JGB 1747

AGISS: A Web-based Inventory and Sales Monitoring System for AG Florendo Beverage Distributor

Emilio Laurence Dimalanta, Meiggan Circa & Randy Domantay

Saint Louis University

rdomantay@gmail.com, 2201851@slu.edu.ph, 2202788@slu.edu.ph

Abstract

Companies need to automate how they handle their resources to make their inventory processes more efficient and conducive to conducting business. Inventory and sales management systems determine which item sells most and which sells less. In contrast, inventory control significantly impacts the stability of the availability of goods being sold by retailers. A Web-Based AG Florendo Inventory and Sales Monitoring System (AGISS) was developed for AG Florendo Inc. This project aimed to design, develop, test, and deploy a web-based system that handles inventory-related transactions, sales, delivery status, reports, and on-call orders. It was developed using the agile software development method. This method divided large tasks of this project into smaller tasks through several iterations. Planning, designing, developing, testing, deploying, and reviewing occurred in each iteration. The technologies used for the front end include HTML, CSS, Bootstrap, Material Design Bootstrap, and Vue.js. In contrast, PHP, MySQL, phpMyAdmin, and Apache Server were used to develop the back-end processes. This project will eventually improve the company's conduct of inventory and management of its resources which will, in turn, contribute to the improvement of the company's bottom line and make it more responsive to the needs of AG Florendo's clients.

Keywords: *AG Florendo, Inventory, Sales Monitoring, Agile*

Introduction

Beverage distributors are usually between the products and the company's connections, such as restaurants, retailers, and others already buying the company's products. They provide the final products to be sold to the consumers of beverage products of a company. AG Florendo Inc. Tarlac City delivers Coca-Cola products to its clients. They sell and distribute products to their clients. This company uses whiteboards and Microsoft Excel to keep track of its inventory supplies and sales records.

Their current process involves multiple roles. The encoder's job is to manage every report of the deliveries given. They review recorded transactions, sales, and inventory to ensure correct records. If there are no discrepancies, recordings are encoded into a spreadsheet application. They serve as cashiers for walk-in orders also. Since the warehouse is located on a highway and many people pass, many cases are sold to these customers. Next are the helpers; they are the ones who accompany the drivers for deliveries. Another part of the company is the checkers. They are assigned to count the cases of the products arriving at the company's warehouse. They verify how many cases of Coca-Cola products arrive at the company and label each case that will be delivered to the destination of an order. The labels are for the delivery drivers to know where the products will be delivered. Usually, checkers accompany drivers and helpers to ensure it is properly delivered to the clients.

Moreover, lastly, the drivers deliver the products to the clients. They are also responsible for communicating their queries and suggestions to the clients. The numbers of employees that handle these different roles in the company are as follows: 1 for the encoder, 4 for drivers, 3 for helpers, and 2 for checkers.

AG Florendo has no prevalent pain points in their current process, but it limits the company to scale. Automation can be one of the improvements that can be applied to streamline operations,

increase efficiency, and facilitate scalability. This enhancement aims to provide transparent information across the business processes. Parties involved, such as employees, retailers, and end customers, are considered. Data is gathered from the different parties to use as information the system can access and utilize to perform business logic.

Review of Related Literature

Inventory control is an activity in which it organizes and manages the items available for clients. It coordinates the distribution functions, production, and procurement to meet the requirements. According to Stevenson (n.d.), this role contains new products, supplies to be sold, outdated products, consumables, spare parts, and all materials.

Inventory and sales management systems determine which item sells most and which sells less (Abdul Aleem, 2013). A traditional way of inventory management is through pen and paper, writing down which, what, and how many products were sold daily, weekly, and monthly. This way of gathering and processing information may also be manual and has yet to use a fully automated system.

Inventory management aims to find the right balance between having enough inventory to meet customer demand and avoiding excess stock. AGISS will be used to facilitate inventory management processes by reducing stock levels. This helps minimize inventory-related costs and maximize service levels.

An inventory management system is a basis for knowing what items are sold and taking up shelf space for companies, smaller businesses, and vendors (Crosby, n.d.). The system is balanced in ensuring that customers should always have enough of whatever they want against the financial needs of a retailer to maintain as little stock as possible (Zierden, 2009).

