Leopold-Franzens Universität Innsbruck
Department of Business Informatics, Production Management and Logistics

PS Value Adding Processes

Proposal Value Adding Processes

Theoretical Part of Deterministic Models for Lot Sizing

from

Tamino Gaub 12314484
Maximilian Stablum 12312185

Submission Date: 30.11.2023
Supervisor: Assoz.-Prof. Mag. Mag. Stefan Haeussler, PhD

Contents

1 Motivation 3

2 Introduction to Deterministic Lot Sizing Models 3

3 Foundation Models for State-of-the-Art Theories 4

3.1 Economic Order Quantity . . . 4

3.2 Wagner-Whitin . . . 6

3.2.1 Calculation . . . 7

3.2.2 Flow of Wagner-Whitin Algorithm . . . 7

3.3 Theoretical Models Prospection . . . 8

3.4 Current Innovations in Lot Sizing Techniques . . . 9

4 Project Development Framework: Methodical Toolbox and Project Planning 9

4.1 Methodical Toolbox . . . 9

4.2 Backend . . . 10

4.3 Apache Maven . . . 10

4.4 Spring Initializr . . . 10

4.5 REpresentational State Transfer . . . 11

4.6 Frontend . . . 11

4.7 Postman . . . 11

4.8 Swagger . . . 12

5 Expected Results of Application of Deterministic Models for Lot Sizing 12

Acronyms

API Application Programming Interface . . . 11

EOQ Economic Order Quantity . . . 4

HTML Hypertext Markup Language . . . 11

HTTP Hypertext Transfer Protocol . . . 11

IDE Integrated Development Environment . . . 9

MRP Materials Requirement Planning . . . 7

MINLP Mixed-Integer Nonlinear Programming . . . 9

REST REpresentational State Transfer . . . 11

UI User Interface . . . 11

1 Motivation

In today's business environment, characterized by extremely narrow profit margins, increasingly stringent customer expectations for product quality, and the pressure of reduced lead times, the competitive landscape has become more intense than ever [1]. As a result, companies are compelled to seize every opportunity to enhance and optimize their business processes. Deterministic models for lot sizing play a crucial role in production and inventory management, particularly in environments where demand can be accurately predicted [2]. These models are essential for optimizing production schedules, minimizing costs, and ensuring timely fulfilment of customer orders.

By optimizing lot sizes, companies can reduce excess stock and associated holding costs, thereby enhancing overall operational efficiency. The models are a crucial subject for modern businesses as they continue to grapple with the challenge of determining the ideal lot size. This topic is particularly relevant in today's competitive market, where effective resource allocation and cost management are key to achieving business success. The decision problem consists of finding the best trade-off between setup costs and holding costs.

2 Introduction to Deterministic Lot Sizing Models

In deterministic models, demand is assumed to be predictable and constant throughout the planning horizon, and must be fulfilled either through immediate orders or existing inventory [2]. These models specifically address the coordinated lot-size problem, which involves determining a timetable for replenishment [2, 3]. The mathematical complexity of the coordinated lot sizing problem is NP-complete, suggesting that it is unlikely that a polynomial bound algorithm can be found for its solution. For this reason, an extensive literature base is rapidly developing that describes alternative mathematical formulations and exact solution approaches for the problem to solve large industrial problems that can involve over a hundred items and time periods.

The models provide a foundation for optimizing inventory levels and production schedules, reducing costs, and improving efficiency. In particular, the coordinated lot sizing problem exemplifies the complexity and practical challenges faced in this field. The problem's NP-completeness underscores the difficulty in finding efficient, exact solutions, driving research towards innovative mathematical models and solution methods.

Despite these advances, there are remaining significant challenges and opportunities for future research in deterministic lot sizing. The complexity of the problem necessitates ongoing efforts to develop more efficient and effective solution methods. These include both exact algorithms, which guarantee optimal solutions, and heuristic approaches, which provide good solutions in a reasonable timeframe.

Furthermore, the integration of deterministic models with other planning tools and technologies, such as demand forecasting and real-time data analytics, presents a promising avenue for enhancing their applicability and effectiveness in dynamic business environments.

