BUS TRANSPORTATION MANAGEMENT SYSTEM LOGISTIC I: DEVELOPING EFFICIENT INVENTORY AND WAREHOUSE MANAGEMENT, AUDIT MANAGEMENT, AND INFRASTRUCTURE MAINTENANCE USING REVERSE LOGISTICS

A Capstone

Presented to the Faculty of

The College of Computer Studies

Bestlink College of the Philippines

In Partial Fulfilment

Of the Requirements for the Degree of

Bachelor of Science in Information Technology

RICARDO C. ARON III

ANDY L. BARRANTES

KRISTEL D. CASINILLO

JOHN VINCENT B. DIZON

KENT MARK C. TEJADA

OCTOBER 2024

INTRODUCTION

Background of the Capstone Project

Effective logistics operations are crucial for any business. However, many companies still manage their inventory and warehousing manually. The lack of modern logistics management practices and skills, coupled with outdated logistics information systems present significant challenges to advancing logistics capabilities (Goh & Pinaikul, 1998). Studies have also shown that efficient inventory management practices are essential for minimizing procurement costs. By effectively managing and optimizing inventory levels, these practices ensure that supply chain members are not adversely affected by either excess stock or shortages (Rubel, 2021). These challenges are common across various types of businesses. However, our approach goes beyond addressing just one issue. We aim to tackle multiple problems simultaneously. The project's goal is to resolve inefficiencies and inaccuracies in inventory and warehouse management systems while also addressing other critical issues, such as inadequate planning and forecasting, and the handling of damaged goods during transit. By creating an efficient and sustainable inventory and warehouse management system for bus transportation, enhanced by reverse logistics, and implementing an infrastructure maintenance management system, the developers ensure comprehensive monitoring and management of supplies.

Context and Scope

This project focuses on logistics operations for bus transportation companies, specifically those offering provincial routes. The scope is confined to inventory and warehousing management—namely, overseeing materials and supplies from the point of receipt from suppliers until they exit the facilities. The team expects to encounter the following challenges: dealing with complex business logic which can lead to time constraints, and making the system scalable given our limited knowledge. The project is limited only to inventory and warehousing management, and managing maintenance resources for bus companies' infrastructures. However, areas such as fleet management, managing transportation of supplies and materials, implementation of IoT devices and sensors, and managing or designing the layout of warehouse facilities are outside the scope of this study.

Problem Statement

Logistics operations are critical to the success of bus transportation companies, but they often face significant challenges that can lead to inefficiency, making them costly and complex to manage. These challenges include:

1. Inefficiencies in inventory and warehousing management can lead to excessive stock, requiring more storage space, which increases warehousing and labor costs.
2. Inaccurate inventory reporting and demand forecasting can cause errors in financial reporting that can result in overstatement or understatement of assets and leading to either understocking or overstocking.
3. Lack of management in used parts and materials leads to increased waste, higher operational costs, inefficiencies in resource utilization, potential safety hazards, and lost opportunities for cost savings or revenue from recycling or reusing valuable assets.

Addressing these issues is essential for improving the overall efficiency and cost-effectiveness of logistics operations within bus transportation companies.

Objective and Goals

1. Develop an accurate and up-to-date inventory and warehousing management system with and centralized database.
2. Implement dashboards with real-time inventory tracking
3. Implement reverse logistic in every module to manage returns, waste management and recycling programs.
4. Develop a straightforward audit management system

5. Integrate infrastructure maintenance management system to manage bus transportation facilities like depot, terminal, and offices.

Significance and Relevance

The objective of this project is to enhance the logistics operations of companies, particularly those who offer transportation services. By making new and improved systems compared to legacy and existing systems, this project will enable efficient management and monitoring of inventory and warehouse assets, providing real-time asset tracking and improving data accuracy. This improved data accuracy will allow precise demand forecasting, and by implementing reverse logistics, it can ultimately lead to more cost-effective logistics management.

Previous research survey results show transportation as a major cost in logistics operations. Furthermore, in the same survey, the companies identified storage, inventory maintenance, and stock costs are also their top concerns (Silva et al., 2014). This emphasizes how crucial it is to enhance warehouse and inventory management. Moreover, it is becoming more crucial in the current environment to incorporate recycling and waste management initiatives into logistics operations.

LITERATURE REVIEW

Agile Scrum Methodology Overview

Agile Scrum methodology combines the principles of agile philosophy with the Scrum framework. Agile promotes incremental progress, enabling teams to develop projects in smaller and manageable parts. Scrum is one of the various agile methodologies, which is known for dividing projects into phases called “sprints.” Hirotaka Takeuchi and Ikujiro Nonaka first described it in 1986 in "The New Product Development Game" as "a flexible, holistic approach to product development, where the development team collaborates as a unified group to achieve a shared objective." It promotes self-organizing teams and encourages online collaboration among all members, along with face-to-face communication across the team members. (Sachdeva, S. 2016). This method is widely used in software product development, particularly beneficial for businesses that need to complete specific projects quickly.

The approach of this project management is to focus on incremental development. Due to the iterative and incremental base model of Agile Scrum, it has high flexibility over the project. Each iteration is a sprint lasting two to four weeks. The primary aim of Agile Scrum is to prioritize and build the most crucial features first, resulting in a potentially deliverable product. Additional features are added in later

sprints and refined based on stakeholders and customers' feedback. Unlike traditional project management methods that aim to produce an entire product in a single effort, Agile Scrum emphasizes delivering multiple iterations of a product, maximizing business value for stakeholders in the shortest possible time. Workflow in Agile Scrum methodology depends on close collaboration between the Scrum Team, Scrum Master, and Product Owner through sprints during the development of the software. The primary responsibility of the Scrum Master is to remove any obstacles that might prevent progress. The Scrum Team consists of developers, testers, and other specialists needed throughout the development of the software (Srivastava, A. 2017).

Enterprise Architecture Concepts

Enterprise architecture is an overall framework that businesses use to arrange the organization's business strategy with its IT infrastructure. The purpose of it is to provide a detailed approach to design and manage the complicated connection between business processes, information systems, technology, and organizational structure. It assures that technology investments and infrastructure enhance business goals, by translating business strategies into specific IT requirements. Enterprise Architecture is becoming increasingly recognized as the only management and technology discipline that can produce holistic designs for organizations that are agile and all encompassing. (Bernard, S. A. 2005). Enterprise architecture (EA) helps businesses to align technology initiatives with organizational objectives and formalize

and integrate various technology systems to ensure the smooth ability of the system and avoid the redundancy. In general, enterprise architecture can be considered as a structured and aligned collection of plans for the integrated representation of the business and IT landscape of the enterprise, in past, current, and future states (Niemann, 2006).

The Enterprise Architecture (EA) and standards such as TOGAF (The Open Group Architecture Framework) play a major role in forming project architecture by providing structured methods and complete guidelines. This enables enterprise architect to redesign an organization and its supporting IT systems in a uniform and standard way. (Lankhorst, 2012). The TOGAF offers an ordered approach through its Architecture Development Method (ADM), which guides the establishment and management of enterprise architecture throughout its lifecycle. Also, it includes essential components for classifying architecture objects, and the architecture repository for storing the architectural assets. These structured methods ensure that project architecture aligns with wider goals of enterprise and enables effective communication between stakeholders. An enterprise architecture should provide an elaboration of an enterprise's strategy, while focusing on the core concerns of the stakeholders. (Greefhorst and Proper, 2011, p. 10). Another effective framework is the Zachman Framework, it offers a complete view that helps organize architectural objects and ensures all aspects are considered.

Business processes make the foundation of EA, workflows and procedures are also involved that run the organization's operations. Business architecture portrays how organize is the enterprise, and what qualifications are needed to make the business vision. The application layer facilitates business operations and procedures. This explains the connections and interactions between various applications and the organization's business process. The Data layer describes the building of organization's logical and physical data assets, and data management, which includes databases and data warehouses, also implementing data governance to ensure the quality and security of the data. The Technology layer describes the software and hardware requirements to run the business, data, and application services.

A coherent Enterprise Architecture helps to align an organization's IT infrastructure with the strategic objectives. This alignment is needed for ensuring that every component including architecture of business process, applications, data, and technology works together as one to support and run the organization's entire services. Obviously, we need the approach that would be based on the strategic business goals and business processes and would form the concept of the required information system. (Kurganov, Dorofeev & Nastasyak, 2019).

Microservices Architecture

Microservice architecture is an architectural approach that structures an application as a collection of services that communicates through lightweight protocols. It is commonly known known as the Microservices. It focuses on an exact business function, in terms of processing orders, managing payments, etc. Microservices are typically organized around business capabilities. In Bus Transportation Management System, microservices could be designed for modules like: Inventory and Warehouse Management, Audit Management, Infrastructure Management, and Reverse Logistics. These modules were self-sufficient but interact flawlessly.