Modern Inventory Systems are improving efficiency and reducing the risk of mismanagement, unrecorded sales, overstock, or understock. Each member has roles that enable

and limit them to conduct specific operations. One aspect of inventory systems is reordered levels. It is a management technique used to assess the margin of safety and create a well-maintained inventory system. It is connected to the reorder point, which is the point in the graph or chart where it tells the system to order the products that almost run out of stock in the inventory (Karakatsoulis & Skouri, 2021). The reorder point is the value of the readily available products to be sold to the customers. When the reorder level hits the minimum level, it points to the lower part of the inventory chart, where stocks must be filled to maintain the supply of goods in the system.

Framework

This section describes the technologies that were used for the development of AGISS and its system architecture. Technologies include web applications and technologies, database storage and management, code editor, testing, collaboration, and documentation tools.

Visual Studio Code (VSCode) was the code editor that the researchers used. It can support multiple programming languages and web technologies such as HTML, CSS, and PHP. It also allows automation testing, which the researchers utilize on every iteration that has been implemented. Vue.js was used as the main framework for front-end development. JavaScript was used alongside Vue.js.

For the backend development, native PHP was utilized. PHP, as the backend, is responsible for extracting the database and allows the creation of dynamic content. MySQL was the database storage technology that the researchers used. Management of MySQL was done by using phpMyAdmin.

Collaboration tools served as the platform for the researchers to collaborate and deliver the requirements of AGISS. Microsoft Word and Google Docs were used for the documentation of the project. Figma and Draw.io were also used to create mockups and diagrams for documentation purposes.

The system architecture of AGISS consists of two components: the administrative modules and the client module. AG Florendo and its employees can only access the administrative modules. It provides functionalities such as creating inventory entries, recording sales, managing orders and user accounts, and others. On the other hand, the client module can only be accessed by the customers of AG Florendo. It provides functionalities such as creating orders, viewing their status, and viewing recent orders. The clear distinction between the administrative and client modules in AGISS opens up opportunities for adopting a web-based inventory system. By utilizing web technologies, AG Florendo can leverage the advantages of accessibility, real-time updates, and streamlined operations.

AGISS runs on a server-side stack consisting of PHP, MySQL, and Apache. These technologies enable efficient data processing and storage, handling inventory-related transactions effectively. Independent variables, such as the computation of sales and inventory counts, can be efficiently processed and stored using these technologies. This process improvement may influence the adoption of AGISS since this enables AG Florendo to scale.

AGISS uses HTML, CSS, Bootstrap, JavaScript, and Vue.js on the client side. These technologies handle the presentation and interaction aspects of the system, such as how the system looks and how users can interact with it. Implementing a user-friendly interface using the mentioned client-side technologies could improve the user experience, making it easier for users to interact with the system, which may encourage them to integrate it into their daily operations actively.

To use AGISS, users need to be connected to a network. Once connected, they can access AGISS through a web browser like Google Chrome or Mozilla Firefox and log in with their credentials to access its features. With this, users can conveniently access AGISS from different devices regardless of location. This accessibility can positively influence businesses to utilize a

web-based inventory and sales system. The visual representation of the system architecture of AGISS can be seen in Figure 1.

Methodology

This section discusses the methodologies used to develop AGISS. The development of AGISS would follow the Software Development Life Cycle procedure (SDLC). Software Development Life Cycle (SDLC) is a method that the industry uses to create, generate, and test high-quality software. The SDLC aims to create high-quality software that meets or exceeds customer expectations while finishing on schedule and under budget.

The Agile Software Development Life Cycle methodology was utilized to achieve the study's objective. The approach in agile methodology is to divide large tasks into smaller tasks through iterations (Balaji & Murugaiyan, 2012). Planning, designing, developing, testing, deploying, and reviewing will occur in each iteration. The iterations the researchers took were two. This allowed the researchers to check if the specified requirements were met. If not, then they are added to the project's backlog. The researchers chose the agile approach to deliver more effective and quality features for the proponent. The agile approach also allowed the researchers more time to review if implementing the specified requirements was proper and worked as expected. See Figure 2.