3 Foundation Models for State-of-the-Art Theories

Production planning and scheduling pose significant challenges for management, constituting a hierarchical process encompassing long-term, medium-term, and short-term decisions [4]. The primary focus of addressing this category of issues lies in determining production lots for multiple items across a planning horizon, whether finite or infinite [5]. The objective is to minimize setup costs, inventory holding costs, production costs, and backlogging costs, all while ensuring the fulfilment of known demand. Notably, there is an absence of interdependency between items, owing to the lack of capacity constraints and parent component relationships. Consequently, decisions regarding lot sizing can be independently made for each item. This section delves into the formulations and solution procedures for two significant incapacitated lot sizing models, namely the Economic Order Quantity (EOQ) model (subsection 3.1) and the Wagner-Whitin model (subsection 3.2). While the EOQ and Wagner-Whitin algorithms may not be considered state-of-the-art in modern operations research, they persist as valuable benchmarks in the field [6]. Their enduring relevance lies in their widespread usage as reference points, providing a basis for comparison and evaluation when assessing the efficacy of more contemporary and sophisticated lot sizing models.

3.1 Economic Order Quantity

The EOQ model was initially introduced in 1913 by Harris with the statement: "How Many Parts to Make at Once" [7]. It can be seen as one of the earliest applications of mathematical modelling to scientific management. Since then, it has been a remarkable benchmark in the literature and is still widely used [8]. The EOQ model, formulated by Harris, represents a seminal contribution to inventory management and remains a fundamental concept in operations research and supply chain management [7, 8]. The EOQ model addresses the critical question of determining the optimal order quantity for minimizing total inventory costs, considering both holding costs and ordering costs.

In general, some assumptions have to be clarified for the EOQ model from Harris [7]. The notation is based on the work by Hopp and Spearman [9, p. 51]. The first assumption is that customer demand for the item is assumed to be known and constant at a rate **D**. Secondly, it should be assumed that for the EOQ model the leadtime is equal to zero, which means that delivery or manufacturing is instantaneous. With this assumption, there is no need to consider when an order should ideally be placed. The answer to this consideration describes the statement that **Q** is ordered every time the stock falls to zero. This can be summarised to the extent that it should be ordered Q when inventory equals a leadtime's supply [10]. In addition, three assumptions must be made regarding costs. The cost of the units themselves, denoted **c** is assumed to be fixed, regardless of the number of units ordered or manufactured. That means that there are no discounts for big purchases or other reductions. Finally, there are fixed manufacturing setup costs, denoted **A** and determined costs for holding items in the inventory, denoted **h**. The setup costs do not change regardless of the amount of products to be produced. Holding item costs h are defined as the product of the annual interest rate (**i**) and the capital invested in inventory (c), denoted by the equation $h = ic$, when the holding cost

primarily comprises interest on the funds associated with inventory.

In the context of the EOQ model, the objective is to determine the order quantity (Q), which minimizes the average cost or time associated with inventory management arising from the act of placing orders for quantity Q whenever the inventory depletes to zero. In the following, the average cost arising from the order quantity Q is represented as **AC(Q)**. The decision variable of the entire EOQ model is therefore the unknown order quantity (Q). Assuming constant and deterministic demand, ordering Q units each time the inventory reaches zero yields an average inventory level of $Q/2$, as depicted in Figure 1. The annual holding costs can be described as

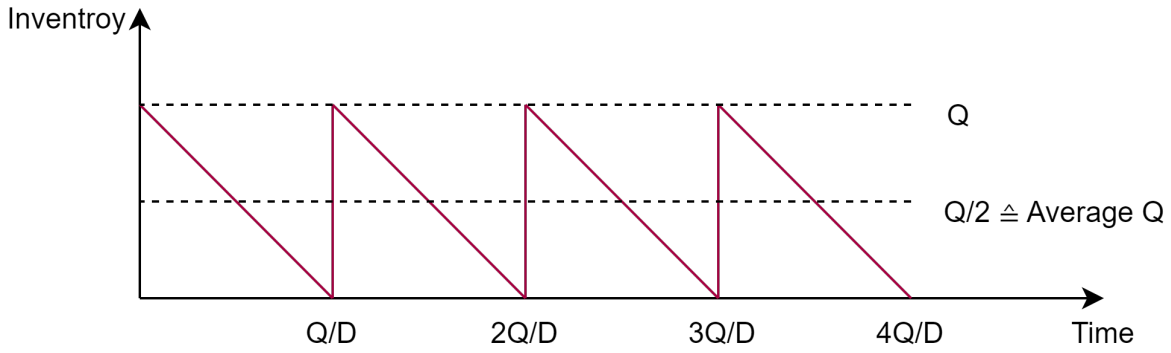

Figure 1: Inventory versus time in the Economic Order Quantity model [9].