There are principles of microservices. First, we have Autonomy. This is an essential characteristic of microservices and wherein it is should be independently deployable, letting update easier, scaling, and development. Secondly, microservices should be Loosely Coupled. This means changes in one service should not need changes in others. Thirdly, is called Single Responsibility. Each microservices should have a sole responsibility such as inventory, tracking, or auditing and focus only on that said domain. Fourthly, is Composability. Each microservices doesn't rely on a single, centralized data store. It also may manage its own database. Lastly, is the Fault Tolerance. Each component has its own failure-handling mechanisms and they are designed to handle failure gracefully.

There are benefits of microservices. It can be flexible as they can be built using different tools, programming languages, and databases, depending on what works best for each task. A real-time tracking may serve as an example. A team can use a different database than a maintenance scheduling system. Microservices gives the agility and speed as this is developed and deployed by a single, small team, independently. It develops readability, modularity, and reduces code interdependency within a codebase. The independency allows the developers to enhance microservices not affecting other services. This also gives fault isolation in the project that every microservices is designed to work on one thing effectively. Should there be changes, only small part of it has to be updated and redeployed that will result to time-saving and effort. Splitting the system into many small services, lets flexibility to manage updates easier and make better use of resources. Unlike monolithic applications, where everything is connected, microservices offer a clear purpose and they can be more complex but have a simple interface to communicate with other services. In terms of handling returns, repairs, and recycling for bus parts, and infrastructure this can be efficiently managed through specialized microservice, reducing overall complexity and ensuring unified reverse logistic flow.

As we have already mentioned the benefits, in this portion, challenges of Microservices will be showcased. As it involves various small services they increase in operational complexity. It also requires careful planning in making communication

between services. The possibilities to encounter some network failures or delays are high during development, deployment, and testing as communications were done through the network which will require approaches and advanced tools.

Given these circumstances, several of the companies still implemented microservices due to its scalability and flexibility. Nonetheless, to have good coordination and management, it is a must to ensure that all services work together efficiently.

DevOps and CI/CD

The agile DevOps workflow known as the continuous integration/continuous delivery (CI/CD) pipeline is centered on a dependable and frequent software delivery procedure. DevOps teams may write code, integrate it, run tests, provide releases, and deploy software updates collaboratively and in real-time thanks to the iterative approach, as opposed to a linear one. The use of automation to guarantee code quality is a fundamental component of the CI/CD process. Automated CI/CD is the foundation of modern software development. CI involves automatically building and testing a system after changes to the source code have been made (Fluri, J., Fornari, F., & Pustulka, E., 2024). Test automation is used to deploy code changes to various environments, deliver applications to production environments, and detect

dependencies and other issues sooner as software changes move through the pipeline. The level of automation utilized in software or app releases is the distinction between continuous deployment and continuous delivery. Code is automatically moved to production-like environments for quality assurance and additional testing in continuous delivery, but after tests are successful, human intervention is needed before the code can be put into production. Automation is extended in continuous deployment. There is no requirement for human consent for the deployment of the code to production once it has passed testing.

Continuous Integration is the first step in the CI/CD process. Developers can work independently with Continuous Integration (CI), starting their own code "branch" to apply minor modifications. The source code can be snapshotted by the developer while they are working, usually using a versioning tool like Git. Git makes it simple to roll back the codebase to a previous version in the event of an issue, leaving the developer free to focus on new features. Individuals' work is then sent into an automated system that builds and tests the code modifications using scripts. Developers submit their code for commitment into the version control repository. Also, it is common practice to establish a minimal rate of daily code commitments per team to facilitate the identification of defects and bugs on smaller delta pieces of code rather than large-scale developments (Muñoz, A., Farao, A., Correia, J. R. C., & Xenakis, C., 2021). The source code updates are assembled into the master code, or "trunk,"

by a continuous integration server following the build phase. Continuous delivery (CD), the following stage of the pipeline, is where the verified code changes from continuous integration are deployed into specific environments or code repositories, like GitHub. They can now be implemented in a live production environment by the operations team. Errors are fixed automatically after the software and APIs are tested. The DevOps team manually sends the most recent build to the deploy stage after receiving notification about it in the final phase of the CD process. The stage of the continuous delivery pipeline aims to release new code as quickly as possible while retaining some degree of human control. Continuous deployment is another meaning of the CD in the CI/CD process. When a set of preset tests, such integration tests that verify code integrity in a simulated environment, are passed, continuous deployment automatically pushes code changes to end users.

In conclusion, for companies that need a dependable delivery process and often enhance their applications, building a CI/CD pipeline is normal procedure. Once implemented, the CI/CD pipeline frees up team members to concentrate more on improving applications rather than the specifics of deploying them to different environments.

Relevant Studies and Research

This chapter presents the related literature sources. This part of the study will discuss various pieces of literature related to the study. The following are collected literary works that will support further knowledge of the researchers about the study.

These are studies that emphasized the efficiency of operation in bus transportation logistics. It identifies and solved problems regarding inventory monitoring. Jorona Aquatic Resources and International Trading, Inc. (2013), is a corporation that markets aquatic resources both native and international in the Philippines. The mentioned corporation has encountered problems in providing quality of service, making worthwhile marketing strategies, communication, and security that slow the processes of business and the rapid growth of income. They discovered that the Integrated Sales and Inventory Management System solved their present problems. They discussed how the system will work, how much the system will cost, what would be the benefits to the company, and also the impact to the employees. It is also stated there that almost three semesters were spent working on this project. With all of the following testing and assessments, the system was established to be more cost-efficient, faster, less reliant, operative, and capable in making immense profit. Aidil Hanafi Amirrudin, Nur Syuhadah Kamaruddin, Nurshahirah Salehuddin, Suraiyu Ibrahim (2023), addresses that numerous business establishments have used the inventory management strategies to strike a balance between

responsiveness and efficiency. The study's primary goal was to measure the outcome of inventory management practices on the warehouse efficiency, specifically supermarkets. This study uses questionnaires as primary data and compiled studies as secondary data in utilizing a quantitative method. The researchers use Statistical Packages for the Social Science (SPSS) to analyze their data. They also use regression analysis to see different factors that affects the main outcome. In result, it shows the researchers two factors inventory reorder points (to know when to order more stock) and supplier partnership management (by working well with suppliers). However, Information Technology did not have much important effect on their study. This study suggested that supermarkets and retails store should implement best strategies for managing inventory and refining how they run their warehouses. A study by Q. Guan and Y. Yang (2020) in Guangxi, China, focuses on solving the optimization of power battery recycling enterprises, and endorse durable and effective progress of the industry. This paper engaged specifically at power batteries, considering the dissimilarities and the two ways to recycle power batteries, namely "vehicle scrap" and "old-for-new". The researchers focused on "third-party" system, where they created a network that involves three key groups: consumers, recovery centers, processing points. They also consider the effect of recycling to the environment or other negative impacts on the environment and built a mathematical model in finding good solutions. They designed a system to recycle batteries with little harm and cost as possible. According to T. Widjaja, A. Gunawan, P. Jodiawan and V. F. Yu (2020), in this study it is said that the reverse logistic has been implemented by different

companies because of its ability to gain more profit. However, due to extensive studies about vehicle routing problem with cross-docking (VRPCD) it only considered forward flow instead of reverse flow. So, they thought how about they incorporate the VRPCD network with reverse logistic, namely VRP with reverse cross-docking (VRPRCD). The researchers proposed the VRPRCD that considers supplier, cross-dock, customers, and outlets. Its main goal is to maximize the efficiency of operation of vehicles and transportation costs. The outcomes of the experiments indicate that CPLEX can only solve small instances of newly generated VRPRCD instances in an optimal manner. They also give sensitivity analysis with respect to the cost structure.

