**CHAPTER I: INTRODUCTION**

In today's dynamic and competitive business landscape, the efficient management and monitoring of sales processes hold paramount significance for enterprises across various industries. The agricultural sector, in particular, is no exception to this imperative, where streamlined sales monitoring and management play a pivotal role in ensuring the growth and sustainability of farming businesses. The pig farming industry in the Philippines, one of the most common types of businesses operated by Filipinos, relies heavily on effective sales processes to cater to the country's strong preference for pork consumption (Statista, 2021). As the industry continues to evolve, the need for technologically advanced and data-driven solutions becomes increasingly evident to meet the demands of a growing market and to navigate the complexities of modern business environments.

In this context, Zarate Piggery Farm, a prominent player in the pig farming sector, seeks to address the challenges posed by its traditional manual transaction system. While it has served the farm well in its initial stages, the limitations of manual processes have become apparent as the farm expands and handles larger transaction volumes. The labor-intensive and time-consuming nature of manual record-keeping can lead to human errors, inaccuracies, and inefficient data management, ultimately hindering the farm's operational capacity and growth potential. Recognizing the need for a more streamlined, automated, and data-centric approach, Zarate Piggery Farm endeavors to embrace modern technologies and develop a comprehensive Sales Monitoring and Sales Management System with SMS Notification.

The proposed system seeks to revolutionize the farm's sales monitoring and management practices by harnessing the power of automation and real-time reporting. By adopting a user-friendly interface and integrating cutting-edge technologies such as Visual Code, SQL Server 2014 Management Studio, Bunifu Framework, and C# as the backend, the system aims to provide an efficient and secure platform for managing sales transactions, inventory, and customer interactions. The incorporation of SMS notification capabilities further enhances the farm's communication with customers, providing real-time updates on their purchases and reinforcing a sense of trust and transparency in the farm's services.

This research endeavor delves into the comprehensive development of the Sales Monitoring and Sales Management System with SMS Notification for Zarate Piggery Farm, focusing on its design, functionalities, and potential impacts on the farm's operations. Through extensive research, data analysis, and expert insights, this study aims to contribute to the broader understanding of the benefits and challenges associated with implementing such automated systems in the agricultural sector, specifically within the context of pig farming.

The significance of this study lies not only in its potential to enhance the efficiency and profitability of Zarate Piggery Farm but also in its capacity to serve as a model for other agricultural enterprises facing similar challenges. The insights and findings from this research can inform decision-makers and industry stakeholders about the advantages of adopting modern technologies to optimize sales processes, improve data accuracy, and ultimately drive growth in the highly competitive pig farming industry.

In the following sections, we will explore the rationale behind the development of the proposed system, its specific objectives, and the scope and limitations of the study. Moreover, the research methodology and the tools employed for the system development will be detailed, providing a comprehensive understanding of the study's approach. By delving into the design and functionalities of the Sales Monitoring and Sales Management System with SMS Notification, we aim to shed light on its potential to transform pig farming sales processes and contribute to the overall advancement of the agricultural sector in the Philippines.

**PROJECT CONTEXT**

Sales monitoring and management system consolidates all sales processes and performance parameters into simple reports. These can be made available to certain other departments of an organization inside the business to help them understand how their activity is converted into sales data.

One of the most common types of businesses run by Filipinos is the pig farming industry. Backyard operations up to large-scale commercial production are included in this business. One factor is responsible for the success of this company: the Filipino population has a strong preference for eating pork (Statista, 2021).

According to Yash Chawlani (2021), data is the important resource in this day and age and they hold tremendous value to what they enable to accomplish in business. Handlin, managing, analyzing and storing for use in the future is important. It all help to see the sales performance parameters, forecast sales, understand the market and achieve objective quickly as possible.

The manual transaction system in a piggery farm may have been the traditional approach, it is important to acknowledge the challenges it can pose. Human errors and inaccuracies can result in financial losses and mismanaged inventory. The manual processes can be time-consuming and labor-intensive, taking away valuable resources that could be utilized elsewhere. Moreover, as the farm expands and handles more orders, manual processes can become inefficient and hinder scalability, leading to delays and reduced productivity.

To address these challenges, it is crucial to minimize human error through automation, improving accuracy and mitigating risks. Adopting automated systems and technologies can streamline operations, enabling the farm to handle larger transaction volumes and improving overall operational capacity. Implementing an automated tracking system can enhance sales and order management, ensuring transparency and efficiency in the process.

As the Zarate Piggery Farm expands tremendously, it will eventually handle, manage, monitor, and record data manually. The system will allow the clients to view and track the process as it is filtered, qualified, and prioritized. It will allow admin to delegate responsibility for execution. Furthermore, the sale analytics is required in the system so that the owner could identify the growth and establish goals for Zarate Piggery Farm because it turns its data into insight and performance metrics shown on our system as a linear graph. Also, used to improve performance and make decisions that will benefit the business in the long run.

Therefore, the proponents propose a study that will help the said client to track their sales and manage the availability of livestock in their classification, and drives the insight record, purchased and sales. The system development may be an excellent opportunity for them to handle their business easily.

**STATEMENT OF THE PROBLEM**

Zarate Piggery Farm is currently facing challenges in managing and tracking sales due to their reliance on a manual system. The labor-intensive process of recording data in physical books consumes valuable time and increases the risk of errors and inaccuracies. Determining the price of each livestock classification and checking availability requires manual searches through multiple books and pages, resulting in inefficiency and potential delays. Additionally, the paper-based system lacks the ability to make simultaneous changes across all volumes, leading to inconsistencies and difficulties in data management. Furthermore, the reliance on physical books poses a significant risk to data security, as theft or missing pages can compromise vital sales, pricing, and livestock availability information.

The manual transaction system employed by Zarate Piggery Farm impedes their ability to effectively manage and track sales. These challenges hinder operational efficiency, increase the likelihood of errors, and compromise data security. Therefore, there is a pressing need for a modernized system that streamlines sales management, ensures accurate pricing and availability tracking, enables real-time updates, and enhances data security. Addressing these issues will significantly improve the farm's sales processes, optimize inventory management, and position Zarate Piggery Farm for long-term growth and success in the industry.

**OBJECTIVES OF THE STUDY**

**GENERAL OBJECTIVE**

Zarate Piggery Farm's sales monitoring and management system include SMS notification to oversee reports on sales monitoring, cut down on time needed to find and retrieve data, offer a backup copy of each transaction's data, and certain for each transaction.

**SPECIFIC OBJECTIVE**

* To create a module that will create user account for customized personnel;
* To create a module that will automate the classification of the livestock in terms of:

1. Piglets
2. Goods
3. Lechon Size
4. Mother Pig
5. Kalsaw/Boar

* To create a module that will notify customer for the livestock be delivered;
* To create a module that will monitor the availability of livestock; and
* To generate reports such as:
* Sales monitoring report,
* Number of available livestock, and
* Number of deliveries.

**SCOPE AND LIMITATION**

**SCOPE**

The primary focus of this study is to develop a Sales Monitoring and Sales Management System with SMS Notification for Zarate Piggery Farm. The system will be designed to handle various tasks and will be utilized by both the admin and the staff members of the farm.

The system has a login form to establish secure access and control between the admin and the establishment users. Data security will be prioritized to protect the integrity of the database. The admin has the authority to create user accounts for customized personnel, ensuring appropriate access and security. They will also be able to add stock information, define classifications for the livestock, and have access to comprehensive sales data and reports. This will enable the admin to closely monitor sales operations, track performance, and generate insightful reports for analysis.

On the other hand, the staff members will have their own set of functionalities within the system. They will be able to add customer information, place orders on behalf of customers, update stock availability, and view sales records. These features will enhance the efficiency of the sales process and enable staff members to manage orders and stocks effectively. One significant aspect of the proposed system is the integration of SMS notifications. This feature will provide automated notifications to customer of their success purchase.

Thus, Zarate’s piggery farm sales will be computed and updated automatically and dynamically every transaction. Regarding livestock, classification will be accurate, fast, and reliable due to the system’s reduced human error accuracy. It will create precise system records and give the system security to secure database data. It presents aggregated data or insight summary from all data and transactions seen. Also, has a feature of linear graph as one of the insight records.

**LIMITATION**

The proposed system for Zarate Piggery Farm is an offline or window-based system, designed to operate independently without relying on an internet connection. While this approach offers certain advantages in terms of flexibility and accessibility, it also presents limitations.

One limitation of the study is that the SMS notification feature of the system requires an internet connection to function properly. Since the system itself is designed to be offline, the SMS notification functionality may not be fully operational without internet connectivity. This limitation implies that the system's ability to provide automated notifications to customers regarding their purchases would be restricted when an internet connection is not available. Additionally, as an offline system, there might be limitations in terms of scalability and remote access.

The proposed system focuses specifically on the sales monitoring and management aspects of Zarate Piggery Farm. It does not encompass other operational areas such as inventory management, financial accounting, or marketing. Therefore, the system's scope is limited to sales and monitoring of the livestock.

**TIME AND PLACE OF THE STUDY**

The study document focuses on the Zarate Piggery Farm, located at Prk. Paglaum, Barangay San Jose, Banga, South Cotabato. This renowned piggery farm has been an integral part of the local community since its inception on December 28, 2014. The visionary behind this successful venture is none other than Lucio Zarate, affectionately known as "Boyboy" among friends, family, and associates.

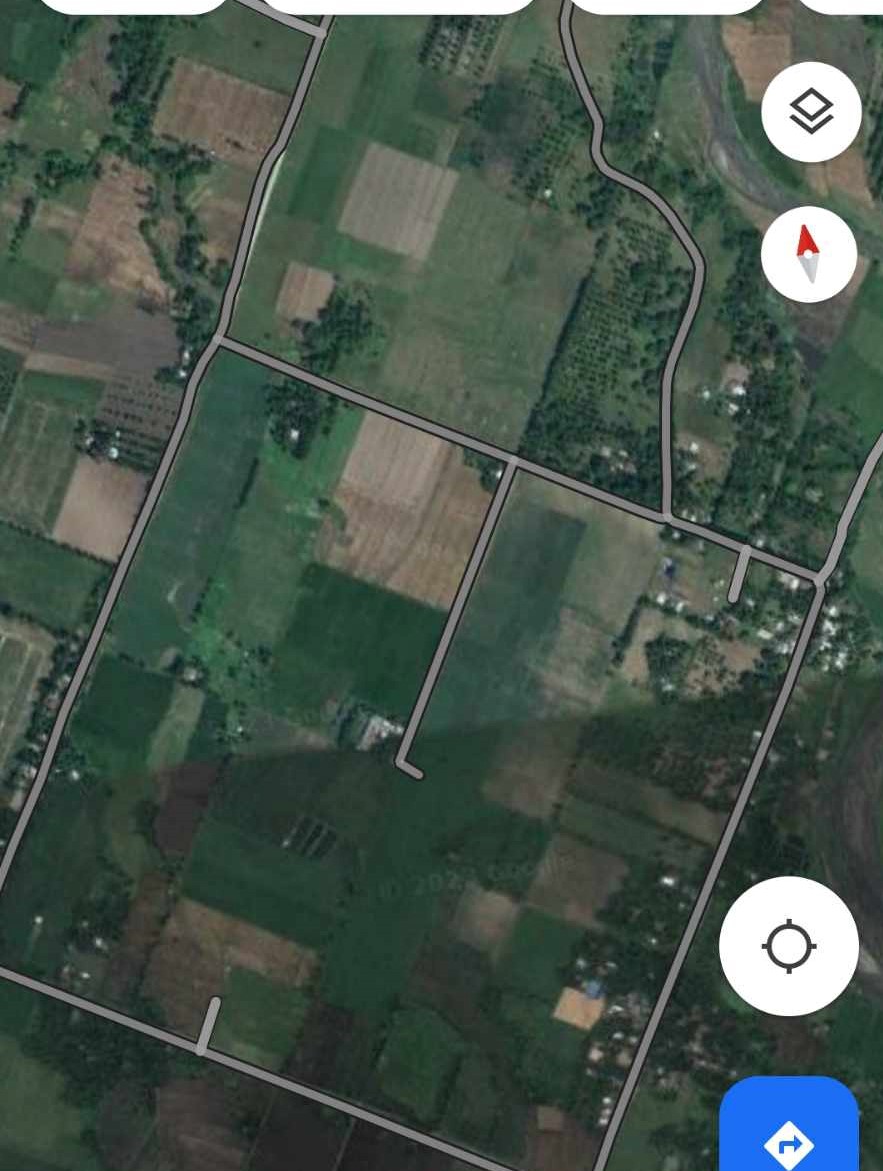
Situated amidst the picturesque landscapes of Banga, South Cotabato, Zarate Piggery Farm stands as a testament to Boyboy Zarate's dedication and passion for the agricultural sector. The farm serves as a hub for pig breeding, production, and distribution, contributing significantly to the region's livestock industry.

