
7. [References](#)

Introduction

This report chronicles my role, contributions, and insights during the development of the Stock Control Application, a pivotal component of the Object-Oriented Programming module's first in-course assessment element. The project's primary objective was to create a robust stock control application with integrated low stock reporting, involving meticulous planning, development, and testing. This encompassed coding, designing a user-friendly interface, and implementing algorithms for stock transactions and reporting. The emphasis was on enhancing the user experience throughout.

Project Aim: Stock Control Application with low stock Reporting

The project's focal point was the creation of a comprehensive Stock Control Application, incorporating meticulous planning, thorough evaluation, systematic development, and rigorous testing. The key components of this aim included:

Planning:

Detailed project planning involved defining essential functionalities, outlining the project scope, and establishing a clear roadmap for development. Identifying critical features, such as product details display and transaction recording, laid the groundwork for subsequent development.

Evaluation:

A thorough evaluation phase involved an in-depth analysis of requirements, aligning application features with effective stock management goals. Researching and selecting appropriate technologies, frameworks, and APIs were crucial steps in optimizing the application's development.

Development:

The development phase focused on translating planned and evaluated features into a functional application. This encompassed coding, designing a user-friendly interface, and implementing algorithms for stock transactions and reporting. The emphasis was on enhancing the user experience throughout.

Testing:

Rigorous testing ensured the reliability, functionality, and security of the application. Unit testing, integration testing, and user acceptance testing were conducted to identify and rectify potential issues. Special attention was given to validating the effectiveness of the low stock reporting mechanism.

User-Friendly Interface:

The primary goal was to create an intuitive and user-friendly interface catering to users with varying technical expertise. Seamless integration of features like product details display, sales transactions, and restocking functionality was prioritized for an efficient user experience.

Sales Transaction Recording:

The application systematically recorded sales transactions, capturing crucial details such as transaction date, product codes, quantity sold, and unit prices. This feature aimed to provide a comprehensive record for inventory management and financial tracking.

Low Stock Reporting:

A pivotal aspect was the application's ability to promptly identify and report low stock levels. The system alerts users when product quantities fall below predefined thresholds, facilitating timely restocking actions. The implementation aimed to prevent stock outs, optimize inventory levels, and enhance operational efficiency.

In essence, the Stock Control Application with low stock reporting served as a holistic solution to challenges associated with effective stock management. The combined efforts in planning, evaluation, development, and testing aimed to deliver a user-friendly and feature-rich application meeting specific inventory control and sales transaction recording needs.

Research & Methodology

Limited knowledge on creating test plans prompted extensive research to fulfill the testing role adequately. Being a developer, research extended to Java features, APIs, and the Swing API. Code explanations were diligently provided, enhancing code readability. Challenges stemmed from limited Java experience, mitigated by leveraging C# proficiency.

Including obtaining progress reports, suggesting a project methodology, and offering support were crucial for project cohesion. In development extensive research was conducted on Java features, APIs, and the Swing API. Code clarity and readability were prioritized through diligent commenting and naming conventions. Challenges were encountered due to limited Java experience, mitigated by leveraging knowledge from C#.

Despite limited knowledge on creating test plans for Java applications, extensive research was undertaken. Unit testing using JUnit-4 ensured the application's expected performance. A proactive approach was taken to gather progress reports and propose a project methodology.

Results and Discussion

Determine which methods needed testing for each method in the Stock and Sales-transaction class required testing. Hence, a test plan was created to document each method tested and detailed Test plans for all Classes.

The essence of Unit tests is to ensure that the application performs as expected. A unit test is a software testing approach where individual units or components of the software are tested (Sneed, 2010). Unit tests were carried out using the JUnit-4 framework. JUnit is an open-source framework used to write and run repeatable automated tests (Wick, Stevenson, and Wagner, 2005a).

Self-Reflection

In my capacity as the sole tester for this individual project, I focused on closely examining the methods' design and classes, providing valuable feedback upon discovering errors. As an independent tester, collaboration with a developer was unnecessary, and I took full responsibility for ensuring the robustness of the application. Due to my limited knowledge in creating test plans for Java applications, I conducted extensive research to acquire the necessary skills.

Taking on the role of the sole developer, I embarked on thorough research into the relevant features of Java, along with exploring Java APIs and the Swing API. Each line of code was meticulously explained through well-documented comments. I adopted a rigorous approach to naming variables, classes, and methods, enhancing code readability. Despite some initial struggles stemming from my limited experience in Java, my background in C# proved beneficial, given its status as an object-oriented programming language.

As this project was an individual undertaking, the role of a team lead was not applicable. The absence of a team necessitated a strong self-directed approach to project management. To ensure progress, I maintained a disciplined approach, obtaining regular progress reports from myself and implementing a personal project methodology. Timely support was consistently provided to myself, acknowledging and addressing specific challenges encountered during different phases of the project.

Conclusion

In summary, this individual project has provided a firsthand glimpse into the dynamics of project development, allowing me to navigate the complexities of working in a team, albeit a metaphorical one. The experience has been instrumental in honing my skills in Java, object-oriented programming, and the utilization of relevant APIs. The insights gained during this crucial stage of application development are poised to be invaluable as I transition into the second element of the in-course assessment, further building upon the foundation laid in this initial endeavor.

Acknowledging that the success of this project rests solely on individual efforts, I extend appreciation to the collective support system that contributed to its accomplishment. This includes the guidance from instructors, the wealth of knowledge drawn from external resources, and the continuous commitment to improvement. This individual venture has not only expanded my technical capabilities but has also fostered a sense of self-reliance and adaptability—qualities vital for future collaborative endeavors in the realm of software development.

References

Sierra, K., & Bates, B. (2005). "SCJP Sun Certified Programmer for Java 5 Study Guide (Exam 310-055)." McGraw-Hill Education. [Online] Available at:

<https://www.mheducation.com/highered/product/scjp-sun-certified-programmer-java-5-study-guide-310-055-sierra-bates/M9780072253604.html>

Fowler, M. (1999). "Refactoring: Improving the Design of Existing Code."

Addison-Wesley. [Online] Available at:

<https://www.oreilly.com/library/view/refactoring-improving-the/9780133065268/>

Oracle. (n.d.). "The Java™ Language Specification, Java SE 8 Edition." [Online]

Available at: <https://docs.oracle.com/javase/specs/jls/se8/html/>

Lea, D. (2000). "Concurrent Programming in Java™: Design Principles and Patterns."

Addison-Wesley. [Online] Available at:

<https://www.oreilly.com/library/view/concurrent-programming-in/0201310090/>

Oracle. (n.d.). "Java Platform, Standard Edition 11 API Specification." [Online] Available

at: <https://docs.oracle.com/en/java/javase/11/docs/api/index.html>

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). "Design Patterns: Elements of Reusable Object-Oriented Software." Addison-Wesley. [Online] Available at:

<https://www.oreilly.com/library/view/design-patterns-elements/0201633612/>

Oracle.(n.d.)."The Java™ Tutorials." [Online] Available at:

<https://docs.oracle.com/javase/tutorial/>

Bloch, J. (2008). "Effective Java (2nd Edition)." Addison-Wesley. [Online] Available at:

<https://www.oreilly.com/library/view/effective-java-2nd/9780137150021/>

Freeman, E., Robson, E., Bates, B., & Sierra, K. (2004). "Head First Design Patterns." O'Reilly Media. [Online] Available at:

<https://www.oreilly.com/library/view/head-first-design/0596007124/>

Oracle. (n.d.). "Java Platform, Standard Edition 8 API Specification." [Online] Available

at: <https://docs.oracle.com/javase/8/docs/api/>