These following studies highlights advancement in logistics and transportation management, aiming to help researchers on their study by giving possibilities, questions and answers. In the handbook presented by Paul Schönsleben (2023) it presents the fundamentals of logistic administration as the main management of products, information, control of its life cycle items and managing both classical and benefit of businesses. This offers justifiable outline for supervisors, specialists and progressed clients. For 6th version of this book, it discussed topics about managing goods, and designing a global product, in terms of distribution, retail, transportation and services. In new section discussion, it indicates the benefit of cooperation between Relationship & Development (R&D) and Engineering departments in producing custom products. At the beginning of each section, it contains that there

are Intended Learning Outcomes (ILO), telling us what to learn from that part. It also includes certification programs like APICS CPIM (Certified in Production and Inventory Management) and ASCM / APICS CSCP (Certified Supply Chain Professional). Carina Pimentel (2023) addresses the energetic operational arranging of an urban logistics framework where an existing transport arrange is coordinates with the urban cargo conveyance prepare, and with a final mile conveyance benefit, to send cargo to city centers. The operational arranging includes the task of each cargo ask to a transport center, where demands are dropped by clients, to a transport benefit beginning at the doled-out center, and to a transport halt to be offloaded by the final mile administrator (LMO). A heuristic calculation centering on the final mile conveyance stage is proposed pointing, to begin with, to minimize the number of transport offloads (i.e., the number of times the LMO needs to go to transport stops) and at that point, to minimize the normal final mile conveyance time of the LMO. The computational tests over a genuine transport arrange dataset in the city of Porto, Portugal, characterized in SOLFI extend, appear the merits of the proposed calculation. According to Gina Ledda and Rhory Fernandez (2015), the transportation and logistics industries include the physical movement of people and merchandise. It is said that logistics specifically engaged to the transport of goods, materials, or products, which incorporates the administration of cargo, capacity, bundling, and stock. The transport and logistics envelop a wide run of exercises and play an imperative part in the development of other segments of the Philippine economy. Extreme competition and growing international trade highlighted the role of the

Service Sector where Transportation, Communication and Storage (TCS) is a component. In 2010, the Service Sector accounted for P3.2B (55%) of the real Gross Domestic Product (GDP) where TCS contributed P428M (13%). The TCS showed the fastest growth of 4.1% for the consecutive years 2008 and 2009. More recently, there has already been a recorded 7.4% increase for the first two quarters of 2011 relative to 2010. These literatures and studies are related to our study about bus transportation management system logistic 1: Developing efficient inventory and warehouse management, audit management, and infrastructure maintenance using reverse logistics which is found very useful for the proponents in making the system.

Ulitskaya, I., Vasilyeva, J., Telushkina, E., & Glagoleva, S. (2022) According to them, from the perspective of value-oriented management of the logistics system for arranging passenger transport, the level of receiver satisfaction determines the parameters of passenger traffic, and two cost groups are developed to achieve, these costs for both the system's operation and its development framework. Meanwhile, the character of the cost behaviour connected to the system's development primarily decides which development plan to use for the system that organizes passenger transportation. Ledda, G., & Fernandez, R. (2015). Discuss that the logistics and transportation sectors are producing higher worth than previously, largely because of the birth of a more competitive climate and the industrial process's functional fragmentation, particularly in the manufacturing industry. It is anticipated that the emphasis will continue to shift from merely moving goods and people across an

archipelagic backdrop to the increasing need for land, maritime, and aviation transport systems as well as logistics to support and ensure the smooth movement of manufactured goods and resources among geographically dispersed businesses both domestically and internationally. They find that transport performance estimations and telecom industry alone estimate that it might be responsible for up to a 7.2% GDP share in 2014, roughly speaking $22.5 billion in nominal revenue. According to Boquet, Y. (2013) Reducing traffic in the metropolis of Manila won't be simple or quick. The design of the streets cannot be changed in a year, yet local traffic jams will still exist. Adding new roads might not be the most effective way to lengthy haul. Better rail infrastructure is required, along with new lines, frequent, roomy trains, and stations that are related. Intermodality seems to function well on several areas of EDSA, with simple change to jeepneys and tricycles from busses and MRT.

Integration of Information Systems in Enterprise Environments

Information systems that have been adopted in an enterprise environment introduce a lot of challenges which could affect success. One example challenge is data fragmentation, where different departments or even systems have their own data and none of them has an overview of the organization. Something that both legacy systems and most of the current applications have in common is that they use a wide range of technologies/standards and the problem emerges when interconnection of these core platform services developed for each system is impossible. Besides, low

quality data, which is manifested in such aspects as how consistent or accurate it may be and so on, can also affect the outcome of integration because the insight that is derived from it is also skewed and hence leads to skewed decisions. However, the integration of information systems (IS) into business environments can also offer numerous benefits. It can restructure industries, realign power dynamics, and enable more effective competitive strategies (Ward,1987). It has many benefits that will help to increase organizational performance and productivity significantly. The next advantage is that data is more coherent and of higher quality, since when all the data that comes from different systems becomes a constituent part of the general picture within the framework of a specific department, it is more standardized and, therefore, less discrepant and redundant.

Information systems integration projects can significantly impact organizational success and efficiency. Successful integration enhances project outcomes, particularly in government units (Kolasa et al., 2020). Many integrated information systems have had considerably rewarding impacts on organizations, irrespective of the field. For instance, at multinational companies such as Siemens, the use of ERP has brought synergy since it combines various processes into one integrated system. This has helped to improve the real-time operation visibility and decisions, thus increasing efficiency and lowering operating expenses. In the health care system, the adoption of an electronic health records system as has been implemented in Mayo

clinic has brought benefits through system integration. Enterprise Architecture (EA) plays a crucial role in aligning business and IT strategies, enabling integration, agility, and change within organizations (Hoogervorst, 2004). An effective integration of an information system (IS) can indeed come a long way in the transformation of an organization by building on efficiency, decision-making and performance. It is related to the Enterprise Architecture (EA) principles because it is the idea that all domain segments such as business procedures, information systems, technologies and structures should be linked to offer synchronized results. Cross-boundary integration is defined as the smooth integration of the above stated domains in the accomplishment of a process that is efficient.

METHODOLOGY

Agile Scrum Methodology in the Project

Agile scrum methodology combines the idea of Agile and the Scrum framework. Agile means “incremental,” this allows teams to develop a project in small increments. And one of the known types of Agile Methodology is called Scrum where its type is known for breaking down projects into ample chunks, called “sprints” where each goal is to build the most important features first and came out with a possibly deliverable product. This is a project management system that relies on incremental development. Also, this methodology is beneficial for the businesses as it will help prompt development of specific project of the business.

The Bus Transportation Management System is a complex project that comprises numerous incorporated areas such as, inventory, warehouse management, audit processes, and infrastructure maintenance. The Agile Scrum was used as a method that emphasizes bringing the project through iterative and incremental development cycles. In this project, the product is built gradually in small, manageable sprints, each focuses on a specific module like inventory management, audit, or reverse logistics. It also considers a working feature where development is delivered, tested, and reviewed. For first sprint, develop basic functionality for warehouse inventory tracking. Secondly, by integrating reverse logistics for maintenance of transportation infrastructure. Lastly, implementing audit tools. The

idea of each sprint, it adds functionality to the system without starting over and allows having flexibility for changes based on stakeholder's feedback.

Agile Scrum promotes flexibility by pointing changes in requirements in development. This is essential for Bus Transportation Management System, as transportation needs may change due to external factors like unstable inventory levels, changes in monitoring requirements for audits, and unexpected maintenance issues. By facilitating stand-up meetings, sprint reviews, retrospectives, and continuous communication among team members, product owners, and stakeholders, guarantees that the project remains associated with user needs and business objectives. Collaboration among developers, infrastructure specialist, and audit experts hold a better understanding of the requirements and challenges in the domain. During sprint planning, input from warehouse management team can outline the development of inventory features, while feedback from the maintenance team might improve reverse logistics.

Using Agile Scrum methodology for this project is very logical as it aligns perfectly with the goals of the Bus Transportation Management System. This is in consideration of frequent updates, dynamic requirements, and stakeholder participation. Frequent updates, emphases on short development cycles that allows for continuous enhancements to features like real-time inventory tracking, warehouse logistics, and infrastructure monitoring. Dynamic requirements, must adapt quickly to new requirements, such as changes in infrastructure difficulties, and logistic

encounters. Stakeholder Involvement is crucial for delivering a system that meets operational needs efficiently. Having the capability to develop a system part by part, while collaborating with stakeholders and adjusting the scope, if necessary, it ensures that the project stays on track to meet its developing objectives.

A focus was placed on integrating reverse logistics into each sprint, to ensure that the system can efficiently manage the return and maintenance of infrastructure components. A dedicated reverse logistic team was included in the sprint planning session to align with the overall inventory and warehouse management goals. The Audit management prioritization, a dedicated sprint was also planned for this feature. The audit module's backlog items were prioritized to guarantee regular updates based on real-time feedback from auditors and early deliveries. Some adjustments are necessary to fit Scrum to the specifics of this logistic-focused project.