Functional and Nonfunctional Requirements

In determining the functional and non-functional requirements, the researchers used data elicitation techniques such as conducting interviews and reviewing related literature to identify and analyze the requirements of AGISS. Interviews have been conducted with the owner of AG Florendo and the warehouse encoder to gather information about their business processes and allow the researchers to assess the company's areas of improvement. The researchers did not interview the other warehouse personnel because the owner and the encoder explained what they

usually do or work inside their warehouse.

The researchers reviewed, tested, and studied existing inventory systems like Snipeitapp and Odoo. By carefully analyzing the processes of these existing systems, the researchers could identify and define which essential features could be added tailored to AGISS. Furthermore, the researchers also conducted reviews on related inventory and sales management studies to understand AGISS further for the company's betterment. These reviews, as mentioned above, studies, and testing aided in providing insights to the researchers for the features, requirements, and functionalities of AGISS.

Creating the Design of the System

In creating the Design of the System, the researchers utilized this stage to understand how AGISS would work and what it would look like and identify the technical requirements needed to develop AGISS. Process Flow Charts were utilized to understand how AG Florendo conducts their operations. See Figures 3, 4, and 5. In addition, Data Flow Diagrams were also utilized to help the researchers to visualize how data will be processed and who will be responsible for the data that will be passed in AGISS, and to show how the data inputs and outputs flow through and are processed in AGISS and how they are stored. Moreover, Entity Relationship Diagrams were also utilized to visually represent what data types are stored and how entities and their attributes relate to each other in AGISS. Lastly, Mockups were utilized to provide an initial mock-up of how the system’s web page will look like. The Process Flow Chart and Data Flow Diagram were the basis for creating the mock-ups.

Development of the System

During the development of AGISS, the researchers utilized web technologies such as HTML, CSS, JavaScript, PHP, and MySQL. These widely-used technologies form the foundation of web development and are supported by various libraries for hosting. They offer comprehensive

features that enable the creation of both the front-end and back-end of AGISS.

Deployment of the System

The researchers deployed AGISS by setting up a virtual server using a LAMP stack (Linux et al.). They first created an Ubuntu virtual machine in Oracle VM VirtualBox and installed Apache, MySQL, and PHP. After installing the mentioned development technologies, the researchers enabled Ubuntu's firewall, configured the Apache server where the AGISS's root directory was specified, and set up the MySQL database. In this process, the researchers imported the tables populated in a local machine using phpMyAdmin. Finally, the researchers cloned the repository of AGISS from GitLab and installed phpMyAdmin in the virtual machine for easy database management.

Testing of the System

In Testing the System, the researchers utilized manual testing during this stage. Test cases were created to ensure that AGISS was functioning correctly without any bugs and hindrances to any processes during the implementation phase. The testing of the system was also utilized to validate if the specified requirements had been met. Unmet requirements are added to the project's backlogs and identified bugs through testing.

Discussion of Results

The researchers were able to develop AGISS following the methodology that was set and chosen and was able to implement the specified requirements, such as allowing users to create, update, and inventory details as well as products, manage customer orders, including walk-ins, notifying users when the inventory is low, allowing customers to order products, modify their orders or cancel it and others.

AGISS comprises five modules: encoder, checker, helper, administrator, and client. In the encoder module, users can create inventory entries, manage product details, record sales, handle

customer orders, and create user accounts. The checker module offers similar functionalities, excluding user account creation. The helper module allows users to access product information, sales data, customer orders, and inventory details. The administrator module encompasses all previous functionalities, providing additional control over user activity, access levels, and password management for employee and customer accounts. The client module allows users to create and track their orders.

Conclusions

By utilizing a web-based inventory and sales management system, AGISS (AG et al.) was created and planned to enhance its system of decision-making and profitability for stakeholders. AGISS makes use of automation and web-based algorithms to reduce manual operations. AGISS provides a summary of inventory and sales. In addition, generated reports are also provided by AGISS, which will, later on, help the employees to base their business model and decisions for the betterment of the establishment. AGISS is accessible to employees of the organization online, providing them with data about the operation of the business. Through AGISS, AG Florendo can maximize their use of available resources and technology inside the organization.