$hQ/2$, whereby the unit holding costs can be calculated by $hQ/2D$. The setup costs has been already defined before to A, from which the unit setup costs can be calculated with A/Q. If the annual holding costs are now added to the unit setup costs and the production costs already defined above, the following cost function is obtained:

$$Y(Q) = \frac{hQ}{2D} + \frac{A}{Q} + c \tag{1}$$

Now that the cost function has been defined, the next goal is to minimise it. Thus, determining Q involves balancing the average ordering cost against the average inventory-holding cost. Observe in Figure 2 that the minimum value of AC(Q) happens where the graphs of average annual ordering costs and average annual inventory-holding intersect, specifically at $A*(D/Q) = h*(Q/2)$ [10]. Consequently, we can identify the optimal Q, represented as $Q*$, by solving Equation 2 for $Q*$.

$$A \times \frac{D}{Q*} = h \times \frac{Q*}{2} \tag{2}$$

Solving this formula gives the following solution:

$$Q* = \sqrt{\frac{2 \times A \times D}{h}} \tag{3}$$

Equation 3 represent the final EOQ Square Root Formula, which is also referred to as the economic lot size [9, p. 53].

The so-called sensitivity analysis, which can be taken from [9, pp. 55–57], will be discussed at this point in a later version of the report.

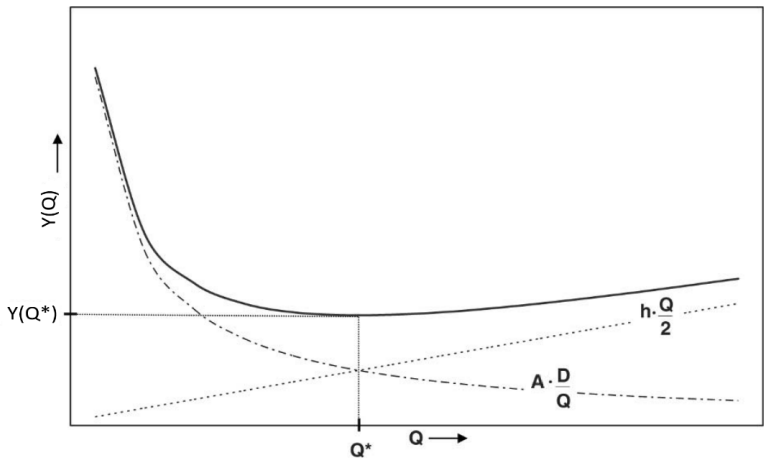

Figure 2: Costs in the Economic Order Quantity model. Based on [10, p. 141].

The EOQ model provides valuable insights into inventory management strategies, aiding businesses in achieving cost-efficiency, improving cash flow, and enhancing overall operational performance. Over the years, the EOQ model has evolved and adapted to various business environments, remaining a cornerstone in the design and optimization of inventory systems across diverse industries.

3.2 Wagner-Whitin

The Wagner-Whitin model represents a further development in inventory management for moving beyond the static assumptions of earlier models like EOQ in subsection 3.1 [11]. Developed by Wagner and Whitin in 1958, this model is a dynamic version of the economic lot size model, addressing variable demand over a finite planning horizon. This dynamic model addresses the complexities of varying demand over a finite planning horizon, which is a critical aspect in inventory management practices [9]. Unlike the EOQ model, which assumes a constant demand rate, the Wagner-Whitin model divides the planning period into several discrete periods, acknowledging that demand can fluctuate significantly over time.