Since its establishment, Zarate Piggery Farm has continuously expanded, modernized, and incorporated sustainable farming practices. The facility boasts state-of-the-art infrastructure, including spacious pig pens, feed storage areas, and veterinary facilities, ensuring the well-being and health of the animals.

Boyboy Zarate's vision for the piggery farm extends beyond mere profitability. He places a strong emphasis on implementing ethical and responsible farming practices, promoting animal welfare, and prioritizing environmental sustainability. The farm employs innovative waste management systems and engages in the proper disposal and treatment of organic waste, minimizing its ecological footprint.

The community surrounding the farm has benefited greatly from Boyboy Zarate's enterprise. The establishment of Zarate Piggery Farm has created numerous employment opportunities, positively impacting the local economy and livelihoods of the residents. Additionally, the farm actively engages with the community through educational programs, sharing knowledge and expertise in pig farming techniques, and promoting sustainable agricultural practices.

Through its years of operation, Zarate Piggery Farm has garnered a reputation for producing high-quality, healthy pigs. Boyboy Zarate's meticulous attention to detail, coupled with his extensive experience in the field, has resulted in a consistent supply of top-grade livestock to the local market. The farm's products have become a preferred choice among wholesalers, retailers, and individual consumers, further solidifying its position as a leading piggery farm in the region.

 As the study document delves into the realm of Zarate Piggery Farm, it unravels the story of Boyboy Zarate's perseverance, dedication, and commitment to excellence. It showcases the significant impact of this establishment on the local community, the agricultural sector, and the broader South Cotabato region. Zarate Piggery Farm serves as an exemplary model for sustainable farming practices and a testament to the power of a passionate visionary like Boyboy Zarate.

*Figure 01. Geographical Map of the Study*

**SIGNIFICANCE OF THE STUDY**

The proposed Sales Monitoring and Management System with SMS Notification for Zarate Piggery Farm holds several significant benefits:

**To the Future Researchers/Developers –** This proposal gives an overview for future researchers and developers with a similar topic. It will also serve as a reference guide for them when they are conducting future studies and give them the idea to propose more unique service applications for the whole nation.

**To the Zarate Piggery Farm –** This will enable Zarate Piggery Farm to effectively track sales activities and manage. It offers a user-friendly interface that allows farm staff to record sales transactions, monitor stock levels, and generate comprehensive reports, all within the confines of their local network infrastructure.

**To the Pork Traders** – the propose sales monitoring and sales management system with SMS notification that will help them monitor sales and the availability of livestock and purchased.

The Proponents – the proponents will be benefited with this study because it will improve and enhance their skills to build a system with the use of modernize technology that will help the community to make their everyday life transaction easier.

**To the Future Proponents** – the propose sales monitoring and sales management system may serve as a guide or basis for the future proponents on creating a more advance and high technology integrated system that is valuable to the users.

**DEFINITION OF TERMS**

**Automated –** is operated automatically an automated process automated equipment/machinery a fully/highly automated factory.

**Database -** is an organized collection of structured information, or data, typically stored electronically in a computer system.

**Deliver –** process of providing a formal confirmation or notification to the involved parties that a transaction has been executed successfully.

**Hardware requirements -** are the requirements of a hardware device. Most hardware only has operating system requirements or compatibility.

**Insight -** is the process of creating and winning sales opportunities and driving change with ideas that matter.

**Livestock -** are farm animals, with the exception of poultry.

**Management –** is the act or manner of managing, handling, direction, or control. It also refers to a skill in managing; executive ability: great management, and tact.

**Mark up price -** the mark up price feature in the Sales Monitoring and Management System helps businesses streamline their pricing processes and ensure profitability in their sales operations.

**Monitoring -** maintain regular surveillance over.

**Peopleware requirements -** refers to the human role in an IT system. In many cases, peopleware forms a kind of conceptual triangle with hardware and software.

**Piggery Farm –** is a farm where pigs are raised or kept.

**Purchased -** Acquiring goods or services in exchange for payment.

**Sales -** refers to the collection and analysis of data related to the quantity, value, and performance of goods or services sold by a business. It involves tracking and monitoring sales transactions, such as the number of items sold, revenue generated, customer information, and other relevant metrics.

**SMS -** is a native service that allows you to receive Short Message Service

Software requirements – is a condition or capability needed by a user to solve a problem or achieve an objective.

**Stock In -** Adding inventory items to a company's stock or warehouse.

**Supplier -** provides products or services to another entity.

**Transaction –** is an exchange or interaction between people.

**Transaction Log -** A chronological record of all activities and changes related to transactions within a system or database.

**The Sale monitoring and management system with SMS notification of Zarate Piggery Farm** – are systems that both monitor and manage the operation at Zarate farm. The said system also generates report and handle transactions.

**REVIEW OF RELATED LITERATURE**

**FOREIGN LITERATURE**

According to Indeed Editorial Team (2022), sales monitoring system is a software program or a networked set of applications that tracks and controls a company's sales data. It may collect and assemble information from many sources and keep sales data about consumers and items.  Many sales monitoring solutions provide analytics or data visualization features to better understand sales patterns. There are some advantages to a sales monitoring system: An effective organization can help maintain track of sales information. Data may be organized by a sales tracking system based on several criteria, such as customers, items, or date periods. Many sales tracking systems may also pull data from other applications, allowing one to access and arrange information from different sources. Analyzing client transaction history and other data might assist in making decisions that may increase the customer base or boost sales. A sales monitoring system enables the collection and analysis of various data to understand customer demands. Many sales tracking systems include visualization tools that may help users better comprehend and show data. A sales monitoring system enables effortless sharing of sales information with other team members, supervisors, and colleagues in other departments. Keeping information readily available will assist in coordinating with other teams or individuals working in separate offices. This can help work well in groups and develop shared sales solutions. Using a sales monitoring system can assist in keeping track of your duties and clients, thereby improving the consistency and dependability of sales. Manage sales methods and track outcomes with a regular sales monitoring system. This may help to replicate previously successful sales procedures. A sales monitoring system is an easy way to keep track of a company's sales history. With data collection and analysis tools, a team may have additional time or energy to devote to other responsibilities. Using a sales tracking system may also help to simplify some activities by giving information from numerous sources.

According to Duijvenvoorde (2019), a sales tracking system is essential for any company. It will assist a manager in keeping track of the business's accomplishments and growth. These indicators are tracked in real-time via the SalesScreen dashboard. It allows

for the building of comprehensive reports. A sales monitoring system does more than track sales goals. It also motivates sales staff and make employees look forward to coming to work every day. That is what does. Managers use the software to increase sales and motivation through activities.

According to Woodruff (2018), an information system for management aids a company's competitive advantage. It reports and detects what is and is not working. These reports provide owners with the information they need to make choices and improve strategy and business performance. A business owner needs a management information system that provides data about the current activities of the company and provides the reports that allow you to keep the business on the road.

An integrated sales monitoring and management system specifically designed for retail businesses, with a case study conducted in the United States. The article discusses the development and implementation of the system, highlighting its features, benefits, and effectiveness in improving sales performance. It explores the use of advanced analytics, data integration, and visualization tools to provide real-time insights into sales data. The reference offers valuable insights into the application of sales monitoring and management systems in a foreign retail context and their impact on business operations (Johnson, M., Smith, A., Anderson, L. 2022).

**LOCAL LITERATURE**

According to Bianca (2016), a sales and inventory tracking system gathers information to help with production scheduling. Some systems, for example, utilize current sales data to anticipate how many of a particular product is required to fulfill customer’s demand shortly. This involves keeping track of a product's levels in all places. A worldwide firm with clients all over the world is an excellent example. The client may reside in another region, yet the system can determine if a product is available for delivery to the said location. The system evaluates a level, and the number slated for livestock (pig) to the number required and determines if the number of production needs is adjusted.

According to Llanto (2015), literature suggests the establishment of an efficient and effective regulatory management system (RMS). An efficient and effective RMS will be a critical mechanism for reducing the costs of doing business, facilitating international trade and investment, and improving regulatory outcomes in areas such as health, safety, and environmental protection. The paper examines the case for a regulatory management system for the Philippines and recommends specific measures for its establishment in Philippine policy space. It describes the overall experience of the country in regulatory reform, highlights the challenges in its journey toward regulatory quality and coherence, and identifies steps in constructing a responsive regulatory management system.

The said study is also comparable to the planned study in that it includes supervising the flow of products into and monitoring the condition of those items. Inventory management may be pretty valuable if done correctly. A well-functioning system is a method of controlling the movement of things in and out. Having enough commodities in the warehouse is a delicate balancing act. Effective management controls stock costs, allowing it to run a profitable firm (Hiemma, 2016).

**FOREIGN STUDIES**

**Sales Monitoring System**

According to Brown (2021), Sales monitoring may aid to organize sales operations and increasing profits: Sales automation can alleviate the stress on sales reps and facilitate their supervision by handling reports and duties. Using a software such as SalesQ, the sales staff may report from the field using their mobile devices. This promotes sales team mobility and allows managers to evaluate sales force success in real-time for improved management. Data visualization is a key aspect of sales monitoring since it displays useful data and underlying patterns that may aid in making critical selling choices and projections. Top customers and sales target achievement are two of the factors that a strong sales management software need to track for enhanced sales force monitoring. SalesQ gives more information about field sales activity through live reports filed by sales professionals from their mobile devices. Geo-fencing also helps you to validate sales rep check-ins, ensuring that salespeople are checking in from the specified region. This improves sales team efficiency by allowing you to keep an eye on the field for remote monitoring.

**Monitoring Sales System**

Kahn (2021), most significant advantages of the monitoring sales system is that it gives you with real-time information on the sales process. Sales are the lifeblood of every organization, thus it is critical to always grasp the sales potential throughout the sales channel. Organization's efficiency can be increased by implementing a systematic sales approach. Sales personnel may spend less time thinking about the next meeting and more time creating a smooth experience for each potential customer by focusing on revenue-generating tasks. Before the sales staff can engage in the process, sales monitoring tools must be in place. Sales professionals may save time by automating activities like lead qualifying and job assignment, giving them more time to advise potential clients. Monitoring the performance of remote field employees is essential for effective sales monitoring. Observation and communication with corporate officials should be possible at all times.

**Monitoring System**

According to Balmes (2016), Monitoring is the systematic collection and analysis of data for a particular activity. It is commonly used in organizations such as schools to keep track of day-to-day activities. There are times, however, when the forms are in unavailable. As a result, most teacher need help submitting their daily monitoring. In this work, the researcher creates an online day-to-day monitoring system to aid professors in submitting day-to-day monitoring quickly and on time. It will also assist the college dean in promptly checking and monitoring the submitted conventional monitoring forms because it is available at any time and from any location.

**REVIEW OF RELATED SYSTEM**

**LOCAL STUDIES**

**Monitoring and Managing System**

According to ELID (2018),Monitoring and managing attendance manually may be a time-consuming and arduous endeavor. As a result, it also ends up being highly pricey. Processing your employee's attendance takes longer. This operation involves receiving and processing time cards, making schedules, and reviewing leave and overtime requests. Getting from receiving a request to authorizing it requires multiple processes. However, with the correct HR software solution, time can be saved by setting up an automated system that handles everything for you. Most software can simply keep track of staff hours, establish timetables, and handle PTOs and sick vacations. TimeCheK by ELID is one such software solution. This adaptable tool can handle and gather data from many sites, and then utilize that information to successfully monitor staff attendance.