During the deployment of AGISS, the researchers encountered a problem where the data needed to be loaded into the web pages of AGISS. Some of the SQL query statements are not executed because a system variable in MySQL restricts it. The researchers resolved this problem by reconfiguring the mentioned MySQL system variable to allow the execution of the SQL query statement the researchers made.

AGISS was designed using several diagrams: The Use Case, Data Flow Diagram, Entity Relationship Diagram, Process Flow Chart, Data Dictionary, and Application Architecture. The basis of these diagrams was from the interviews made by the researchers with the proponent.

The development of AGISS was based on functional and non-functional requirements.

Most requirements were fulfilled, though some non-essential requirements were left unresolved due to time constraints.

Limitations and Recommendations for Future Research

Limitations

The researchers determined the functional and non-functional requirements through requirements gathering and analysis. These requirements were further realized using the SDLC methodology the researchers followed. Some of the requirements specified by the researchers, such as the Sales Forecasting feature, were not implemented due to integration issues. The researchers were able to create a model. However, the technology stack did not appropriately allow the Python code to integrate into AGISS.

Recommendations for AG Florendo:

1. Migrate the database to another with an improved capacity to achieve scalability, support, and sustain AGISS as it improves over time.

For Developers:

1. When connecting MySQL and Vue, the Axios library can access data.
2. When using the Datatables library with Vue, one of the challenges in making these technologies work together is that Datatables gets introduced first before Vue. To resolve this, developers can use promises that would wait for Vue to finish rendering table data and then instantiate the Datatables object.
3. When using native PHP for queries, compiling them in a single file makes it easier to maintain. This makes it easier for developers to debug and update query statements.
4. The form action should be in a separate PHP file. This also makes it easier for developers to maintain, debug and make changes as necessary in the source code.
5. Continuously validate and clean user input before storing it in the database. Ensure it does not

contain anything damaging the system or the database.

6. An account should not be logged in to other devices/browsers while currently logged in, especially if the system handles important data. The account should either be logged out when someone attempts to log it in in other accounts/devices or prevent the login attempt to protect the integrity of the system's data.
7. Lessen the user input to simplify the user's task on the system.
8. Button labels should be clear and understandable to avoid confusing the user.

For Future Researchers:

1. One way of improving the system is through a full-featured inventory location visualization which would help provide insight into the product's whereabouts and lifespan without using RF scanners.
2. Business Intelligence features such as sales or demand time-series forecasting would also contribute valuable insights. This would help in countering overstocking and understocking.
3. Customer profiles, where the system can determine good customers from bad ones or what products they often buy using data.
4. First In, First Out feature to determine which products should be sold first depending on the product's arrival date.
5. Automated calculation of the order's delivery fee based on the customer's location would help the user process the customer's orders.
6. Using a wizard format for the user interface of processing customer orders will also make the user's job easier and make the user experience pleasurable.

References

Abdul Aleem, R.B. (2013). Sales and Inventory Management System. Universiti Teknologi Petronas. http://malrep.uum.edu.my/rep/Record/my-utp-utpedia.13591

Balaji, S., & Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. International Journal of Information Technology and Business Management, 2(1), 26-30.

Crosby, T. (n.d.). How Inventory Management Systems Work. HowStuffWorks. https://money.howstuffworks.com/how-inventory-management-systems-work.htm

Karakatsoulis, G., & Skouri, K. (2021). Optimal reorder level and lot size decisions for an inventory system with defective items. Applied Mathematical Modelling, 92, 651–668. https://doi.org/10.1016/j.apm.2020.11.025

Stevenson, W. (n.d.). Operations Management 8th edition. McGraw Hill. https://slideplayer.com/slide/6909061/

Zierden, T. (2009). 4 Keys to Inventory Management. F&I and Showroom. https://www.fi-magazine.com/309238/4-keys-to-inventory-management