In mathematical terms, the Wagner-Whitin model can be viewed as a series of decisions for each period, where the decision to produce or is based on the trade-off between setup costs and holding costs. These decisions are made in the context of varying demand forecasts for each period, emphasizing the need for flexibility in inventory management. An important insight is that if production occurs in period t (including setup costs) to meet the demand of period t + 1, then it would not be cost-effective to also produce in period t + 1, as this would entail an additional setup cost. It has to be cheaper to produce the products of period t and t + 1 in the first period. It is not economical to schedule production of each product in every individual period.

A unique feature of this model is its expression as the shortest path problem. This perspective enables efficient computational strategies to find optimal solutions, making the model not only theoretically grounded but also practically applicable in various inventory management scenarios.Wagner and Whitin's model deals with the problem of determining production lot sizes when

demand is deterministic but time-varying and all other assumptions of the EOQ model are valid. Furthermore, the Wagner-Whitin model's dynamic approach to lot sizing has been influential in both academic literature on production control and the practical development of Materials Requirement Planning (MRP) systems [9]

3.2.1 Calculation

In order to understand the calculation of the Wagner-Whitin model, the following notations must first be defined [11]:

- $f_t(I)$ - the minimized cost function of a period t with a given inventory I in the beginning.
- i_{t-1} - the inventory cost per unit, if the product is stored until period t.
- x_t - manufactured amount in period t.
- s_t - setup costs for manufacturing in period t.
- d_t - the demand in period t.
- $\delta(x_t)$ - is an indicator function that is used to determine whether in a given period t an order is placed or not.
  - $\delta(x_t) = 1$, if $x_t > 0$, which means that there is production in period t, and therefore setup costs are incurred.
  - $\delta(x_t) = 0$, if $x_t = 0$, which means that there is no production in the period t, and hence no setup costs are incurred.

The Wagner-Whitin algorithm proceeds to calculate the optimal production schedule across the planning horizon. It does this by evaluating each period on its own, determining whether it is more cost-effective to produce in the current period or in a future period. Followed is the formula which is needed to calculate the optimal lot size:

$$f_t(I) = \min[i_{t-1}I + \delta(x_t)s_t + f_{t+1}(I + x_t - d_t)] \tag{4}$$

3.2.2 Flow of Wagner-Whitin Algorithm

The Wagner-Whitin algorithm advances sequentially from the first period to the final period, denoted as period T [9]. As the model progresses through each period, the algorithm evaluates the cost-effectiveness of producing in the current period versus a future period, accounting for the trade-offs between setup costs and holding costs. An optimal approach is characterized by each value of $\delta(x_t)$ precisely matching the sum of a series of future demands d_t. Followed, the possible solutions for the algorithm will be listed [9]:

$$\begin{array}{llllll}
\delta(x_1) = d_1 & \text{or} & \delta(x_1) = d_1 + d_2 & \text{or} & \delta(x_1) = d_1 + d_2 + ... + d_T \\
\delta(x_2) = 0 & \text{or} & \delta(x_2) = d_2 + d_3 & \text{or} & \delta(x_1) = d_1 + d_2 + ... + d_T \\
\vdots & & & & \\
\delta(x_T) = 0 & \text{or} & >\delta(x_2) = d_T & &
\end{array}$$

An exact requirement's management policy is fully specified by indicating the time periods in which the order should be placed. The number of exact requirements policies is much smaller than the total number of realizable policies. The sequential process is in Figure 3 displayed. The

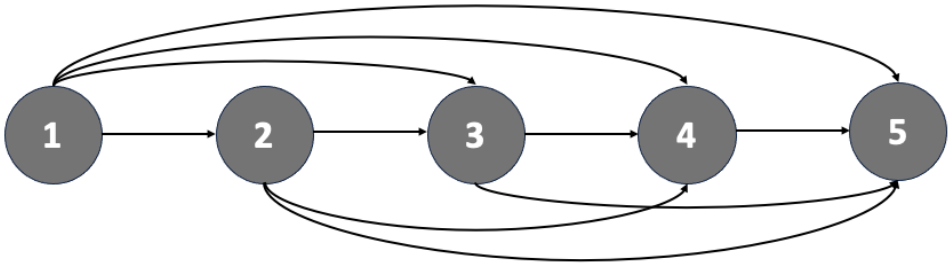

Figure 3: Combination representation for Wagner-Whitin. Based on [12].