Roles and Responsibilities

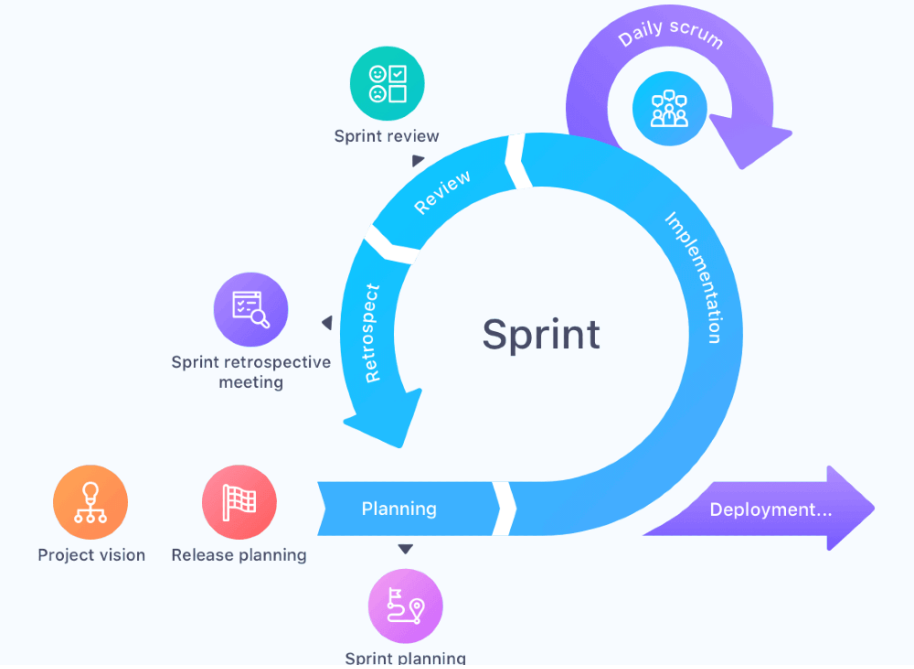
Everyone has a set responsibility in an Agile Scrum team, as guarantee for project success through organized teamwork. Among them is the roles of the Scrum Master in regard to Scrum ceremonies, the identification of barriers to the improvement of the process, educating the team on issues related to Agile values, and the promotion of the required level of change. The Product Owner has the vision

and direction of the product and is in charge of the product backlog where the items are prioritized according to value and the feedback from the stakeholders, organizationally the Product Owner is accountable for interaction with the stakeholders. The Development Team is responsible for providing potentially shippable products at the end of a sprint, in short, working closely with the Product Owner to garner requirements, doing the estimation and commitment within the Sprint Planning, as well as attending all the ceremonies of the Scrum Framework. Although QA often belongs to the Development Team members, they are responsible for testing and, with the help of developers, guaranteeing that the product is up to the expected quality, as well as being involved in setting the criteria that must be met with regard to user stories. A product owner interacts with stakeholders who contribute to product development by offering their perspective and informing the process, as well as collaborating during the Sprint Reviews so they can judge the results. Organized around cross-functional and self-organized structure, organizations will be capable of managing change and delivering valuable products.

Name	**Roles**	**Responsibilities**
Aron III, Ricardo	Scrum Master Developer	• Facilitate Scrum ceremonies • Remove blockers • Ensure agile processes are followed • Write, test, and maintain code • Implement features and fix bugs.
Tejada, Kent Mark	Database Designer UI Designer	• Design database schema

		• Ensure data integrity, optimize queries. • Create user interfaces • Design wireframe • Ensure usability and visual consistency.
Casinillo, Kristel	Technical Writer	• Develop documentation (user manuals, guides, APIs) • Ensure clarity and accuracy.
Barrantes, Andy		
Dizon, John Vincent		

Sprint Cycles



A Scrum Sprint Cycle is that period in time when a team delivers a fixed amount of work. It is usually two to four weeks in length, and each sprint starts once the previous sprint is completed. It is generally known as a continuous development

process. There are four key Agile ceremonies that happen in each sprint cycle, focusing on their roles and purposes. The first one is the "Sprint Planning Meeting" that is held at the beginning of a sprint where the development team along with the Scrum master or product owner plans the work for the following sprint, usually taking some one hour per week of work. This meeting is very important for aligning the team on goals and task prioritization. Following that, there is the "Daily Stand-Up Meeting," held every day, during which the team members may share briefly what they have accomplished, their plans for the day, and what blockers are in their way, this leads to synchronization and the rapid resolution of blockers within about 15 minutes. The "Sprint Review Meeting" shall be conducted at the end of a sprint with the development team, Scrum master, product owner, and stakeholders wherein completed work will be presented that shall be discussed in terms of its possibilities of whether there is a need to amend or offer feedback and can last as long as the planning meeting. Last is the "Sprint Retrospective Meeting," which occurs after it. This enables the team to discuss what was done, what worked, and what did not for the entire completed sprint, along with planned improvements to be made next iteration. It usually takes approximately 45 minutes to 1.5 hours. All these ceremonies are purposed to encourage teamwork and continual improvement and alignment within the Agile framework.

Scrum Artifacts

The term artifact often relates to archaeological ruins and ancient relics. However, in the world of software development, an artifact refers to information one needs during the actual development of the product.

Product backlog is a compilation of all the new features, enhancements, bug fixing, tasks, or any work requirement necessary to create a product. It is compiled from input sources like customer support, competitor analysis, market demands, and general business analysis. As new information arrives, the product backlog is the "live" artifact updated on demand. A cross-team backlog, which is maintained and curated by the product owner in between sprint cycles and whenever new ideas come up, would include tasks that were initially in an active sprint but were deprioritized and moved to the backlog.

The sprint backlogs are the sets of product backlog tasks that get promoted and are to be developed within the next product increment. Development teams create sprint backlogs to not only plan for the deliverables of future increments but also to detail work that is expected to be done in producing the increment. Sprint backlogs are created by choosing any task from the product backlog and then breaking that task into smaller, actionable sprint items. For instance, an example task would be "build a shopping cart page," which might include numerous design and development subtasks. The primary work is in the product backlog, while those secondary supporting tasks like "create a shopping cart visual design mockup" or "program the

shopping cart sessions" are found in the sprint backlog. The sprint backlog is modified during the sprint planning phase of scrum. The smaller sprint tasks are assigned to the appropriate teams, for example, design and development. If all the sprint tasks cannot be delivered by a team, the leftover sprint tasks are kept on standby in the sprint backlog for the next sprint.

A product increment is the customer deliverable produced by completing product backlog tasks during a sprint, and this also includes the increments of all previous sprints. There is always one increment for each sprint, and an increment is decided during the scrum planning phase. An increment happens whether the team decides to release to the customer. Product increments are really very useful and complementary to CI/CD in version tracking and, if needed, version rollback.

In addition, teams reap the benefits of keeping all their work aligned to the backlog items. For example, a team would create a branch and build for each backlog item. When teams have version control and CI/CD integrated into the scrum tool that they track, they can use all that information to better understand work history, enabling reasoning about the backlog items being deployed and released to customers. That also enables the team to look at commits in reverse and then backtrack to a scrum increment to see the history and planning of that code.

Scrum Board

To do	In progress	Testing	Done
Create initial UI for Logistic 1			✓
Create initial database schema			✓
Implement Authentication			✓
Create pre-requisite module (user, category, product)			✓
Create inventory and warehouse frontend			✓
Create inventory and warehouse backend	✓		
Apply initial responsiveness		✓	
Update database schema	✓		
Create Infrastructure Maintenance frontend	✓		
Create Infrastructure Maintenance backend	✓		
Create Initial Analytics frontend	✓		
Create Initial Analytics backend	✓		
Create Audits frontend	✓		
Create Audits backend	✓		
Implement Permission based Access	✓		
Implement Reverse Logistics	✓		
Implement Special/Advance Features	✓		

Product Backlogs

User Story No.	User Story	Tasks	Priority	Sprint	Status
1	As a Logistics Manager, I want to see an initial user interface for Logistic 1 so I can manage operations.	Create initial UI for Logistic 1	1	1	Done
2	As an Admin, I want to create the initial database schema so the system can store data efficiently.	Create initial database schema	1	1	Done
3	As an Inventory Clerk, I want to be able to register and log in so I can access the platform securely.	Implement Authentication	2	2	Done
4	As an Admin, I want to manage users, categories, and products so the system can be set up accordingly.	Create pre-requisite module (user, category, product)	1	2	Testing
5	As a Warehouse Manager, I want a system to track inventory and manage the warehouse.	Create inventory and warehouse frontend and backend	2	3	In Progress
7	As an Inventory Clerk, I want the system to be responsive so I can use it on various devices.	Apply initial responsiveness	2	4	In Progress

8	As an Admin, I want to update the database schema so that it aligns with new business requirements.	Update database schema	2	4	In Progress
9	As a Maintenance Executive, I want a module to manage the system's infrastructure and perform maintenance tasks.	Create Infrastructure Maintenance frontend and backend	1	5	In Progress
10	As an Inventory Analyst, I want to view analytics on the system's performance and operations.	Create Initial Analytics frontend and backend	1	6	In Progress
11	As a Supply Chain Auditor, I want an auditing module to monitor user activities in the system.	Create Audits frontend and backend	2	7	In Progress
12	As an admin, I want to implement permission-based access so I can control who sees what information.	Implement Permission based Access	2	8	In Progress
13	As a Logistics Manager, I want to implement reverse logistics so returned products can be managed efficiently.	Implement Reverse Logistics	1	9	In Progress