Appendix

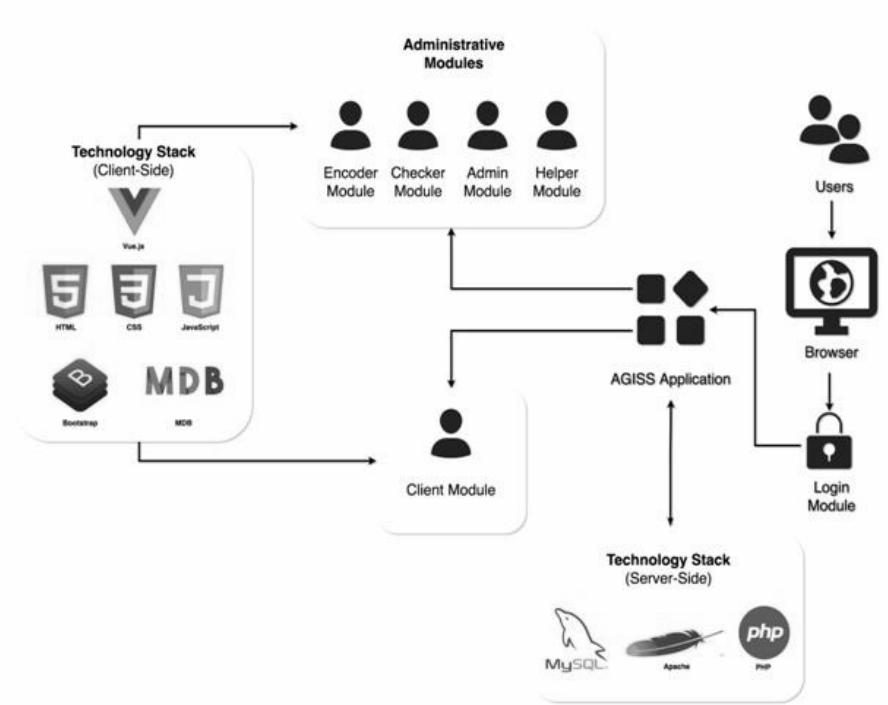


Figure 1. AGISS System Architecture