Wagner-Whitin algorithm conducts a forward calculation for each period t. In every period, it evaluates the potential production options, starting from the last production period up to the current period t. For each period, the algorithm assesses setup costs and holding costs.

If the most favourable option within the examined scope is to adhere to the existing production plan and maintain the last production date, this option is selected. Conversely, if initiating production in a more recent period proves to be more cost-effective, the management will opt for this alternative. When the production plan is updated, the algorithm, in its subsequent phase, will consider only two potential production periods.

This decision-making process is grounded in the principle of cost minimization, aiming to optimize the total costs across the entire planning horizon.

3.3 Theoretical Models Prospection

The EOQ model and the Wagner-Whitin algorithm have long been stalwarts in the realm of deterministic models for lot sizing, providing a solid foundation for research studies in inventory management [6]. Glock, Grosse, and Ries continue that various extensions of Harris lot size model have been developed over the years, including models that incorporate multi-stage inventory systems, incentives, and productivity issues [6, 7]. Despite the evolving landscape of supply chain dynamics and the advent of more sophisticated models, these classic frameworks continue to captivate the attention of researchers for several compelling reasons.

Its simplicity and intuitive appeal make it an attractive starting point for scholars exploring lot sizing problems. Researchers often use the EOQ model as a benchmark or baseline against which they can compare the performance of more intricate models, helping to assess the practical significance and real-world applicability of newer methodologies [6].

On the other hand, the Wagner-Whitin algorithm extends the scope of lot sizing models by considering production capacity constraints and dynamic demand patterns. This algorithm, building upon the EOQ principles, introduces a finite production rate and allows for backlogging, thus capturing more realistic scenarios. Its enduring relevance lies in its ability to address complexities beyond the scope of the original EOQ model, making it a valuable tool for researchers delving into more nuanced lot sizing problems.

Various extensions of Harris lot size model have been developed over the years, including models that incorporate multi-stage inventory systems, incentives, and productivity issues [6]. In summary, the enduring appeal of the EOQ model and the Wagner-Whitin algorithm in research studies on deterministic lot sizing models can be attributed to their simplicity, practicality, and relevance in addressing real-world inventory challenges. These models serve as essential building blocks, providing a solid starting point for researchers to explore and develop more sophisticated approaches in the ever-evolving field of supply chain management.

3.4 Current Innovations in Lot Sizing Techniques

Lot sizing approaches nowadays are focusing on a more advanced problem which isn't capable with EOQ or Wagner-Whitin. This shift towards advanced models is driven by the increasing complexity of manufacturing environments, where hybrid flow shop systems are becoming more prevalent. The publication "Reformulation, linearization, and a hybrid iterated local search algorithm for economic lot-sizing and sequencing in hybrid flow shop problems" is researching a new model of solving the lot sizing problem [13]. Their study introduces a novel Mixed-Integer Nonlinear Programming (MINLP) model, addressing the unique challenges posed by these hybrid systems. This is a production system featuring multiple parallel machines at one or more stages of the production process, enhancing flexibility and efficiency in handling diverse product types and processing requirements. Their approach integrates economic lot sizing with production sequencing, optimizing both inventory levels and production flow. The MINLP model improves an existing model, and presents a more manageable technique. The research demonstrates significant improvements in solvability and optimality gaps over existing models and algorithms, particularly for large-size instances, highlighting a state-of-the-art solution in lot sizing models.

4 Project Development Framework: Methodical Toolbox and Project Planning

The goal of this project is to develop an application that provides a web-based frontend for calculating the EOQ and the Wagner-Whitin algorithm. This frontend will transfer data to a backend, which handles the actual computation.

4.1 Methodical Toolbox

To implement such a program with front- and backend, a couple of different software is needed. Beside, a decision for specific programming languages fitting Integrated Development Environment (IDE) has to be chosen. When the decision for the programming languages is done, specific middleware which can provide an interface between front- and backend has to be chosen. The selected software is described below.