14	As an Admin, I want to implement special/advanced features so the system meets our business needs.	Implement Special/Advance Features	2	10	In Progress

Sprint Backlogs

User Story No.	User Stories	Task	Estimated Effort	Owner
	SPRINT 1 Management			
1	As a Warehouse Manager, I want to add new inventory items for stock tracking.	Design and implement inventory input screen	16.8 hrs	R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
2	As an Admin, I want real-time inventory level updates to	Implement real-time inventory tracking system	16.8 hrs	R. Aron III K. Casinillo K. Tejada

	prevent stock-outs.			V. Dizon D. Barrantes
3	As an Admin, I want to integrate the inventory system with suppliers.	Setup supplier integration API	16.8 hrs	R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
4	As an Admin, I want to set up alerts for inventory anomalies.	Implement anomaly detection and alert system	16.8 hrs	R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
5	As a Inventory Planner, I want to view stock reports to assess stock levels and demand.	Implement stock reporting feature	16.8 hrs	R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes

6	As a Inventory Analyst, I want a comprehensive report of inventory.	Create executive reporting module	16.8 hrs	R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
7	As an Admin, I want to set up multiple warehouses for inventory management.	Design and implement multi-warehouse setup	16.8 hrs	R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
8	As a Warehouse operation Manager, I need a stock entry form to register stock quickly.	Create stock entry form and functionality	16.8 hrs	R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
9	As a Warehouse Manager, I want stock transfer	Implement stock transfer report feature	16.8 hrs	R. Aron III K. Casinillo K. Tejada

	reports between locations.			V. Dizon D. Barrantes
10	As a Warehouse Operation Manager, I want to automate stock replenishment from suppliers.	Develop automation for stock replenishment from suppliers	16.8 hrs	R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
11	As an Admin, I want to optimize warehouse layout for better stock retrieval.	Implement warehouse layout optimization module	16.8 hrs	R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes

User Story No.	User Stories	Task	Estimated Effort	Owner
	SPRINT 2 **Performance**			
12	As a Warehouse Manager, I want to track KPIs of warehouse efficiency.	Implement KPI tracking for warehouse efficiency		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
13	As a Warehouse Lead, I want to monitor system performance and uptime.	Develop system performance monitoring dashboard		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
14	As a Maintenance Executive, I want to track maintenance	Implement team productivity tracking for maintenance tasks		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes

	team productivity.			
15	As a Maintenance Executive, I want an integrated report of warehouse, logistics, and maintenance KPIs.	Develop integrated KPI report		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
16	As a Logistic Manager, I want insights into transportation costs.	Implement transportation cost analytics		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
17	As an Admin, I want to manage users and roles	Develop user and role management functionality		R. Aron III K. Casinillo K. Tejada

	to control access to modules.			V. Dizon D. Barrantes

User Story No.	**User Stories**	**Task**	**Estimated Effort**	**Owner**
	SPRINT 3 **Reverse Logistic**			
18	As a Logistic Operator, I want to automate reverse logistics entries.	Implement automation for reverse logistics entries		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
19	As a Logistic Manager, I want to track returned items via reverse logistics.	Set up reverse logistics tracking system		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes

20	As a Logistic Manager, I want a real-time reverse logistics dashboard.	Develop real-time reverse logistics dashboard		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
21	As a Logisitc Auditor , I want to track cost savings from reverse logistics.	Create cost-saving analytics for reverse logistics		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes

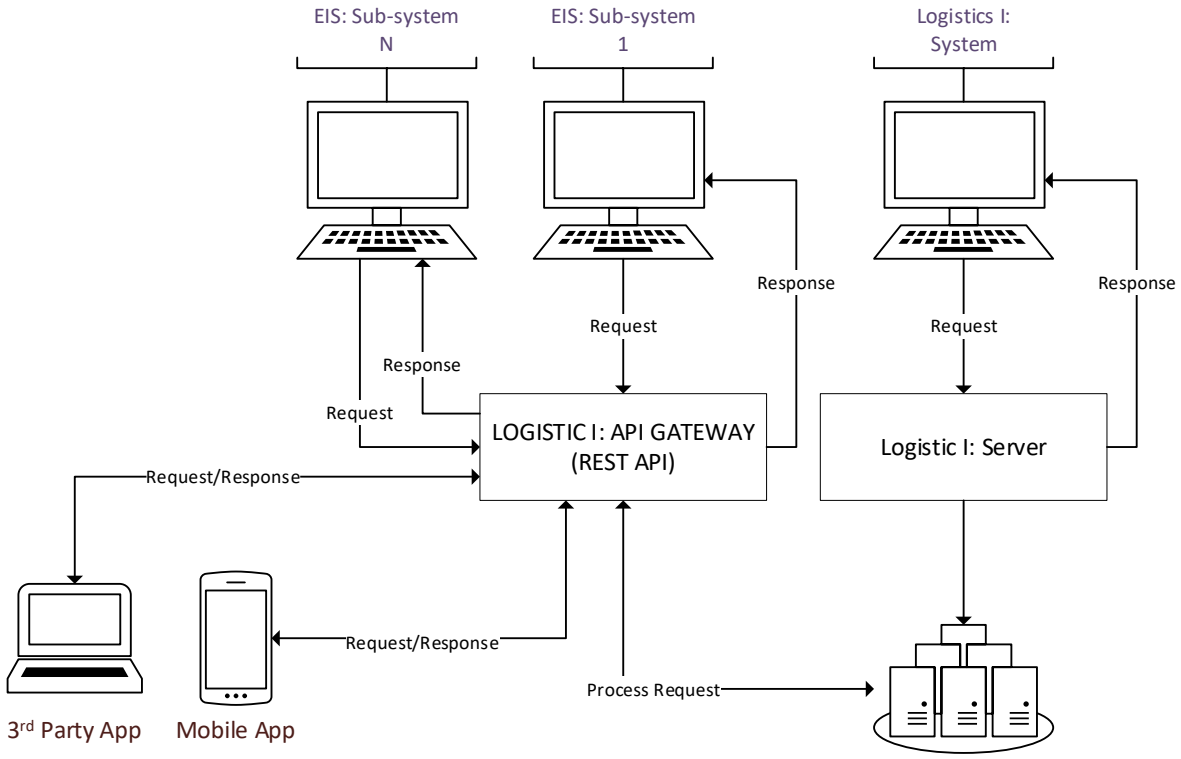
User Story No.	**User Stories**	**Task**	**Estimated Effort**	**Owner**
	SPRINT 4 **Audit and Reporting**			
22	As a Logistic Auditor, I want a dashboard to	Create audit management dashboard		R. Aron III K. Casinillo K. Tejada

	view and track audit records.			V. Dizon D. Barrantes
23	As a Compliance Officer, I want to generate audit reports for compliance.	Finalize audit report generation functionality		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
24	As a Compliance Officer, I want a detailed audit trail of stock movements.	Implement stock audit trail functionality		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
25	As an Admin, I want enhanced security features to protect sensitive data.	Develop and implement enhanced security protocols		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes

User Story No.	User Stories	Task	Estimated Effort	Owner
	SPRINT 5 **Maintenance**			
26	As a Maintenance Planner, I want to track maintenance schedules.	Implement asset maintenance schedule tracking		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
27	As a Maintenance Supervisor, I want automated maintenance notifications.	Develop automated notifications for maintenance tasks		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
28	As a Maintenance Executive, I want	Develop maintenance activity log		R. Aron III K. Casinillo K. Tejada

	log maintenance activities.			V. Dizon D. Barrantes
29	As a Logistic Operator, I want predictive analytics for asset failures.	Implement final predictive analytics model		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes
30	As a Maintenance Executive, I want to see the logging of maintenance completed tasks.	Implement task completion logging for maintenance activities		R. Aron III K. Casinillo K. Tejada V. Dizon D. Barrantes

Integration Approach for Information Systems



The Bus Transportation Management System (BTMS) uses microservices architecture as a software architectural pattern. The developer conducted a research study and chose the most commonly used method for integration in the microservices architecture. There are numerous methods and tools needed to implement integration in the system, but there are several factors that must be considered. First is the familiarity of the developers with the tools and the complexity of how to use them. Second is the compatibility of these tools with the system that is currently being developed. The learning curve and technicality of the tools must also be considered. The first choice for implementing integration in the system is RESTful API, and the second is GraphQL. The development team chose RESTful API for the system's

integration considering its simplicity and flexibility, and it is also highly compatible with the Laravel framework and vanilla WAMP. Although the REST API uses JSON as a standard, it is easily convertible in both tech stacks. MySQL also accepts JSON files as a valid datatype. Moreover, REST API uses HTTP as a primary communication protocol due to its widespread support across different platforms. While GraphQL is also highly compatible, learning this language and understanding its learning curve is uncertain. The team did not take the risk and also considered the time constraints in developing the system. Logistic I group particularly used WebSockets to implement real-time data updates and communications.