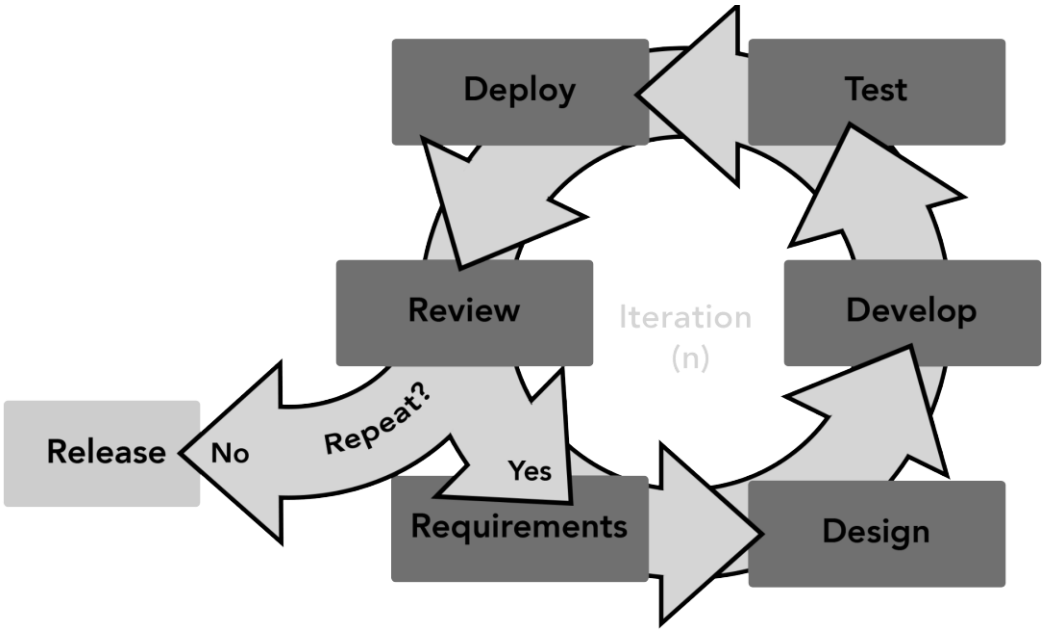
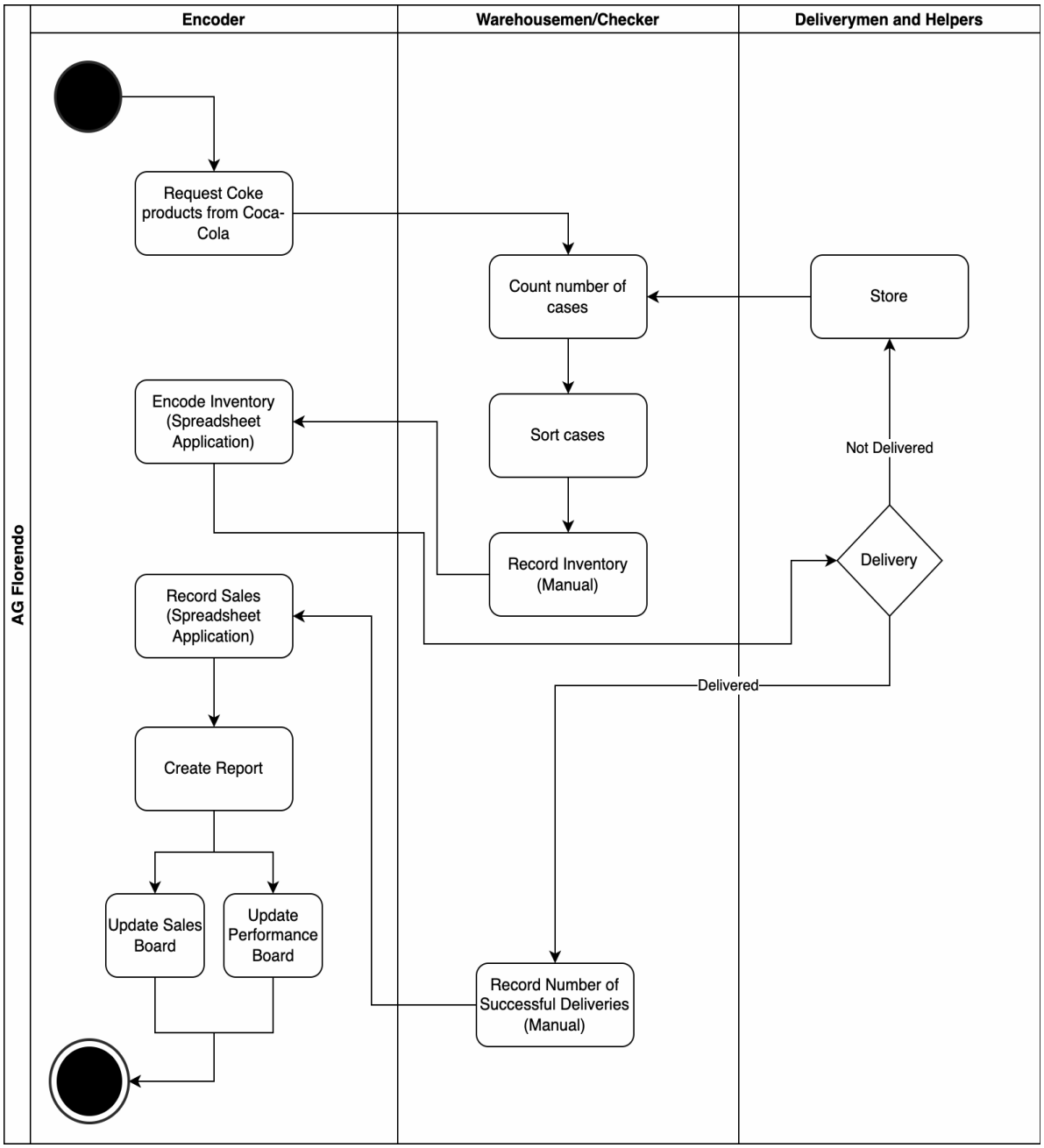