4.2 Backend

Java[1], a versatile and widely-used programming language, is renowned for its platform independence, robustness, and scalability. Its object-oriented nature and extensive standard libraries contribute to the language's popularity for building diverse applications, ranging from web and mobile to enterprise systems. Implementing EOQ and Wagner-Whitin models in Java make sense due to the language's platform independence, robust numerical computation capabilities, and extensive libraries, providing a scalable and efficient solution for inventory management that can be easily maintained and integrated across diverse systems.

As IDE the decision was made for IntelliJ IDEA[2], which describes themselves as leading Java IDE. IntelliJ IDEA is a comprehensive suite of tools and features tailored for Java programming, streamlining the development process by offering advanced code editing, syntax highlighting, and code completion, thereby enhancing coding accuracy and efficiency. Additionally, it offers robust debugging capabilities, allowing the developers to identify and rectify errors swiftly. IntelliJ IDEA provide real-time error checking, breakpoints, and step-through debugging, facilitating a more systematic and time-effective approach to identifying and fixing bugs in Java code.

4.3 Apache Maven

Apache Maven is a software project management tool, which relieves the software development process and also unifies the development procedure [14]. The foundation about Maven is the Project Object Model. This *pom.xml* is a file inside the dedicated project. Apache Maven will be utilized for its superior project management and build automation capabilities. By simplifying the build process and ensuring consistent project structure, it will play a crucial role in maintaining the integrity and efficiency of the development workflow. Maven is normally used in the command line, but IntelliJ IDEA supports a fully-functional integration inside the project [15].

4.4 Spring Initializr

Spring Initializr is a convenient, web-based tool designed to streamline the process of setting up a new Spring Boot project [14]. A Spring Boot project is essentially an application that simplifies the development process by providing a framework for creating stand-alone, production-grade applications quickly and with minimal configuration. The Initializr offers a user-friendly interface where developers can quickly generate and download a basic project structure, tailored to their specific needs. By selecting desired dependencies, Java version, and build tools like Maven, developers can create a ready-to-use Spring Boot project with minimal setup. This initial setup includes all the necessary configurations and dependencies, allowing developers to immediately start working on the business logic of their application, rather than spending time on initial project setup and configuration. Spring Initializr acts as an essential starting point for building robust, efficient Spring Boot applications, significantly reducing initial development time and complexity.

[1] https://www.java.com/en/ accessed on 27.11.2023
[2] https://www.jetbrains.com/idea/ accessed on 27.11.2023

4.5 REpresentational State Transfer

REpresentational State Transfer (REST) is an architectural style in web services development and was conceptualized by Roy Fielding [16]. It operates under a set of principles emphasizing simplicity and scalability in networked applications, closely intertwining with the concept of an Application Programming Interface (API) [14]. An API acts as a conduit for communication between different software components, often dictating the rules and methods for data exchange. This allows the application to efficiently handle requests and transmit data in a format that is both developer-friendly and optimized for performance. In RESTs context, the API is the medium through which clients and servers interact, typically via Hypertext Transfer Protocol (HTTP) protocols. REST is characterized by its stateless nature, meaning the server does not retain client state information, thus ensuring that each interaction is independent and self-contained. This principle of REST contributes to a distinct separation between client and server, enhancing both the independence and scalability of the system, which aligns seamlessly with the fundamental role of an API in facilitating flexible and efficient client-server communication.

4.6 Frontend

Now that the backend and the appropriate interface, which is described in subsection 4.5, have been defined, the technical conditions for the frontend are described below. The calculations from the backend should be displayed to the end user using a website. Input and output should therefore be possible for the user on this website. The plan is to develop a website based on Hypertext Markup Language (HTML)[3]. In order to display content more dynamically, React.js in particular is to be used, which in turn is based on JavaScript, an HTML extension.

In the realm of frontend development, choosing React.js[4] emerges as a strategic decision driven by its distinctive advantages. React's declarative syntax empowers developers to concisely articulate User Interface (UI) components, facilitating a more straightforward and intuitive development process. The component-based architecture promotes code modularity, enabling the creation of reusable and maintainable building blocks for complex user interfaces. React's Virtual Document Object Model implementation optimizes performance by minimizing unnecessary re-rendering, ensuring swift and efficient updates. With an extensive and collaborative community, React.js offers a wealth of resources, support, and pre-built components, expediting development and troubleshooting. Backed by Facebook, React.js not only exemplifies reliability but also aligns with industry best practices, making it a compelling choice for developers aiming to streamline and elevate their frontend projects. The whole architecture from subsection 4.2 until here can be seen in Figure 4.