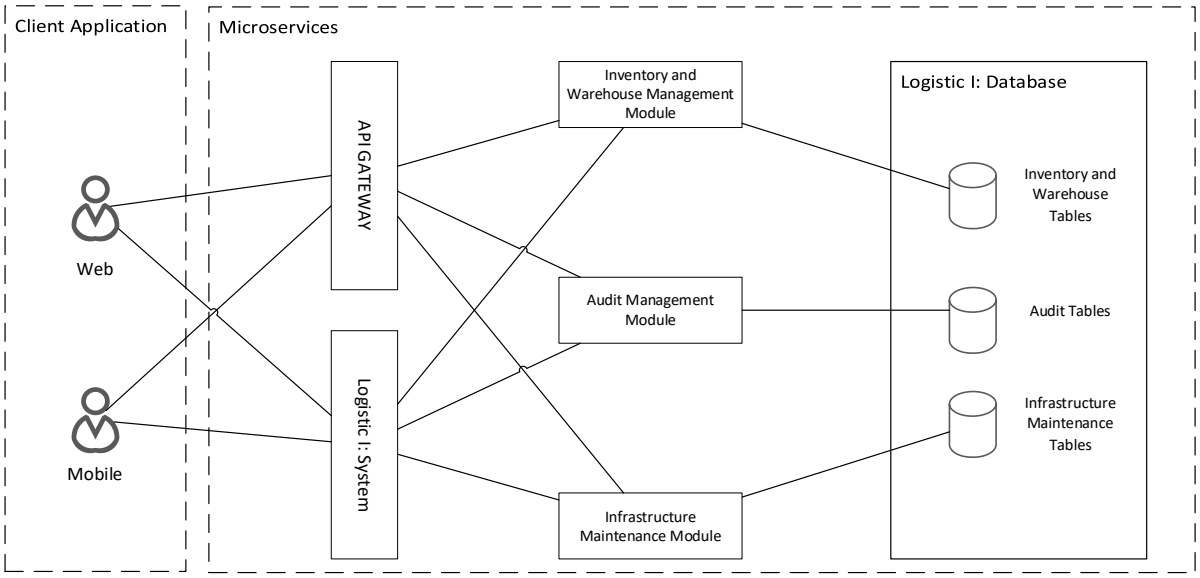
Introduction to TOGAF and the Four Architectural Domains

The Open Group Architecture Framework provides an approach for designing, planning, implementing, and governing an enterprise information technology to secure the enterprise's success when integrating an information system in their company. This framework is highly relevant considering that the goal of this project is to design and to develop an enterprise information system for the bus transportation companies, specifically for those who offer a provincial trip. TOGAF will help the team and the developers to create a highly compatible, flexible, efficient and effective system in this kind of enterprise.

To guarantee success in integrating an information system in an enterprise, TOGAF provides four architectural domains including business, data, application, and

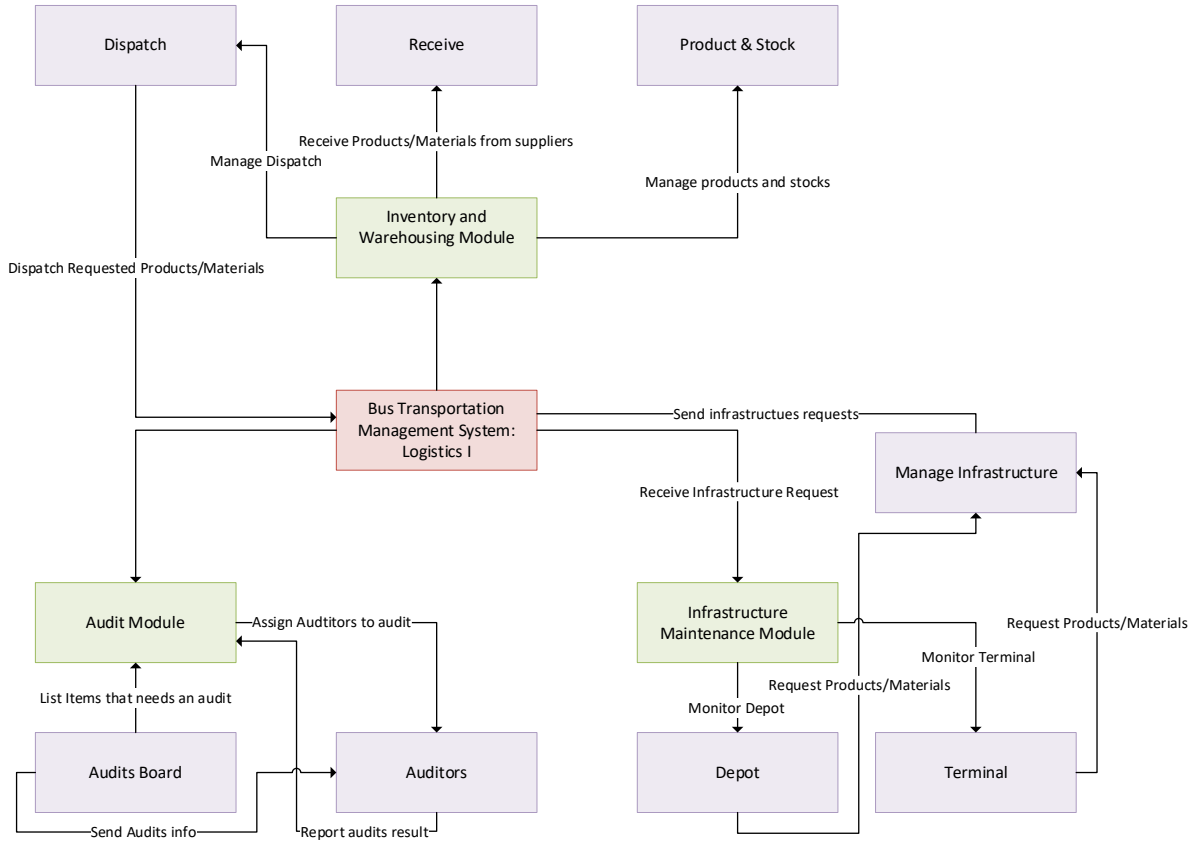
technology domains. Business architecture defines the business strategy, governance, organization, and key business processes of the enterprise. It focuses on understanding the business structure and operations, and how they align with business goals. Data architecture provides a framework for managing the enterprise's data assets, describing how data is stored, processed, and used. This architecture focuses on the structure of an organization's logical and physical data assets. Application architecture describes the individual applications, their interactions, and how they support business processes. The scope of application architecture is to focus on designing the application systems, their components, and their interconnections. Lastly, technology architecture defines the hardware, software, and technology infrastructure required to support the business, data, and application architecture. Its scope is on the IT infrastructure including networks, servers, storage, and cloud services. These four domains will give the developers a process for developing and managing an enterprise architecture. This project will leverage these domains to create a comprehensive approach for designing and developing an enterprise system tailored for bus transportation companies. By aligning business processes, application systems, data management, and technology infrastructure, the solution will ensure that all aspects of the enterprise architecture work cohesively to optimize operations and enhance overall service delivery.