Figure 2. Agile Methodology Process

Figure 3. Current Business Process Flowchart



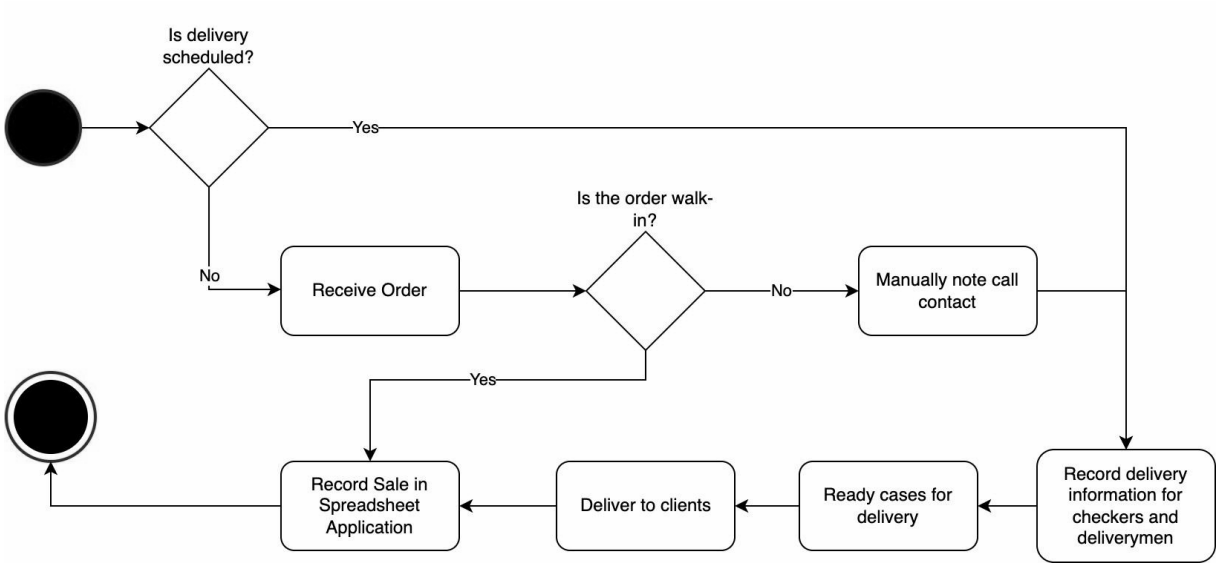


Figure 4. Current Delivery and Order Process Flowchart

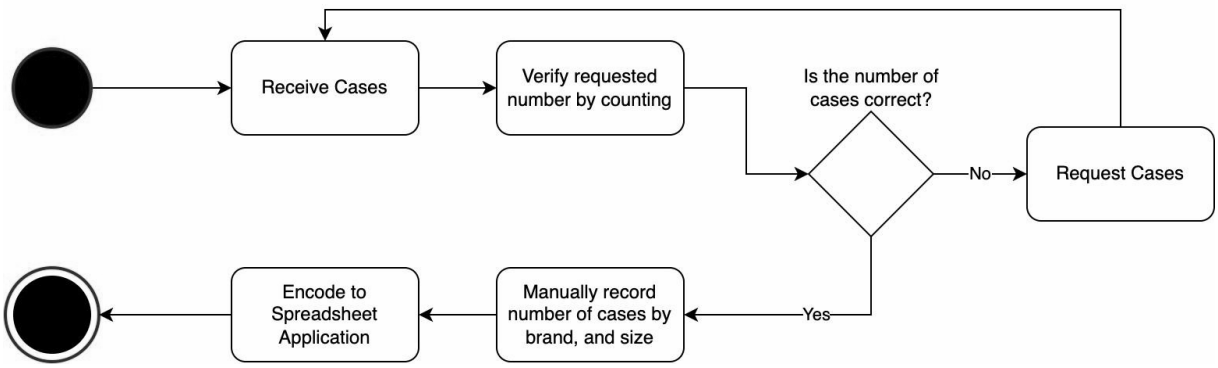


Figure 5. Current Inventory Process Flowchart