**Management System**

According to Bass (2016), Logistics management attempts to account for every imaginable eventuality that might affect the ultimate customer’s delivery. Originally, logistics was a military word acquiring goods and transporting them to their destinations. These same techniques have been applied by private firms to improve the efficiency of merchandise moving. Warehousing, inventory management, order fulfillment, scheduling, and coordination with other logistics providers are among these responsibilities. Customer service, coordination with manufacturers, sourcing, and procurement are all other aspects of logistics management. One of the most significant aspects of logistics management is information technology. Businesses may now locate a product at any time of day or night, in any place throughout the world, thanks to advance technology.

**Management System**

According to Carole (2017), Management System, investigates the pedagogical approach, usability and user-interface satisfaction aspect. The result showed that LMS was an effective tool to facilitate learning in an undergraduate engineering program in the Philippines because of its interactive environment and availability though it can be made more efficient by adding collaborative learning tools for students, which is deemed vital since engineering is a multidisciplinary and highly collaborative discipline.

**Inventory Management System with Real-Time Stock Tracking and Monitoring**

An inventory management system that incorporates real-time stock tracking and monitoring capabilities. The study explores the design and functionality of the system, which enables businesses to accurately track and manage their stock availability. It discusses the use of advanced technologies, to ensure accurate and up-to-date stock information. The reference provides insights into the benefits of implementing such a system, including improved inventory accuracy, reduced stockouts, and increased operational efficiency. It is a valuable resource for businesses looking to enhance their stock tracking and management processes (Chen, L., Wang, Q., Zhang, H., 2021).

**CHAPTER III**

**METHODOLOGY**

**Conceptual Framework of the Study**

The input phase or the system requirements determination consists of the following:

**User Requirements**

The proposed system collects the customer data, order data, user data, stocks and classification data such as; Full Name, Address, Contact Number, Order and classification.

**Process**

The process phase or the system development uses the Software Development Life Cycle (SDLC) which consist of the following phase;

**Requirement Analysis**

During this phase, the project proponents utilized concept mapping to outline the anticipated study outcomes. Subsequently, they commenced gathering data and information about the target client, with a primary focus on comprehending its current operations and pinpointing areas for improvement. Drawing from this research, the proponents devised inventive ideas and crafted a plan to enhance the overall situation of the said target.

To facilitate this process, the proponents collaborated with the Zarate Piggery Farm Admin, submitting a questionnaire with specific inquiries to obtain concrete and detailed insights into their operational processes. By gathering all necessary requirements, the proponents intend to develop a system that addresses the identified concerns and aligns seamlessly with the data they have collected.

**System Design**

The system design phase involves the proponents providing detailed blueprints and prototypes for the system. They develop a model and design the system's interface and functions using editing software. The model is based on the requirements gathered from the farm, and the proponents ensure that all the necessary information comes from the farm itself. Additionally, the proponents make additional changes to enable fast transactions for the Zarate Piggery Farm.

**Implementation**

The Sales Monitoring and Management System with SMS Notification will enable the management to monitor and manage their business transactions effectively. The system will be developed using the C# programming language, while the database will be managed using SQL. To ensure accuracy and precision, thorough testing and debugging will be conducted in this phase before the system is deployed. This process will guarantee that everything operates smoothly and without any issues during deployment.

**Testing**

Prior to the system deployment, the proponents prepared a comprehensive test plan to ensure a smooth and appropriate deployment process. The test plan will be presented in a tabular format and will serve as a checklist for verifying the proper functioning and achievement of all targets before the actual deployment. The main objective of the test plan is to guarantee that all the specific requirements of the Zarate Piggery Farm are fully met, providing them with the assurance of a successful implementation.

**Deployment**

During the system deployment phase, the proponents will install the system in the Zarate Piggery office. They will ensure that the system is fully functional, meeting all the requirements outlined in the test plans. Once the proponents confirm that all the necessary criteria have been met, the system will be ready for deployment.

To ensure a smooth implementation, the proponents will conduct a dry run testing, demonstrating the proper functioning of all system features and showcasing its flow. Additionally, they will take measures to secure all the files required by the farm for future reference and use.

**Maintenance**

Following the deployment, the system's maintenance will be carried out by the farm's staff and admin. They will be responsible for monitoring data and transactions regularly. Additionally, the proponents will take charge of updating the database on a monthly basis, ensuring the data's security and integrity. By dividing these tasks, the system will continue to function efficiently, and any necessary updates or security measures will be appropriately handled.

**Planning**

The proponents initiated the development of the proposed system. This involves defining the project scope, objectives, and requirements. It also includes conducting initial research, identifying potential challenges, and outlining the overall strategy to achieve the desired outcomes. The planning sets the foundation for the entire project and guides the subsequent stages of development and implementation**.**

**Output**

The final output and implementation of this study will be the Sales Monitoring and Sales Management System with SMS Notification, designed to replace the manual process of monitoring at Zarate Piggery Farm. This system will offer an automated and efficient solution for managing sales and monitoring business transactions, streamlining processes, and improving overall operational efficiency. By eliminating the need for manual monitoring, the new system aims to enhance accuracy, reduce errors, and provide real-time insights into the farm's sales and business operations.

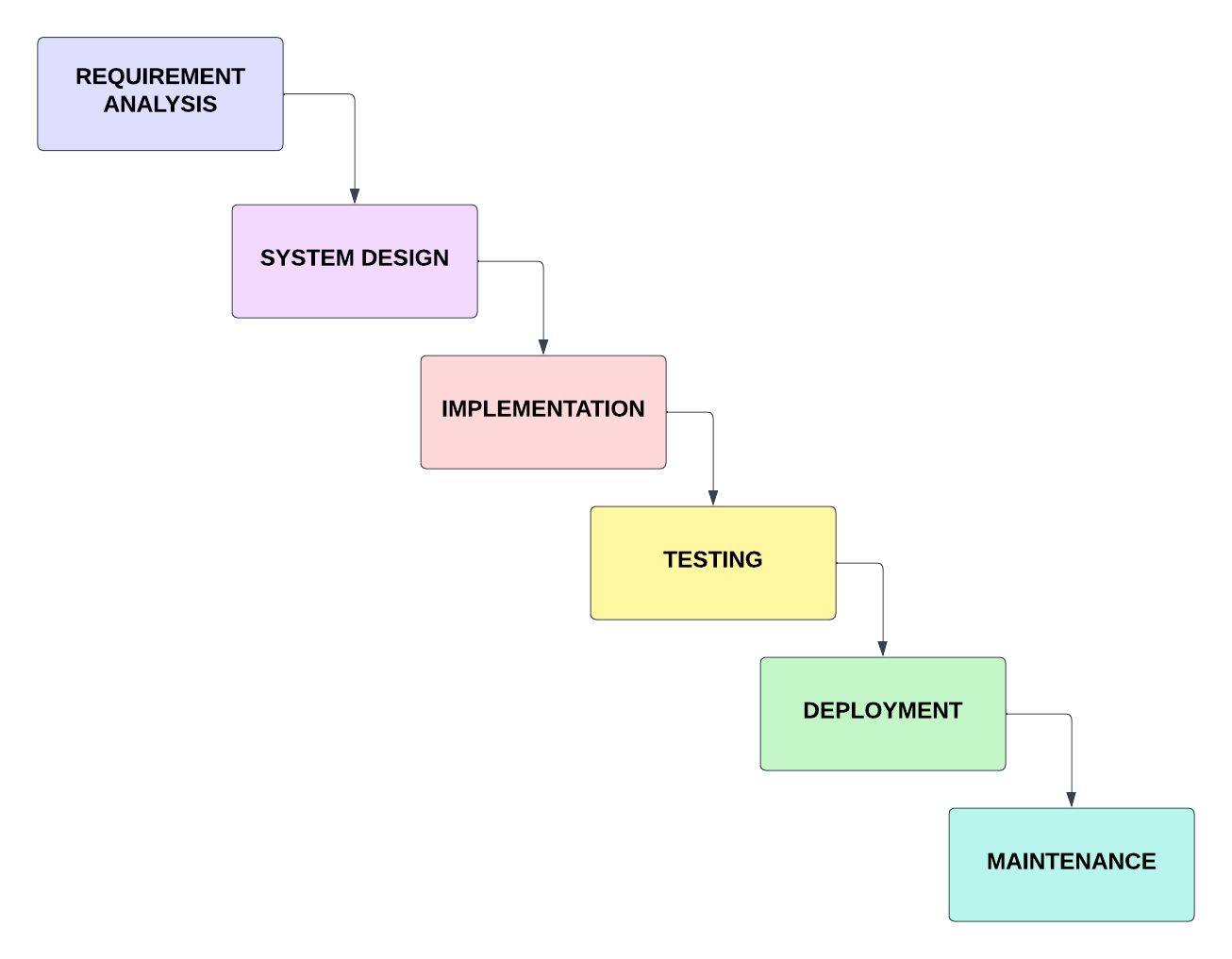
**CONCEPTUAL FRAMEWORK**

*Figure 02. Conceptual Framework of the study*

**SYSTEM DEVELOPMENT METHODOLOGY**

The System Development Life Cycle (SDLC) is a framework that outlines the various stages involved in the development of a system or software application. It provides a structured approach to the entire process, from initial planning and requirements gathering to implementation, testing, and maintenance.

The proponents use this framework to guide them through the development of the Sales Monitoring and Sales Management System for Zarate Piggery Farm. The Waterfall Method follows a sequential flow, where each phase is completed before moving to the next. This approach offers clarity and structure but may be less flexible in accommodating changes or feedback during the development process. It is most suitable for projects with well-defined and stable requirements.



*Figure 03. Waterfall Method*

**REQUIREMENTS ANALYSIS**

**System Requirement Specifications**

Specifically, the technology and its description are listed below.

|  |  |
| --- | --- |
| **TECHNOLOGY** | **DESCRIPTION** |
| **C#** | C# is a versatile and robust programming language used for building a wide variety of applications. Its combination of object-oriented features, integration with the .NET framework, cross-platform capabilities, and extensive tooling support make it a popular choice for developers worldwide. |
| **Sql** | SQL (Structured Query Language) is a standard programming language used for managing relational databases. It provides a set of commands and statements that allow users to interact with databases, perform various operations, and manipulate data. |

*Table 01. System Requirements Specifications*

**HARDWARE TECHNOLOGIES**

The hardware minimum requirements and recommendation requirements are listed below.

|  |  |  |
| --- | --- | --- |
| **Hardware** | **Minimum Requirements** | **Recommend** |
| Processor | AMD PRO A10-8770 R7, 10 COMPUTE CORES 4C+6G 3.50 GHz | AMD PRO A10-8770 R7, 10 COMPUTE CORES 4C+6G 3.50 GHz |
| Motherboard | [GIGABYTE GA-A320M-S2H](https://buildmypc.net/component-details/motherboard/B079NYQQJJ) | [GIGABYTE GA-A320M-S2H](https://buildmypc.net/component-details/motherboard/B079NYQQJJ) |
| Memory | 4.00 GB | 8.00 GB |
| Storage | 500 GB | 500 GB |
| Monitor | 14" FHD (1366 x 768) | 24" UHD (1920 x 1080) |

*Table 02. System Hardware Technologies*

**SOFTWARE REQUIREMENTS**

The table below shows the name of software and its specification.

|  |  |
| --- | --- |
| Name of Software | Specification |
| Operating System | Windows 10 Enterprise |
| Database | Sql Server 2014 Management Studio |
| IDE | Visual Studio 2015 |