Microservices Architecture

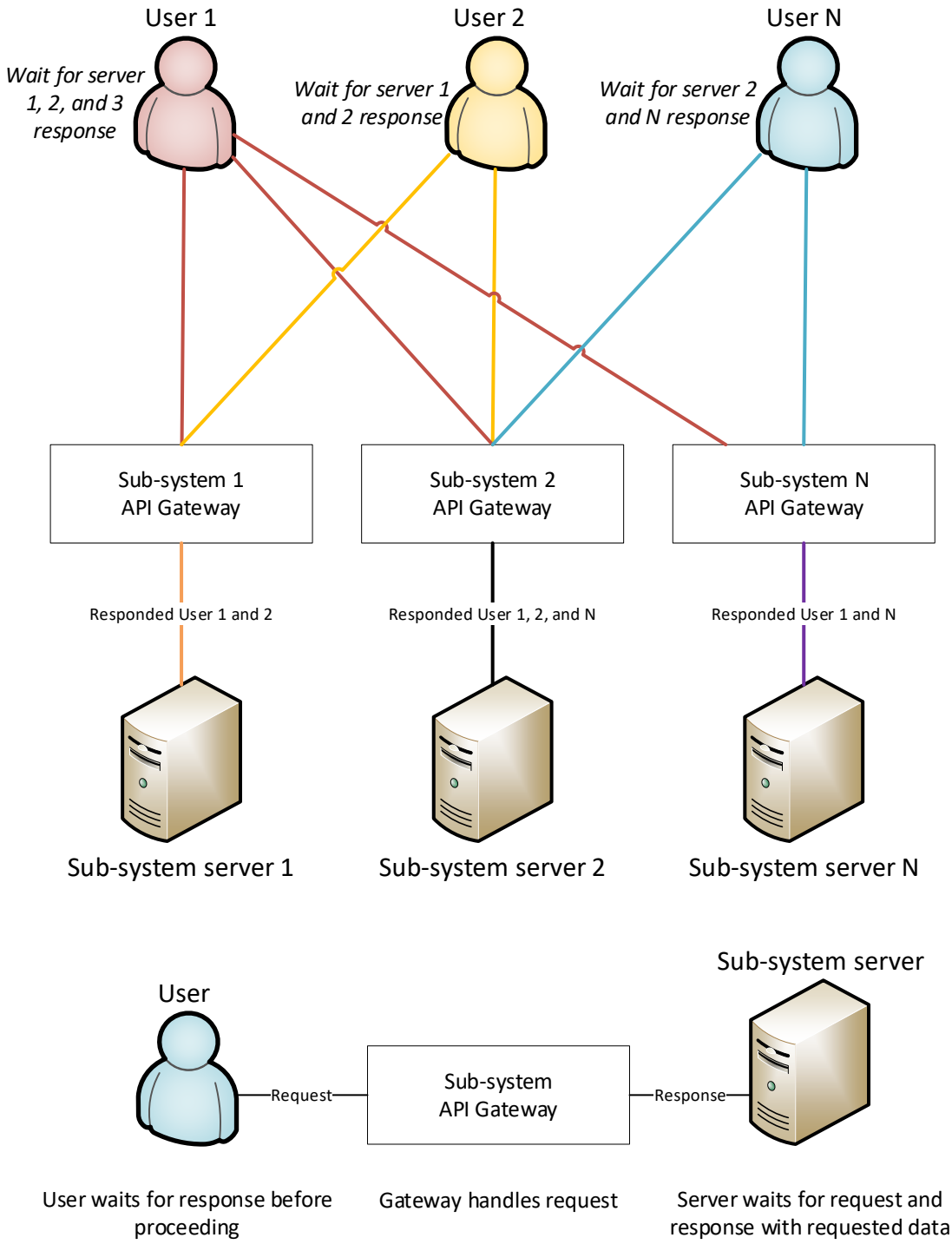


Microservices Architecture is the most suitable software architectural pattern for the project considering that the cluster is composed of 10 sub-systems, which are independent from each other. Also, each sub-system uses microservice resulting to great increase in flexibility and compatibility in different level of our system. The developers decided to choose microservice as the architectural pattern to satisfy our system needs to increase efficiency, productivity, optimize development costs and specially to maintain its flexibility all throughout development period. The diagram shows two options on how to make a request on a server. The developers decided to separate the gateways for internal and external modules to maintain availability and to prioritize logistics operation. *Logistic I: System* is for logistic I sub-system only, while the *API GATEWAY* is for external modules inside bus transportation management system and for 3rd party web or mobile application.

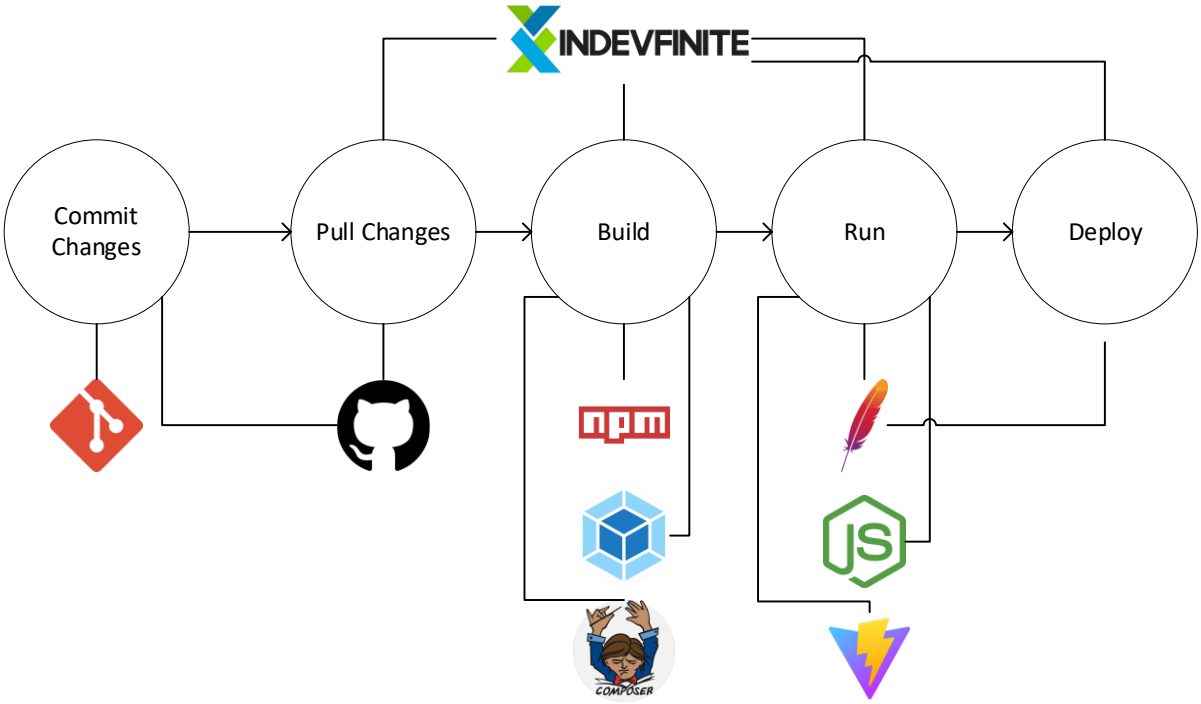
Data Flow Diagram: Level 2 (Logistic I)



Communication Patterns (Asynchronous)



DevOps Implementation



The diagram illustrates the CI/CD (Continuous Integration / Continuous Deployment) workflow in Logistic I sub-system, outlining the software development lifecycle from code changes to deployment.

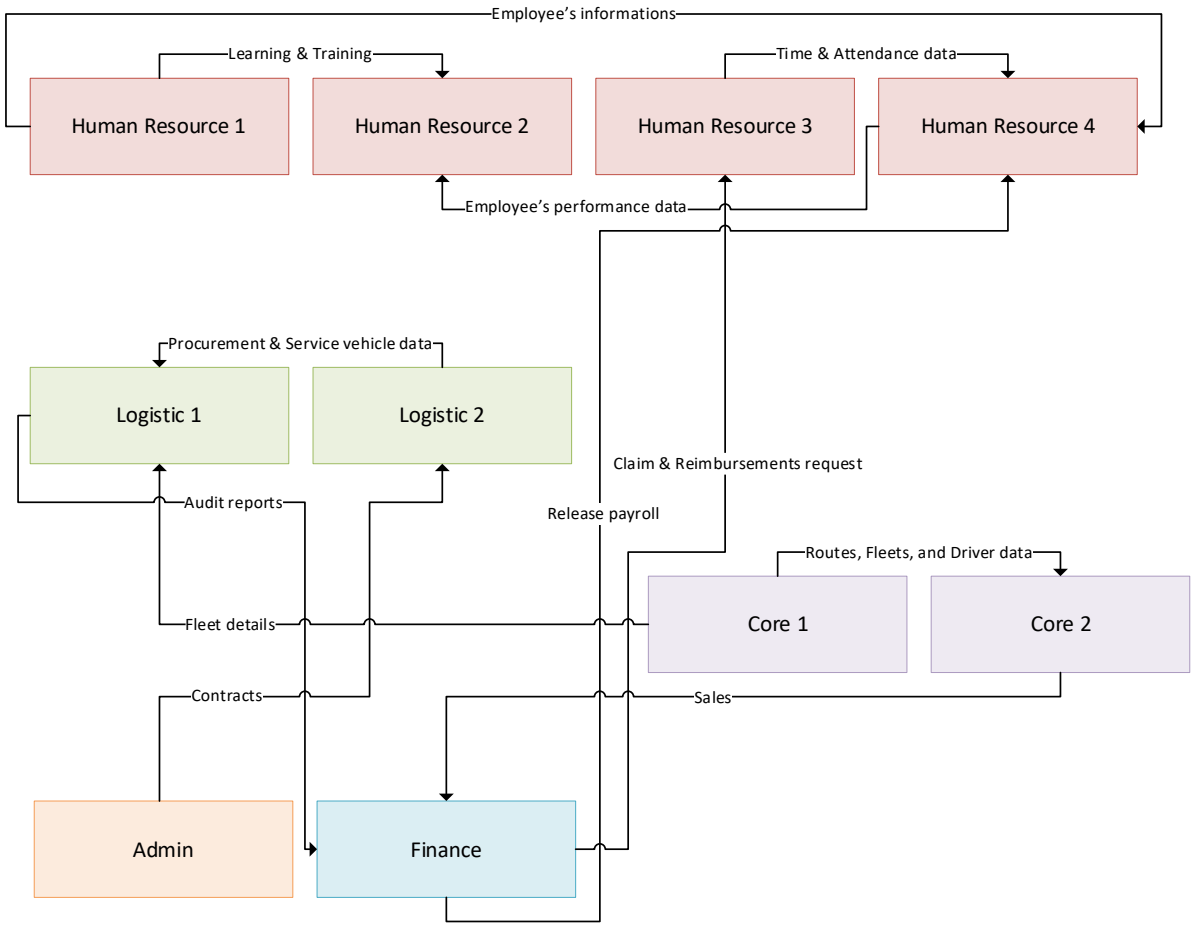
It begins with developers committing changes to a version control system like Git to track modifications. Once the changes are pushed to a shared repository such as GitHub, other developers or CI pipelines can pull these changes to keep their code up-to-date. After pulling the latest updates, the build process is initiated, which ensures the code compiles correctly and all dependencies are resolved. Key tools involved in the build phase include NPM for JavaScript package management,