4.7 Postman

Postman is an application which offers multiple tools to test REST-API [14]. Notably, it simplifies the process of API testing by allowing users to create and save collections of requests, which can be reused and shared among team members. It offers a user-friendly interface for

[3] https://html.spec.whatwg.org/ accessed on 28.11.2023
[4] https://react.dev/ accessed on 29.11.2023

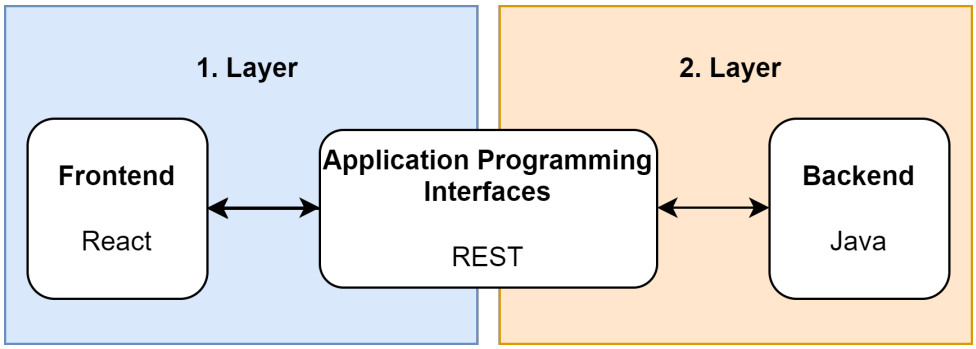

Figure 4: Software architecture.

sending requests to web servers and viewing responses [16]. This feature is particularly useful for demonstrating API functionality to stakeholders who may not be technically adept. Additionally, Postman supports various types of HTTP requests, authentication methods, and has an extensive documentation. Its built-in tools for testing and validating API responses ensure that APIs meet their intended specifications. Furthermore, Postman facilitates environment management, enabling users to develop and test their APIs in different configurations, thereby enhancing the robustness and reliability of API services. The application also integrates with continuous integration and continuous deployment pipelines, automating the API testing process in different stages of software development.

4.8 Swagger

Swagger is a pivotal tool in software development, primarily used for designing, documenting, and testing REST-APIs [17]. It serves as a bridge between various teams involved in software development, including developers, testers, and business analysts. This application enhances understanding among stakeholders without requiring deep dives into source code. Moreover, its intuitive interface simplifies the process of creating and managing API endpoints, making it accessible even to those with limited technical expertise. Additionally, Swagger UI offers an interactive platform for executing API requests and visualizing responses, aiding in debugging and demonstration. The tool's capacity to facilitate the integration of APIs into various applications through its support for multiple programming languages significantly streamlines the development process. Additionally, Swagger's broad compatibility with a range of platforms and programming languages amplifies its adaptability in a variety of development contexts.

5 Expected Results of Application of Deterministic Models for Lot Sizing

The expected outcome of this project is the development of an advanced web-based application tailored for businesses to effectively calculate the EOQ and implement the Wagner-Whitin algorithm. The application let the users decide which of the calculation approaches they want to, use. Based on this selection, the UI changes. This flexibility ensures that businesses of various sizes and with different operational complexities can benefit from the tool. The dynamic nature

of the UI enhances user engagement and provides a tailored experience for different calculation methods. The data will be transferred with REST-API from the frontend to the backend and vice versa.

This tool is designed to offer an easy-to-use application for inventory management and lot sizing. With its user-friendly interface, the application allows users to input data effortlessly and receive accurate, optimized calculations swiftly. Its development is expected to bridge the gap between complex inventory management theories and practical, real-world application. The integration of robust frontend and backend technologies ensures a seamless user experience and reliable results. This application will be beneficial in a business context, aiding companies in making strategic decisions about inventory management, leading to significant cost savings, improved resource allocation, and enhanced operational efficiency.