*Table 03. Software Requirements*

**PEOPLEWARE REQUIREMENTS**

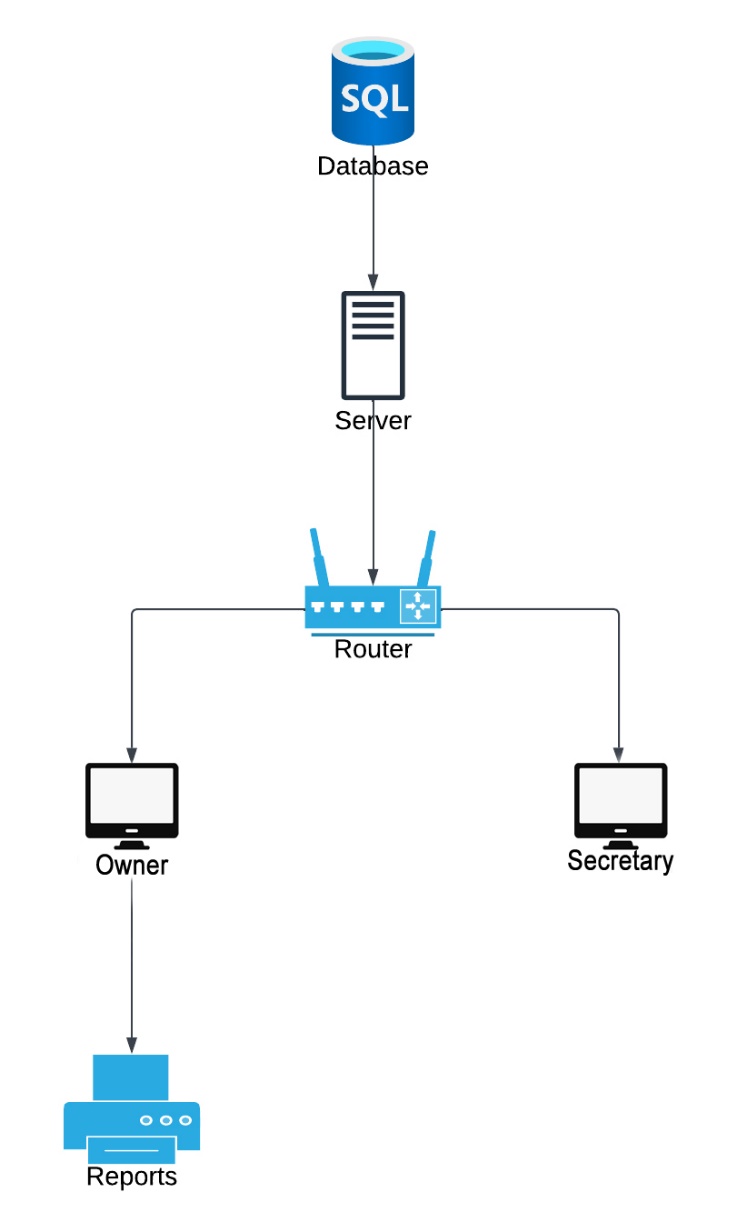
The table below shows the use of peopleware requirements and its role and their access.

|  |  |  |
| --- | --- | --- |
| **User** | **Role** | **Access** |
| Owner | Admin | - Manage Admin Page  - Manage Users Accounts  - Manage and View Customer  - Manage and View Order  - Manage Stock  - Monitor Sales  - Add Classification  - Generate Reports |
| Staff | User | - Manage and View Customer  - Manage and View Order  - Manage Stocks  - Monitor Sales |

*Table 04. Peopleware Requirements*

**NETWORK TOPOLOGY**

This figure below show that the system manages the database, while the admin device and user devices connect to the server to access their respective functions. The admin device handles tasks such as user management, data management, reporting, and system maintenance. The user devices access functionalities such as sales monitoring, stock management, sales reporting, and user account access. This topology allows for control, efficient communication, and secure data management. Additional network devices and security measures can be implemented based on system requirements.



*Figure 04. Network Topology*

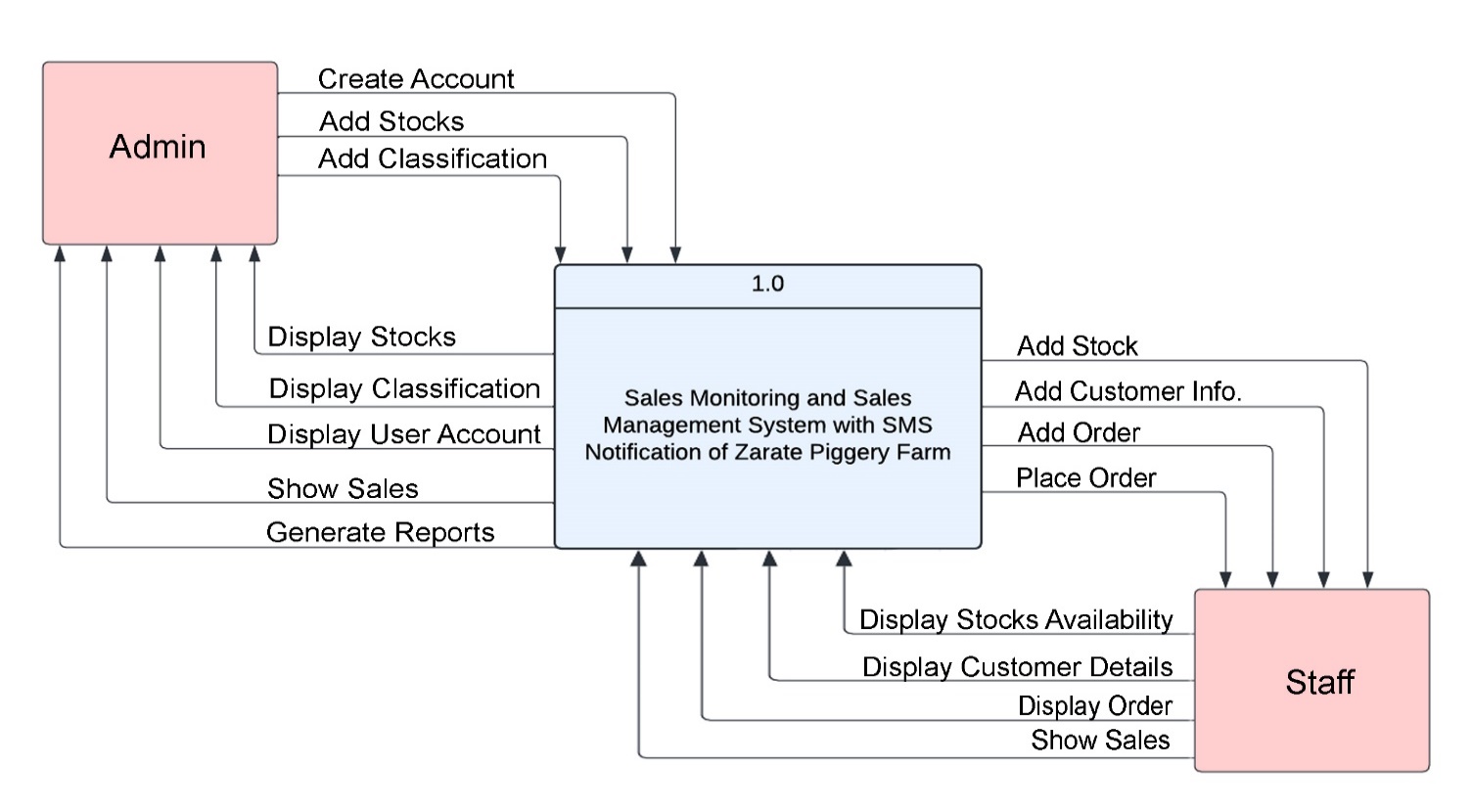
**BUDGETARY REQUIREMENTS**

**DEVELOPMENT COST**

The table below show the budgetary requirements of the development cost.

|  |  |
| --- | --- |
| **Description** | **Amount** |
| Hardware | ₱35, 715.00 |
| Semaphore for 1000 text | ₱560.00 |
| System | ₱25,000.00 |
| **TOTAL** | **₱65,275.00** |

*Table 05. Software Development Cost*

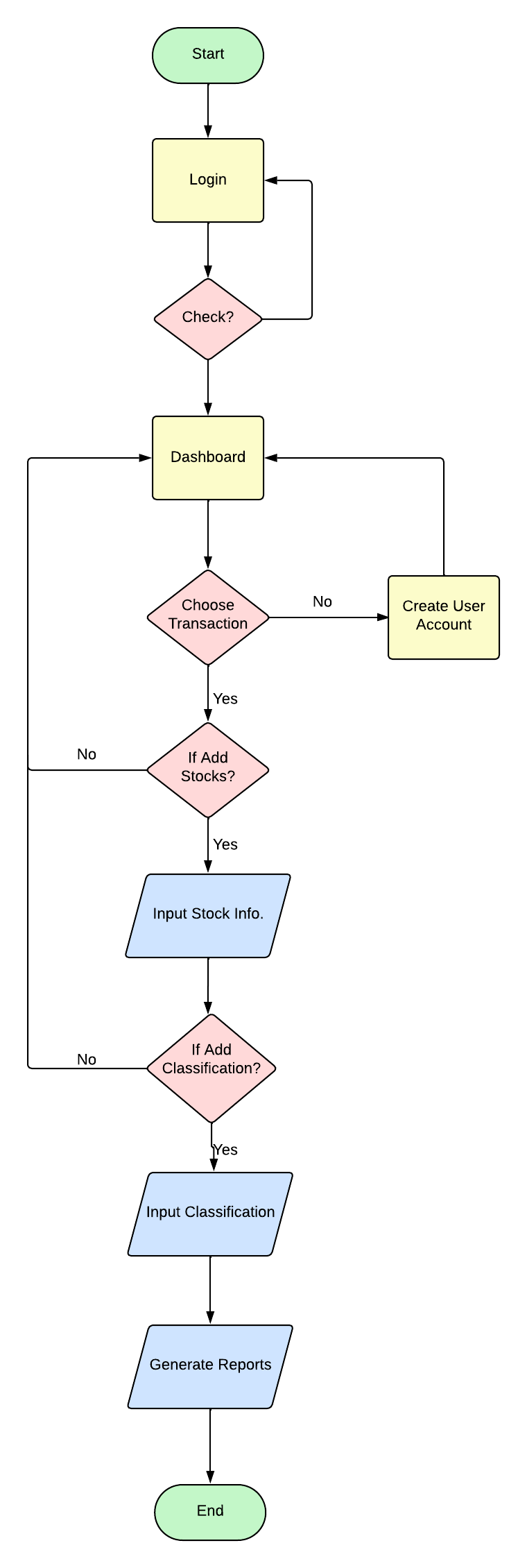
**CONTEXTUAL DIAGRAM OF THE CURRENT SYSTEM**