Webpack for bundling code and assets, and Composer for managing PHP dependencies.

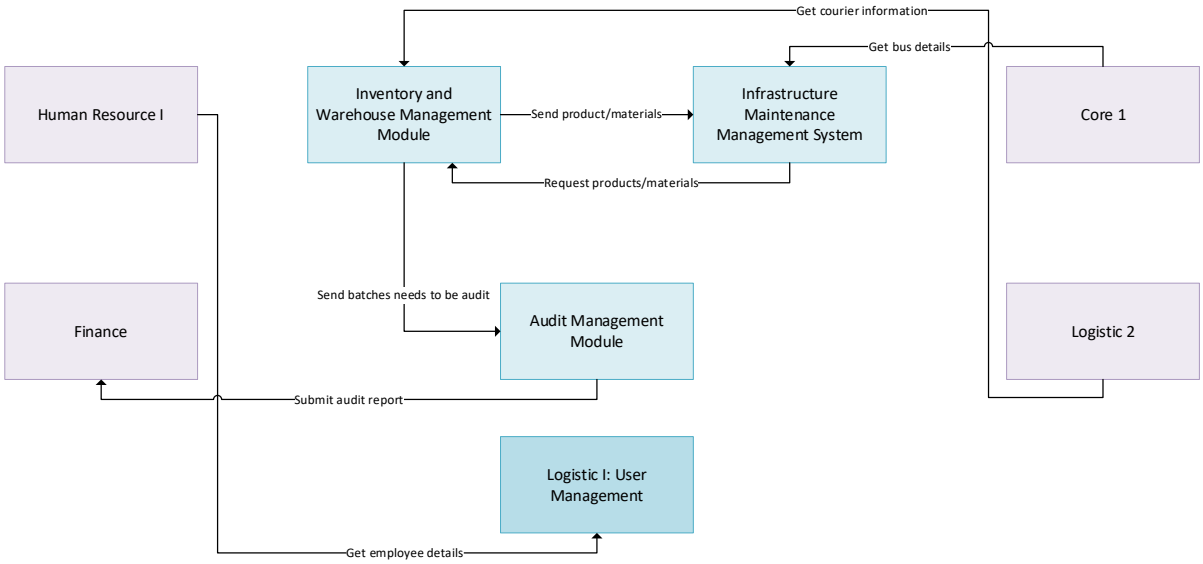
Once the code is successfully built, it moves to the run stage, where the application is tested or executed using various tools and technologies. Node.js provides a runtime environment for server-side JavaScript, Apache serves web applications and APIs, and Vite helps with fast development, offering features like hot module replacement (HMR) for modern JavaScript frameworks. If the application performs as expected, the final step is deployment, where it is released to a production or staging environment, making it accessible to end-users. At the center of this process is INDEVFINITE, a web hosting platform coordinating and managing the entire pipeline from code changes to deployment.

Integration

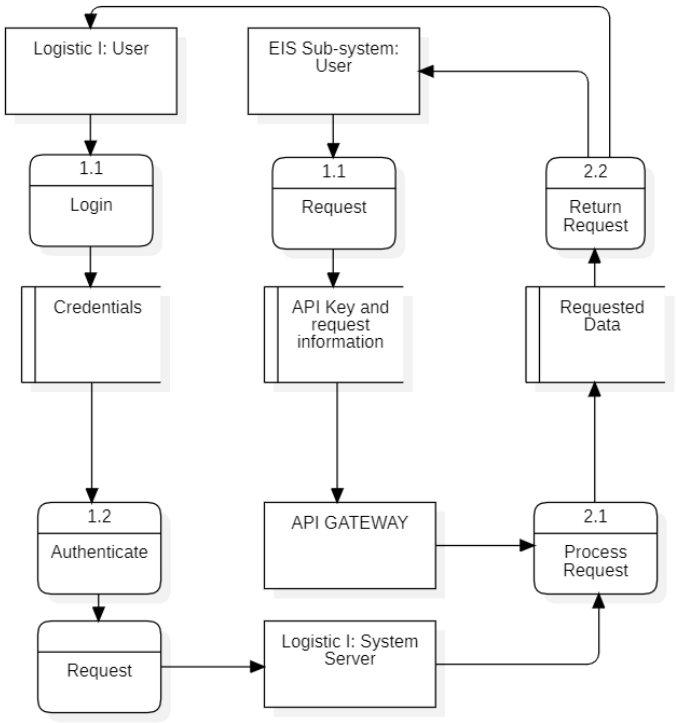
Integration Diagram: BPA Level 1



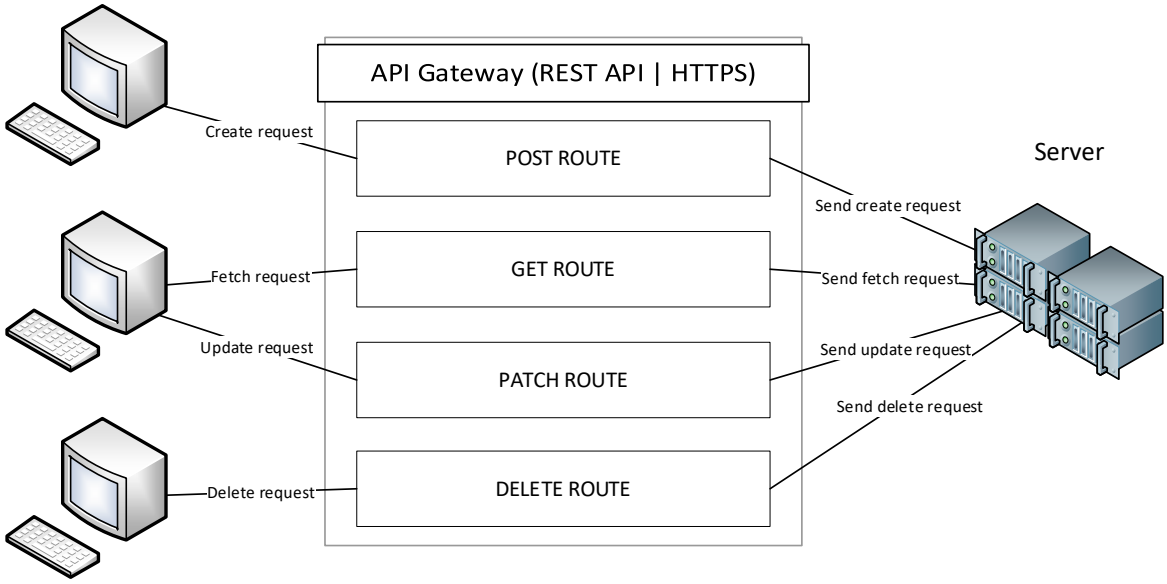
Integration Diagram: BPA Level 2



Data Flow Diagram: Level 3



API Gateway


API Gateway (REST API | HTTPS)
POST ROUTE
GET ROUTE
PATCH ROUTE
DELETE ROUTE
Create request
Fetch request
Update request
Delete request
Server
Send create request
Send fetch request
Send update request
Send delete request