References

[1] G. De Souza Amaro, D. J. Fiorotto, and W. A. De Oliveira. "Impact analysis of flexibility on the integrated lot sizing and supplier selection problem". In: *TOP* 31.1 (July 2022), pp. 236–266. DOI: 10.1007/s11750-022-00636-2.

[2] P. Robinson, A. Narayanan, and F. Sahin. "Coordinated deterministic dynamic demand lot-sizing problem: A review of models and algorithms". In: *Omega* 37.1 (2009), pp. 3–15. ISSN: 0305-0483. DOI: https://doi.org/10.1016/j.omega.2006.11.004.

[3] E. Farhi et al. "A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem". eng. In: *Science (New York, N.Y.)* 292.5516 (2001), pp. 472–475. ISSN: 0036-8075. DOI: 10.1126/science.1057726. eprint: 11313487.

[4] A. Drexl and A. Kimms. "Lot sizing and scheduling — Survey and extensions". In: *European Journal of Operational Research* 99.2 (1997). PII: S0377221797000301, pp. 221–235. ISSN: 03772217. DOI: 10.1016/S0377-2217(97)00030-1.

[5] M. Salomon. *Deterministic lotsizing models for production planning*. Vol. 355. Lecture notes in economics and mathematical systems. Berlin: Springer, 1991. VII, 158. ISBN: 978-3-540-53701-4.

[6] C. H. Glock, E. H. Grosse, and J. M. Ries. "The lot sizing problem: A tertiary study". In: *International Journal of Production Economics* 155 (2014). PII: S0925527313005689, pp. 39–51. ISSN: 0925-5273. DOI: 10.1016/j.ijpe.2013.12.009.

[7] F. W. Harris. "How Many Parts to Make at Once". In: *Operations Research* 38.6 (1990). (reprint from Factory - The Magazine of Management 10 (1913) 135-136, 152), pp. 947–950. ISSN: 0030-364X. DOI: 10.1287/opre.38.6.947.

[8] D. Erlenkotter. "Ford Whitman Harris and the Economic Order Quantity Model". In: *Operations Research* 38.6 (1990), pp. 937–946. ISSN: 0030-364X. DOI: 10.1287/opre.38.6.937.

[9] W. J. Hopp and M. L. Spearman. *Factory physics*. 3. ed. Long Grove, Ill.: Waveland Press, 2011. XXV, 720. ISBN: 1-57766-739-5.

[10] L. B. Schwarz. "The Economic Order-Quantity (EOQ) Model". In: *Building Intuition*. Ed. by F. S. Hillier, D. Chhajed, and T. J. Lowe. Vol. 115. International Series in Operations Research & Management Science. Boston, MA: Springer US, 2008, pp. 135–154. ISBN: 978-0-387-73698-3. DOI: 10.1007/978-0-387-73699-0_8.

[11] H. M. Wagner and T. Whitin. "Dynamic version of the economic lot size model". In: *Management Science* 5 (1958), pp. 89–96.

[12] S. Nahmias and T. L. Olsen. *Production and Operations Analysis*. 7th ed. Waveland Press, Inc., 2015, pp. 485–489. ISBN: 9781478623069.

[13] H. Zohali et al. "Reformulation, linearization, and a hybrid iterated local search algorithm for economic lot-sizing and sequencing in hybrid flow shop problems". In: *Computers & Operations Research* 104 (2019), pp. 127–138.

[14] J. Hinkula. *Full Stack Development with Spring Boot and React: Build modern and scalable web applications using the power of Java and React*. Vol. 3. May 2022. ISBN: 9781801818643.

[15] JetBrains s.r.o. *Maven | IntelliJ IDEA*. Sept. 7, 2023. URL: https://www.jetbrains.com/help/idea/maven-support.html (visited on 11/27/2023).

[16] A. Martin-Lopez, S. Segura, and A. Ruiz-Cortés. "Online Testing of RESTful APIs: Promises and Challenges". In: ESEC/FSE 2022. New York, NY, USA: Association for Computing Machinery, 2022, pp. 408–420. ISBN: 9781450394130. DOI: 10.1145/3540250.3549144.

[17] H. Wu et al. "Combinatorial Testing of RESTful APIs". In: New York, NY, USA: Association for Computing Machinery, 2022, pp. 426–437. DOI: 10.1145/3510003.3510151.