*Figure 05. Contextual Diagram*

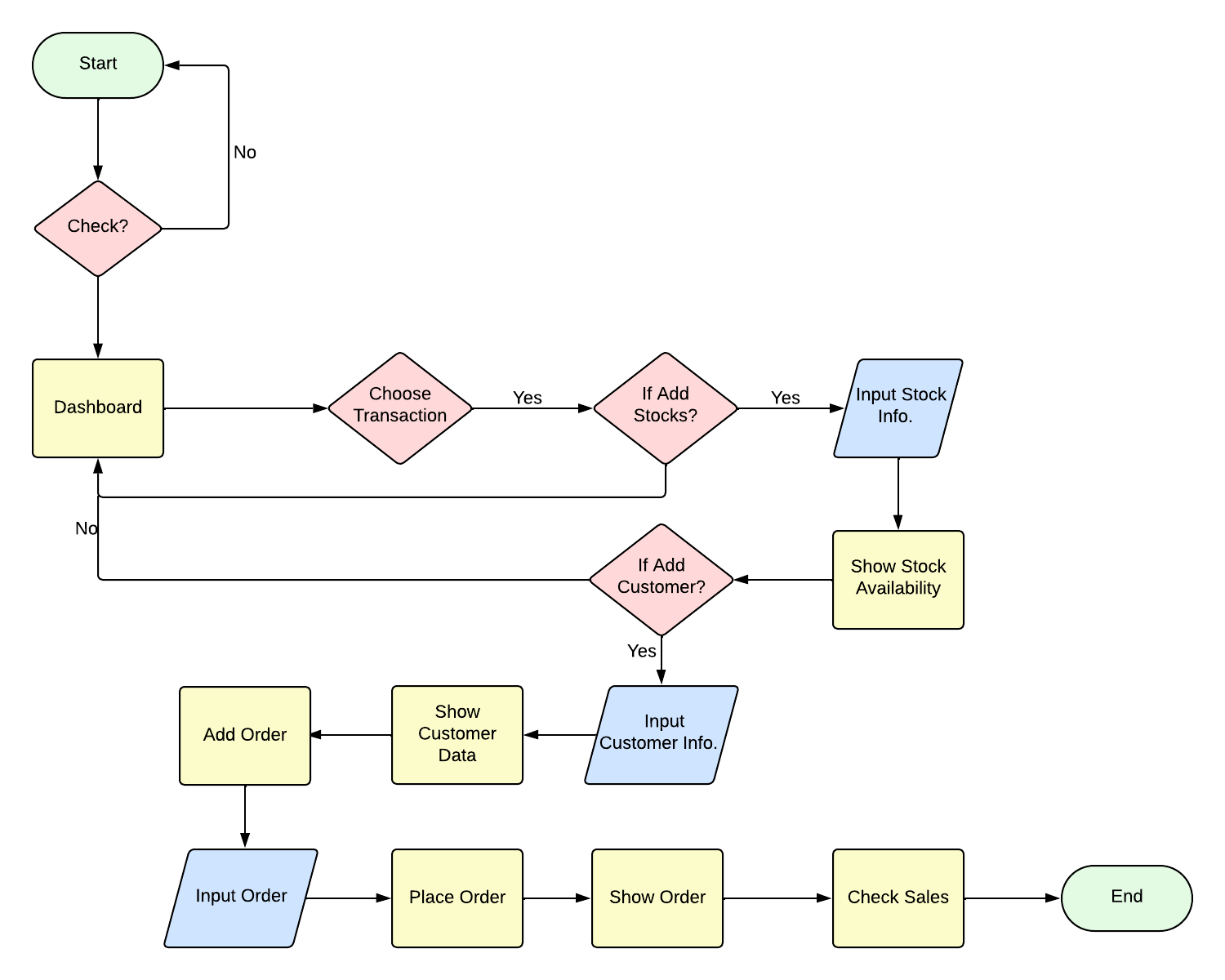
**Description:**

The contextual diagram showcases how the Sales Monitoring and Sales Management System of Zarate Piggery Farm integrates with various processes and entities to support efficient sales monitoring and management. The system allows respondents to view and track the sales process, ensuring that it is filtered, qualified, and prioritized. The Zarate Piggery Farm system monitoring in tracking sales, managing livestock availability based on classification and delivery methods, and recording valuable insights and sales data.

In the addition of contextual diagram, the Sales Monitoring and Sales Management System is depicted as the central component connecting customers, delivery services, and stock management. It serves as a hub where the sales process is tracked, data is analyzed, and decisions are made. The system's integration with these entities highlights its role in enhancing sales monitoring, improving performance, and facilitating smooth operations for Zarate Piggery Farm.

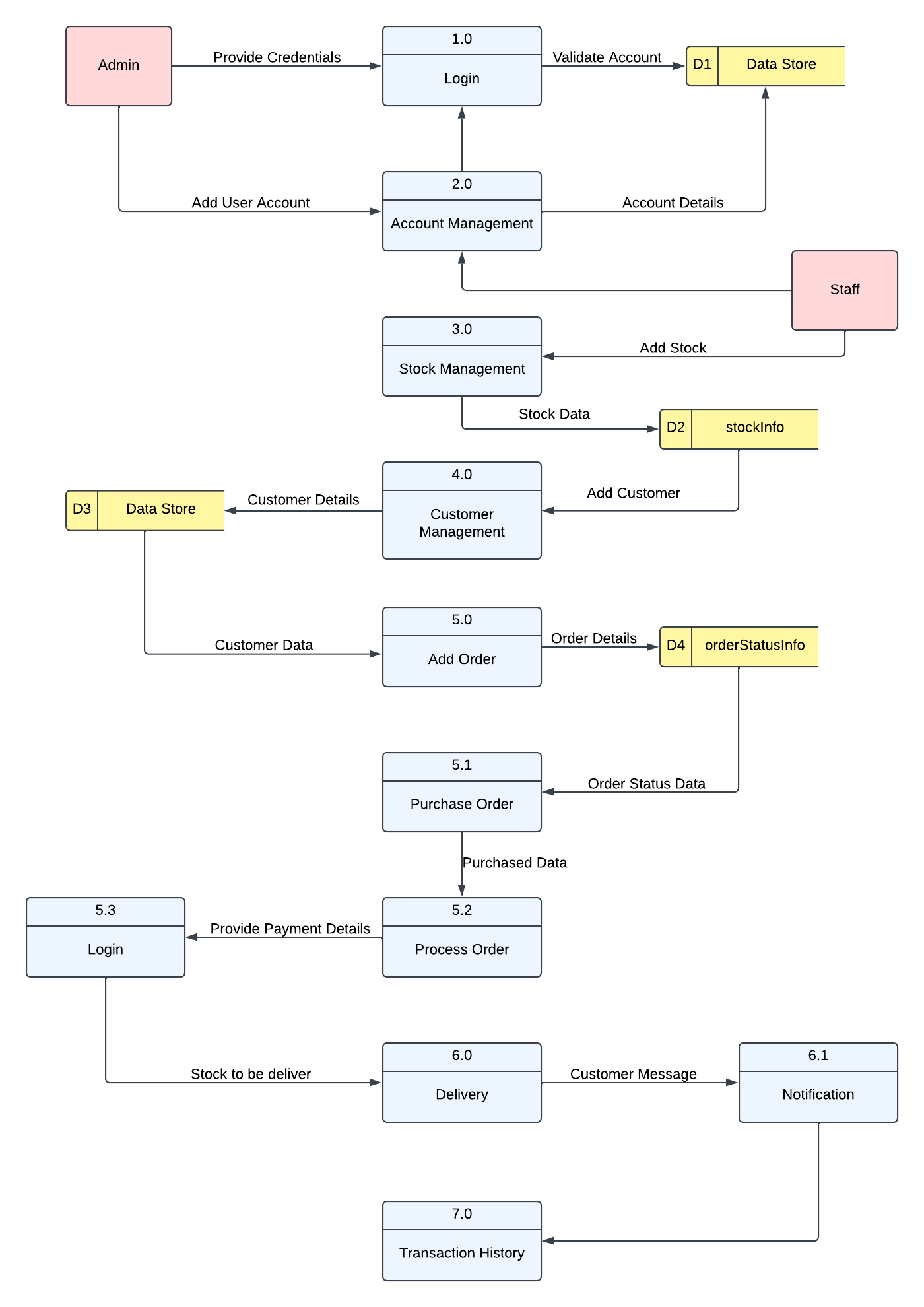
**SYSTEM FLOWCHART**

*Figure 06. Admin Flowchart*

**

*Figure 07. User Flowchart*

**LOGICAL DATAFLOW DIAGRAM OF THE PROPOSED SYSTEM**

****

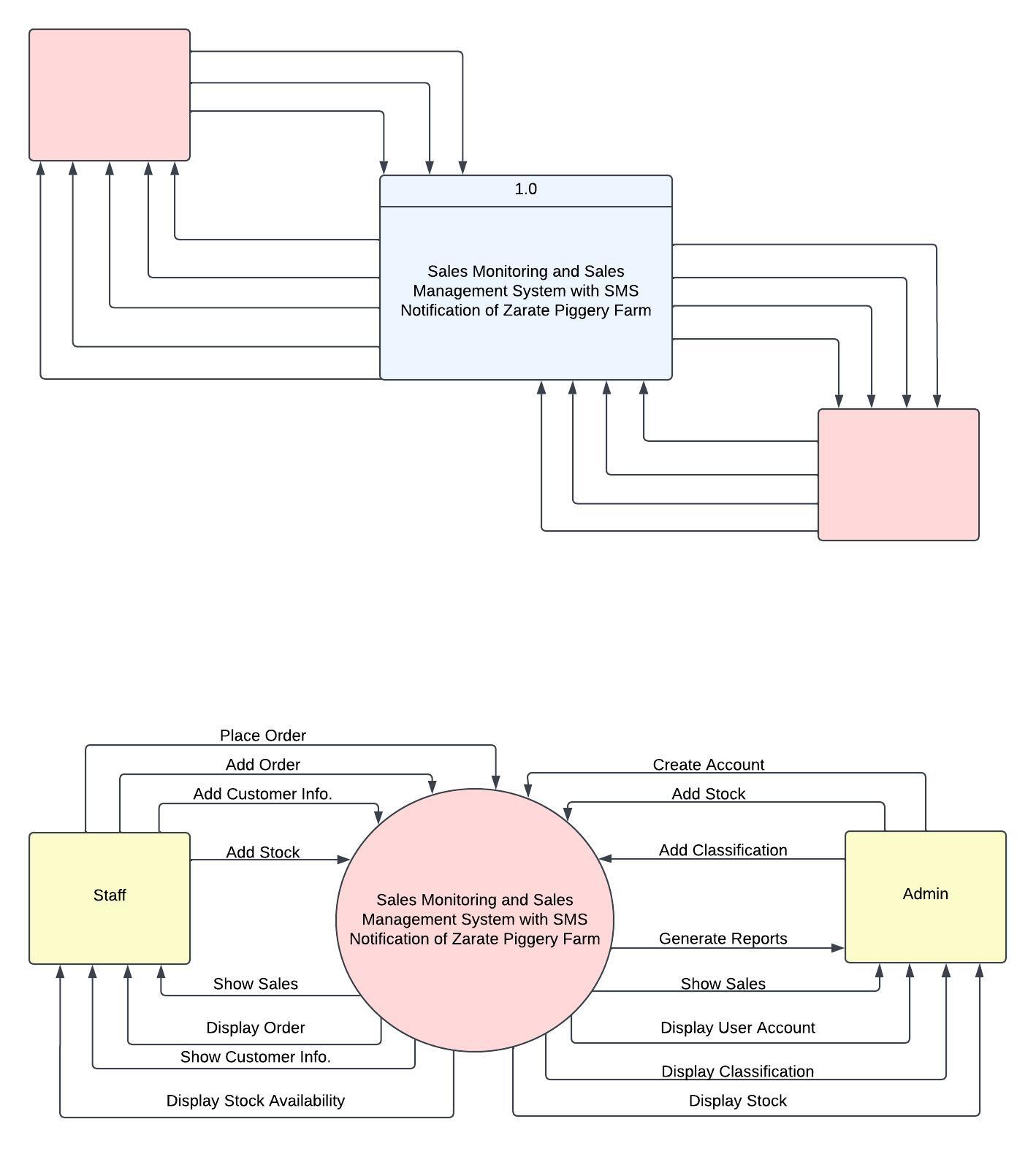
*Figure 08. Logical Dataflow Diagram*

**DESCRIPTION:**

The Sales Monitoring and Management System of Zarate Piggery Farm operates through various processes, starting with the admin validating their account credentials for system access. Once logged in, the admin can manage account and user details, as well as access customer information. Both admin and users can add orders with customer and stock details, which the system processes and records. Payment details are securely managed, with transaction logs are maintained.

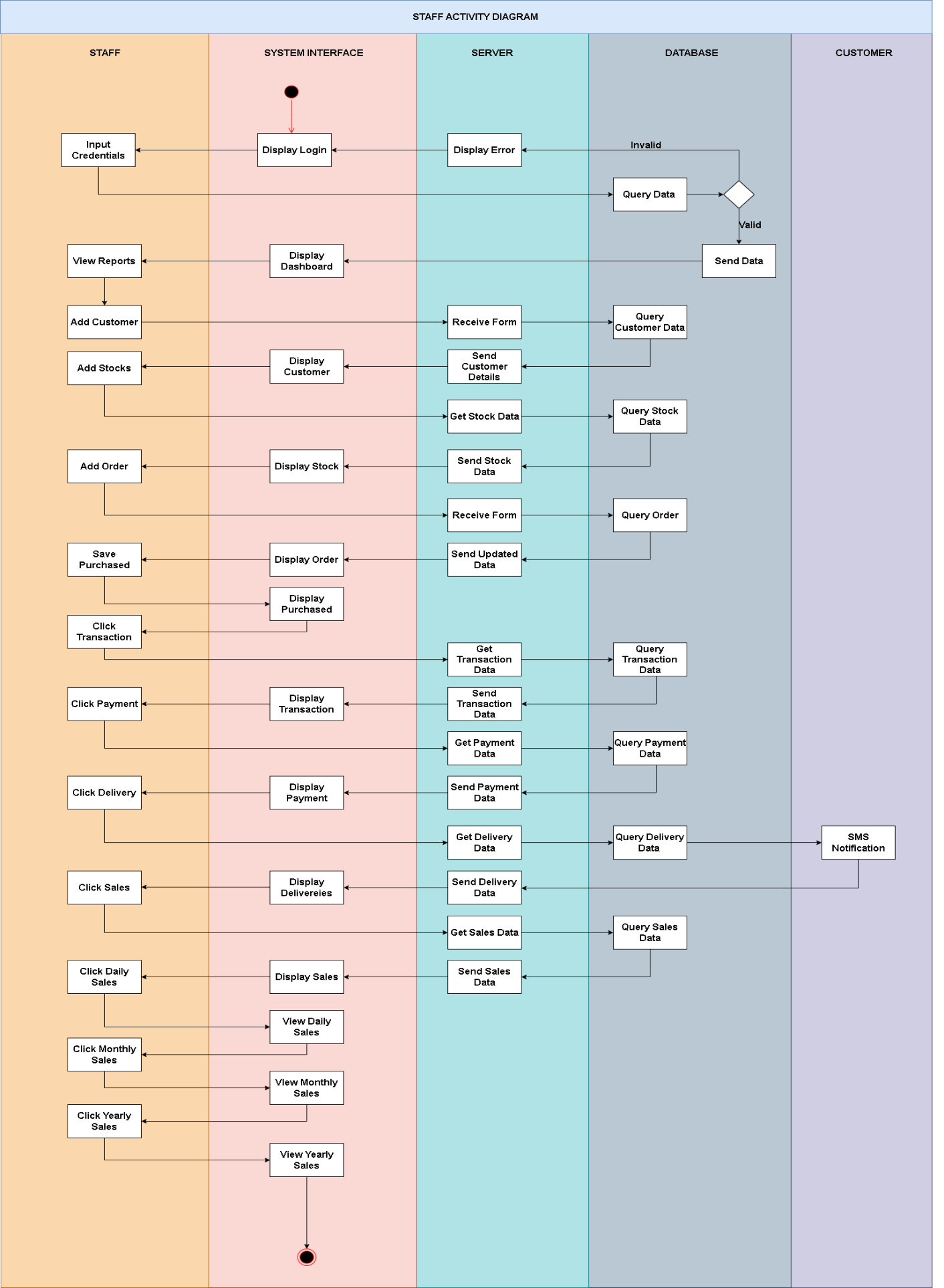
The system also handles the delivery process, tracking livestock shipments and updating delivery status. Stock management is facilitated, allowing both admin and users to add or update stock items. Overall, the system streamlines account validation, customer management, order processing, payment handling, delivery management, and stock management. This enhances efficiency and accuracy in sales monitoring and management, benefiting the operations of Zarate Piggery Farm.

**0 LEVEL DFD**

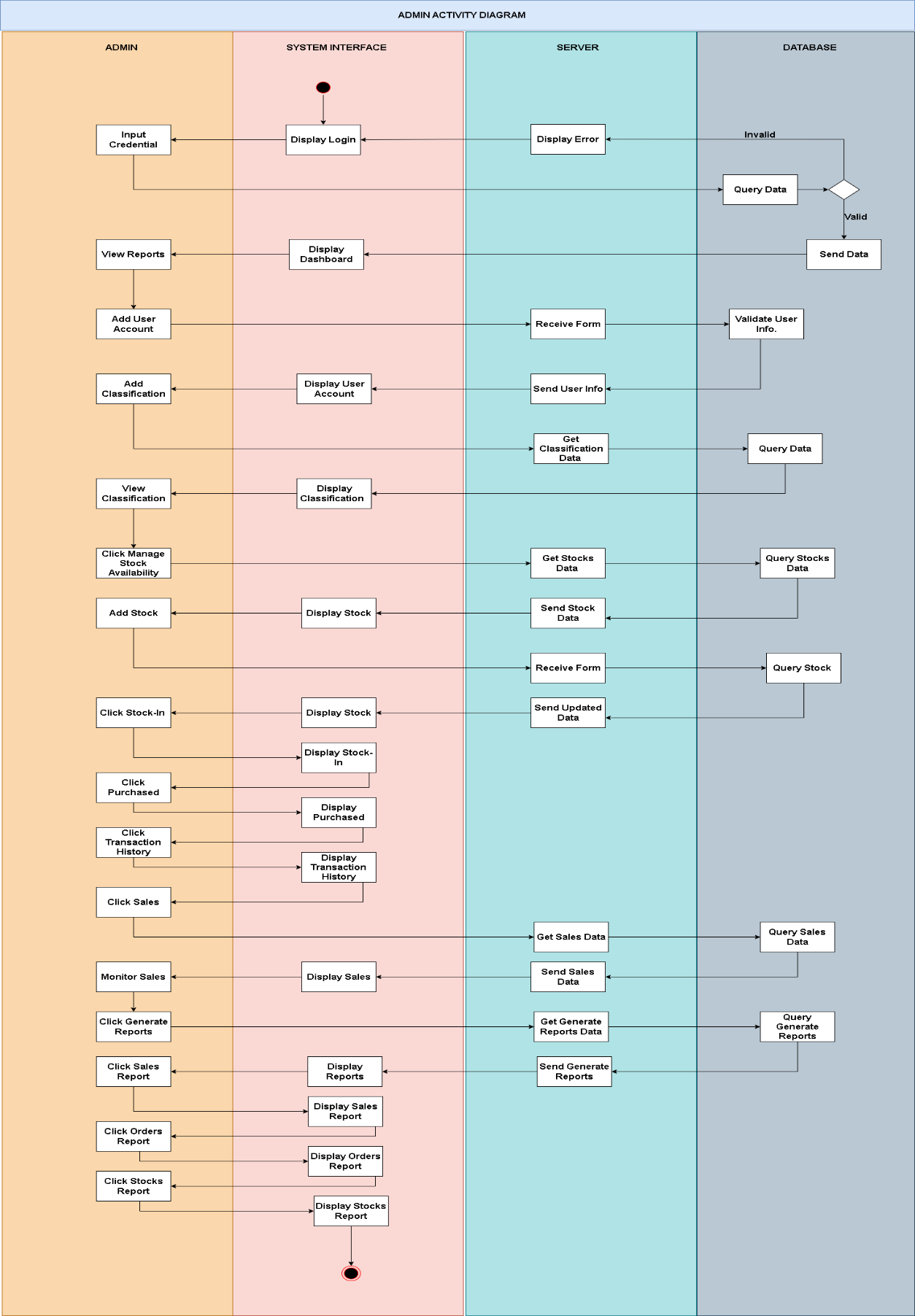
****

*Figure 09. 0 Level DFD*

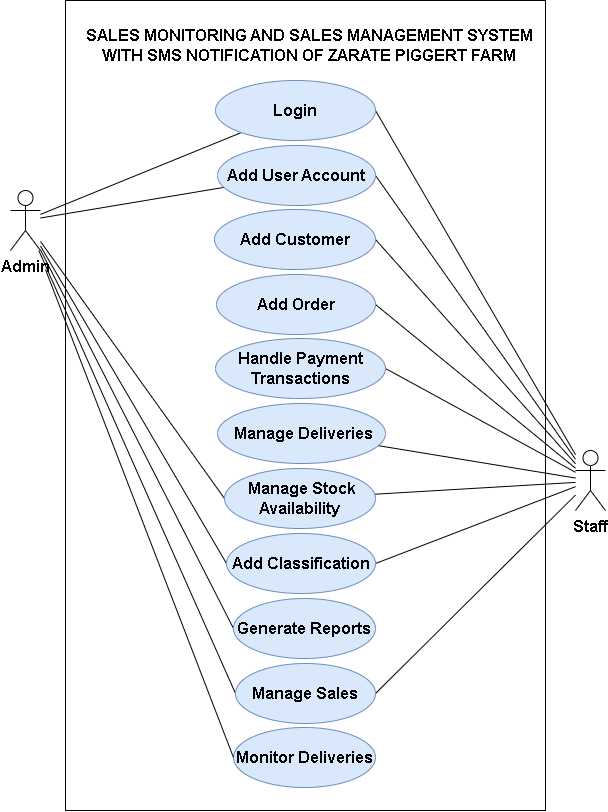
**ACTIVITY DIAGRAM**

**

*Figure 10. Admin Activity Diagram*

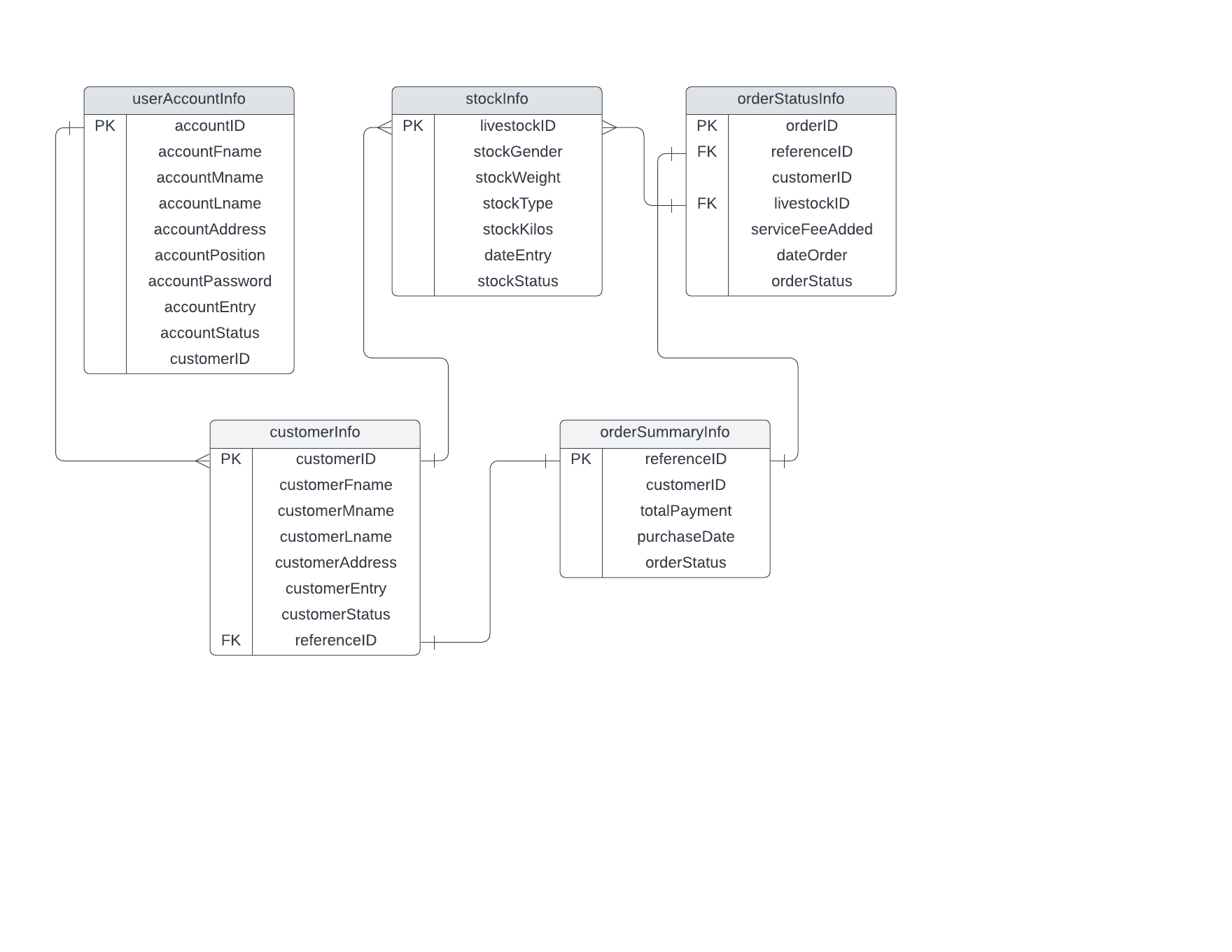


*Figure 11. Staff Activity Diagram*

**USE CASE DIAGRAM**

*Figure 12. Use Case Diagram*

**DATABASE ENTITY RELATIONSHIP DIAGRAM**

****

*Figure 13. Entity Relationship Diagram*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FIELD NAME | DATA TYPE | FIELD SIZE FOR DISPLAY | DESCRIPTION | EXAMPLE |
| classificationDescription | varchar | 50 | Livestock Classification | Piglet |
| weightFrom | Int | 10 | Minimum of Weight | 10 |
| weightTo | Int | 10 | Maximum of Weight | 20 |
| pricePerKilo | Decimal | 17, 2 | Price per kilo of Pigs | 170.00 |
| customerID | varchar | 50 | Unique ID for Customer | 202306-001 |
| customerFname | varchar | 50 | Customer First Name | Kent |
| customerMname | varchar | 50 | Customer Middle Name | Celestial |
| customerLname | varchar | 50 | Customer Last Name | Borbon |
| customerAddress | varchar | 50 | Customer Address | Banga |
| customerContact | BigInt |  | Customer Contact Number | 09262466536 |
| customerEntry | Date |  | Entry Number | 1 |
| customerStatus | varchar | 50 | Customer Status | Active |
| quantityOrder | Int | 10 | Number of Order | 100 |
| dateOrder | date | 10 | Date of Order | 18/06/2023 |
| referenceID | varchar | 50 | ID No. Reference | REF-20230618-002 |
| totalPayment | decimal | 18.2 | Amount of Payment | 25000.00 |
| totalPaid | decimal | 18.2 | Amount of Total Paid | 25000.00 |
| stockQuantity | int | 10 | Number of Stock | 150 |

**DATABASE DICTIONARY**

*Table 06. Database Dictionary*

**EXPERIMENTAL DESIGN AND TREATMENTS**

**FUNCTIONALITIES**

|  |  |
| --- | --- |
| **FUNCTION ID** | **01 – LOGIN** |
| EXPECTED RESULT | The user login successfully and proceeded to the user dashboard |
| ACTION RESULT |  |
| STATUS |  |
| TEST ENVIRONMENT | Visual Studio 2015/C#, Sql, OS: Windows 10 |

*Table 07. Functionalities of Login*

|  |  |
| --- | --- |
| **FUNCTION ID** | **02 – REDIRECT TO DASHBOARD** |
| EXPECTED RESULT | The USER is redirected to dashboard |
| ACTION RESULT |  |
| STATUS |  |
| TEST ENVIRONMENT | Visual Studio 2015/C#, Sql, OS: Windows 10 |

|  |  |
| --- | --- |
| **FUNCTION ID** | **03 – ADD CUSTOMER** |
| EXPECTED RESULT | Customer is the successful and accurate creation of a new customer record in the system's database, ensuring data validation, security, and accessibility for future interactions. |
| ACTION RESULT |  |
| STATUS |  |
| TEST ENVIRONMENT | Visual Studio 2015/C#, Sql, OS: Windows 10 |

*Table 08. Functionalities of Dashboard*

*Table 09. Functionalities of Customer*

|  |  |
| --- | --- |
| **FUNCTION ID** | **04 – ORDER MANAGEMENT** |
| EXPECTED RESULT | Order management table is to have a well-organized and updated record of all orders, including relevant details such as order numbers, customer information, order items, quantities, prices, dates, and status. |
| ACTION RESULT |  |
| STATUS |  |
| TEST ENVIRONMENT | Visual Studio 2015/C#, Sql, OS: Windows 10 |

*Table 10. Functionalities of Order Management*

|  |  |
| --- | --- |
| **FUNCTION ID** | **05 – ORDER FORM** |
| EXPECTED RESULT | Collect and record the necessary information such as order number, customer details, order items, quantities, prices, dates, and status, ensuring the accuracy of data entry and updating the table accordingly |
| ACTION RESULT |  |
| STATUS |  |
| TEST ENVIRONMENT | Visual Studio 2015/C#, Sql, OS: Windows 10 |

*Table 11. Functionalities of Order Form*

|  |  |
| --- | --- |
| **FUNCTION ID** | **06 – TRANSACTION MODULE** |
| EXPECTED RESULT | the system manages and facilitates various financial activities, including processing, recording, and verifying transactions, supporting multiple payment methods, maintaining transaction status and history, and generating reports for analysis and auditing purposes. |
| ACTION RESULT |  |
| STATUS |  |
| TEST ENVIRONMENT | Visual Studio 2015/C#, Sql , OS: Windows 10 |

*Table 12. Functionalities of Transaction Module*

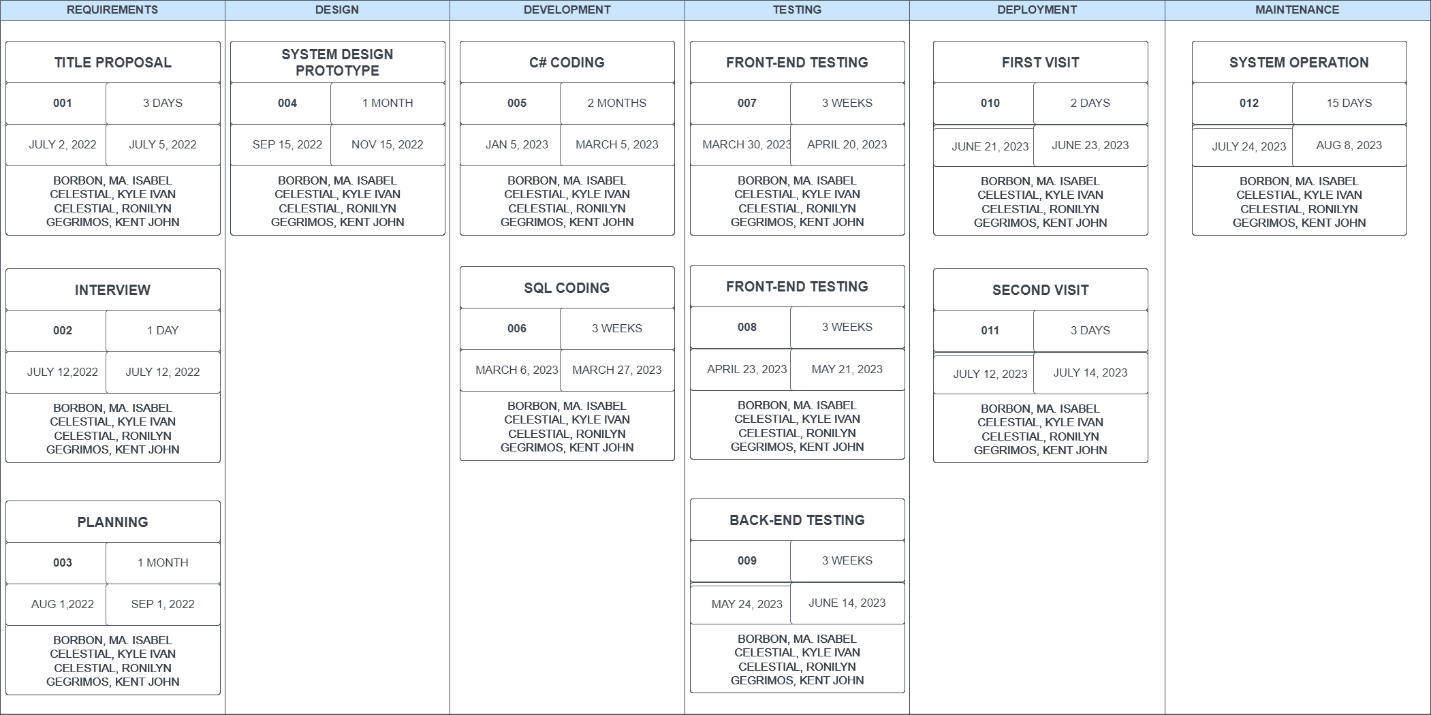
|  |  |
| --- | --- |
| **FUNCTION ID** | **06 – STOCK MANAGEMENT** |
| EXPECTED RESULT | Efficiently record inventory levels, monitors stock movement, and provides real-time data on stock availability. |
| ACTION RESULT |  |
| STATUS |  |
| TEST ENVIRONMENT | Visual Studio 2015/C#, Sql, OS: Windows 10 |

|  |  |
| --- | --- |
| **FUNCTION ID** | **07 – CLASSIFICATION** |
| EXPECTED RESULT | Organizes and categorizes data, documents, or entities based on predefined criteria, facilitating easy retrieval, analysis, and management of information. |
| ACTION RESULT |  |
| STATUS |  |
| TEST ENVIRONMENT | Visual Studio 2015/C#, Sql , OS: Windows 10 |

*Table 13. Functionalities of Stock Management*

*Table 07. Functionalities of Classification*

**PERT CHART**

****

*Figure 14. Pert Chart*

**CHAPTER III: RESULTS**

**QUESTIONNAIRE FOR THE MANUAL PROCESS**

**NAME: DATE:**

**POSITION:**

**INSTRUCTION:**

Check (🗸) the appropriate box to the right of the corresponding statement to indicate your agreement or disagreement with each of the following statements.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **QUESTIONS** | **Strongly Agree** | **Agree** | **Strongly Disagree** | **Disagree** |
| 1 | The manual process for recording sales in the piggery farm prone to errors and inconsistencies. |  |  |  |  |
| 2 | The manual process for tracking sales in the piggery farm is time consuming and requires significant effort. |  |  |  |  |
| 3 | The manual process for tracking sales in the piggery farm is time-consuming and requires significant effort. |  |  |  |  |
| 4 | The current manual process for sales monitoring in the piggery farm hinders timely decision-making. |  |  |  |  |
| 5 | The manual process for sales monitoring does not provide real-time insights into sales performance. |  |  |  |  |
| 6 | The piggery farm would benefit from a system that automates the sales monitoring process. |  |  |  |  |
| 7 | A computerized system for sales monitoring would help improve the accuracy of sales data. |  |  |  |  |
| 8 | An automated system for sales monitoring would enhance the efficiency of the sales process in the piggery farm. |  |  |  |  |
| 9 | A digital sales monitoring system would enable real-time tracking of sales performance. |  |  |  |  |
| 10 | The piggery farm should invest in a sales monitoring system to overcome the limitations of the manual process. |  |  |  |  |

Comment: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SALES MONITORING AND SALES MANAGEMENT SYSTEM WITH SMS NOTIFICATION OF ZARATE PIGGERY FARM**

**QUESTIONNAIRE FOR THE SYSTEM USER**

**NAME: DATE:**

**POSITION:**

**INSTRUCTION:**

Check (🗸) the appropriate box to the right of the corresponding statement to indicate your agreement or disagreement with each of the following statements.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **QUESTIONS** | **Strongly Agree** | **Agree** | **Strongly Disagree** | **Disagree** |
| 1 | The sales monitoring and management system provided accurate information about the number of pigs sold. |  |  |  |  |
| 2 | The system effectively tracked the sales revenue generated by Zarate Piggery Farm. |  |  |  |  |
| 3 | The system provided timely and detailed reports on the sales performance of the farm. |  |  |  |  |
| 4 | The system allowed for easy identification of the customers who purchased the pigs. |  |  |  |  |
| 5 | The system provided insights into the profitability of different pig breed |  |  |  |  |
| 6 | The system helped in identifying and addressing any issues or challenges in the sales process. |  |  |  |  |
| 7 | The system enabled the farm to forecast sales |  |  |  |  |
| 8 | The system enabled the farm to analyze the sales data and make data-driven decisions. |  |  |  |  |
| 9 | The system facilitated effective management of the available pigs for sale. |  |  |  |  |
| 10 | The sales monitoring and management system presented to Zarate Piggery Farm has been effective in enhancing sales performance and improving overall efficiency. |  |  |  |  |

Comment: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CHAPTER IV: ANALYSIS AND DISCUSSION**

The analysis and discussion provided highlight the importance and benefits of developing a Sales Monitoring and Sales Management System with SMS Notification for Zarate Piggery Farm. The manual transaction system that the farm currently relies on poses various challenges, including labor-intensive processes, increased risk of errors, inefficiencies in data management, and potential data security risks.

The proposed system offers significant improvements by automating sales processes, enhancing data accuracy, and providing real-time reporting. By adopting Visual Code as the text editor, SQL Server 2014 Management Studio as the database, Bunifu Framework for UI enhancement, and C# as the backend, the system is equipped with the necessary tools to create a user-friendly and efficient interface.

The system's ability to automate livestock classification, track availability, and provide SMS notifications to customers about their purchases is especially valuable for a business like Zarate Piggery Farm. These features can streamline operations, improve customer satisfaction, and enable the farm to make data-driven decisions.

Furthermore, the system's use of SQL for data storage ensures that all transactions and information are securely recorded and can be easily accessed for analysis and reporting. By eliminating manual record-keeping, the system minimizes human errors and saves valuable time and resources.

The enthusiasm displayed by the management and staff of Zarate Piggery Farm towards the implementation of the proposed Sales Monitoring and Sales Management System reflects their proactive approach in embracing technological innovation. This receptiveness to adopting new systems showcases a commitment to continuous improvement in their business operations, underscoring their readiness to leverage technology's potential benefits for enhanced efficiency and competitiveness.

Overall, the proposed Sales Monitoring and Sales Management System with SMS Notification appears to be a valuable solution to address the challenges faced by Zarate Piggery Farm. With its implementation, the farm can enhance its sales processes, optimize inventory management, and position itself for growth and success in the pig farming industry. The use of modern technologies and automation aligns well with the importance of data in business, as emphasized by Yash Chawlani's statement in the introduction.

As the system is implemented, it will be essential for the proponents to provide adequate training and support to the farm's staff to ensure a smooth transition and efficient utilization of the system. Regular updates and maintenance will also be crucial to keep the system running smoothly and to address any potential issues that may arise.

In conclusion, the proposed system has the potential to revolutionize sales monitoring and management for Zarate Piggery Farm and improve their overall business efficiency. By leveraging automation, real-time reporting, and data accuracy, the farm can make informed decisions and stay competitive in the Filipino pig farming industry.

**APPENDICES**

**APPENDIX A - USER MANUAL**

User Manual: Sale Monitoring and Sales Management System with SMS Notification of Zarate Piggery Farm

Table of Contents:

**1. Introduction**

- 1.1 About the System

- 1.2 Key Features

- 1.3 System Requirements

**2. Getting Started**

- 2.1 Installation

- 2.2 User Registration

- 2.3 Logging In

**3. Dashboard Overview**

- 3.1 Sale Overview

- 3.2 Stock Status

- 3.3 Notifications

**4. Sale Monitoring**

- 4.1 Adding a New Sale

- 4.2 Viewing Sale Details

- 4.3 Editing and Deleting Sales

**5. Sales Management**

- 5.1 Customer Management

- 5.2 Product Management

- 5.3 Price Management

**6. SMS Notification Setup**

- 6.1 Phone Number Verification

- 6.2 Setting Notification Preferences

**7. Reports and Analytics**

- 7.1 Sales Reports

- 7.2 Stock Reports

- 7.3 Customer Reports

**8. Troubleshooting**

- 8.1 Common Issues

- 8.2 Frequently Asked Questions

- 8.3 Contact Support

---

**1. INTRODUCTION**

**1.1 \*\*About the System\*\***

The Sale Monitoring and Sales Management System with SMS Notification of Zarate Piggery Farm is an advanced software solution designed to streamline and optimize sales-related processes in the piggery farm. This system allows the farm to monitor sales activities, manage customer information, track product stock, and send SMS notifications to customers in the delivery.

**1.2 \*\*Key Features\*\***

- Intuitive Dashboard: Provides an overview of sales, stock status, and insights.

- Sale Monitoring: Easily add, view, and edit sales record

- Sales Management: Efficiently manage customer information, product details, and pricing.

- SMS Notifications: Send automated SMS updates to customers for order status of delivery.

- Reports and Analytics: Generate various sales, stock, and customer reports for analysis.

**1.3 \*\*System Requirements\*\***

To run the Offline Version of the Sale Monitoring and Sales Management System, ensure your system meets the following requirements:

* Operating System: Windows 10 or later
* Processor: Intel Core i5 or equivalent AMD processor
* RAM: 8GB or higher
* Hard Disk Space: 500MB or more

**2. GETTING STARTED**

**2.1 \*\*Installation\*\***

To install the system, download the installation file from the given source. Double-click the file and follow the on-screen instructions to complete the installation process.

**2.2 \*\*User Registration\*\***

During the installation, you will be given the userID and the password.

**2.3 \*\*Logging In\*\***

After that, log in using your userID and password. You can now access the system's features and functionalities offline. The system will redirect you to the dashboard.

**3. DASHBOARD OVERVIEW**

**3.1 \*\*Sale Overview\*\***

The dashboard provides a summary of recent sales activities, including total sales, number of stock availability, and a sales graph.

**3.2 \*\*Stock Status\*\***

Monitor the current stock status of products to ensure efficient inventory management.

**4. SALE MONITORING**

**4.1 \*\*Adding a New Stock\*\***

To record a new sale, navigate to the "Stock" section and enter the relevant details.

**4.2 \*\*Viewing Sale Details\*\***

Access the "Stock" section to view a list of all recorded sales. Click on a sale to see its detailed information.

**5. ADDING USER ACCOUNT**

To create a new user account in the Sale Monitoring and Sales Management System, use the following steps:

* Access the "User Account" section from the dashboard.
* Click on "Add User"
* Enter the required details for the new user, including name, and a strong password.
* Assign appropriate user roles and permissions based on their responsibilities.
* Save the user account, and the new user can now log in with the provided credentials.
* Remember to manage user accounts with care and grant access only to authorized personnel to maintain the system's security and integrity.

**6. ADDING ORDER**

To add an order in the Sale Monitoring and Sales Management System, follow these steps:

* Navigate to the "Add Order" section on the dashboard.
* Select the customer for whom you want to create the order or add a new customer by providing their details (name, contact information).
* Choose the products the customer is ordering from the product list. Specify the quantity for each product.
* Enter the payment details, such as the payment method and total amount paid.
* Click the "Add Order" button to save the order.

**7. ADDING CUSTOMER**

To add a new customer to the Sale Monitoring and Sales Management System, use the following steps:

* Go to the "Customer Management" section on the dashboard.
* Click on "Add Customer" or "Create New Customer."
* Enter the customer's details, including:
  + Full Name: The customer's complete name.
  + Contact Information: Phone number, email address, or any other relevant contact details.
  + Address: The customer's physical address or location.
  + Click the "Save" button to create the new customer record.

**8. ADDING CLASSIFICATION**

To add a new classification to the products in the system, do the following:

* Go to the "Classification" section on the dashboard.
* Click on "Add Classification"
* Provide a name and description for the new classification.
* Save the classification.

**9. SALES MANAGEMENT**

**9.1 \*\*Customer Management\*\***

Manage customer information, including names, contact details, and order history.

**10. REPORTS**

**10.1 \*\*Sales Reports\*\***

Generate detailed sales reports based on custom date ranges, customers, or products.

**10.2 \*\*Stock Reports\*\***

View reports showing product stock levels, availability of livestock and reorder recommendations.

**10.3 \*\*Customer Reports\*\***

Analyze customer data, purchase history, and trends to improve marketing strategies.

**10.4 \*\*No. of Deliveries Reports\*\***

Provide a comprehensive overview of the total number of deliveries made during the specified period.

**11. TROUBLESHOOTING**

**11.1 \*\*Common Issues\*\***

Refer to this section for solutions to common problems encountered while using the system.

**11.2 \*\*Frequently Asked Questions\*\***

Find answers to frequently asked questions about the system's features and functionalities.

**11.3 \*\*Contact Support\*\***

If you encounter technical difficulties or have specific inquiries, contact our support team for assistance.

---

Note: This user manual is a general guide for the Sale Monitoring and Sales Management System with SMS Notification of Zarate Piggery Farm. Actual features and interfaces may vary slightly depending on the specific implementation and updates of the system. For further details or specific instructions, refer to the system's official documentation or contact the Zarate Piggery Farm support team.

**APPENDIX B – SOURCE CODE**

**Login**

private void bunifuFlatButton1\_Click(object sender, EventArgs e)

{

SqlConnection conn = new SqlConnection();

conn.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd = new SqlCommand();

cmd.Connection = conn;

conn.Open();

cmd.CommandText = "SELECT \* FROM userAccountInfo WHERE accountID= '" + txtLogin.Text + "' AND accountPassword = '" + txtPassword.Text + "' COLLATE Latin1\_General\_CS\_AS";

SqlDataAdapter adap = new SqlDataAdapter(cmd);

DataTable dat = new DataTable();

adap.Fill(dat);

if (dat.Rows.Count > 0)

{

for (int i = 0; i < dat.Rows.Count; i++)

{

if (dat.Rows[i]["accountPosition"].ToString() == "ADMIN")

{

ADMIN.adminDashboard frm = new ADMIN.adminDashboard();

this.Hide();

frm.ShowDialog();

}

if (dat.Rows[i]["accountPosition"].ToString() == "STAFF")

{

if (dat.Rows[i]["accountStatus"].ToString() == "ACTIVE")

{

SECRETARY.secDashboard frm = new SECRETARY.secDashboard();

this.Hide();

frm.ShowDialog();

}

else

{

MessageBox.Show("USER INACTIVE");

}

}

}

}

else

{

MessageBox.Show("INVALID ACCOUNT");

}

}

**addAccountPanel**

private void addAccountPanel\_Load(object sender, EventArgs e)

{

String date = DateTime.Now.ToString("yyyyMM");

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd1 = new SqlCommand();

cmd1.Connection = con;

con.Open();

cmd1.CommandText = "SELECT COUNT(\*) FROM userAccountInfo;";

int num1 = Convert.ToInt32(cmd1.ExecuteScalar());

int res = num1 + 1;

txtacc.Text = date + "-" + res;

}

private void bunifuFlatButton1\_Click(object sender, EventArgs e)

{

this.Hide();

}

private void bunifuFlatButton2\_Click(object sender, EventArgs e)

{

if (txtFname.Text.Equals("") || txtMname.Text.Equals("") || txtLname.Text.Equals("") || txtAddress.Text.Equals("") || txtPassword.Text.Equals(""))

{

MessageBox.Show("Please Fill In all the fields");

}

else

{

DialogResult result = MessageBox.Show("Are you sure you want to save the data?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

String date = DateTime.Now.ToString("yyyy-MM-dd");

//String date1 = DateTime.Now.ToString("yyyy");

Random rand = new Random();

int number = rand.Next(0, 100);

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "Insert into userAccountInfo VALUES('" + txtacc.Text + "','" + txtFname.Text + "','" + txtMname.Text + "','" + txtLname.Text + "','" + txtAddress.Text + "','" + txtJobRole.selectedValue + "','" + txtPassword.Text + "','" + date + "','ACTIVE')";

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("ACCOUNT ADDED TO SYSTEM SUCCESFULLY");

this.Hide();

txtFname.Text = "";

txtLname.Text = "";

txtMname.Text = "";

txtAddress.Text = "";

txtPassword.Text = "";

}

}

}

**dashboard**

public void nav(Form form, Panel panel)

{

form.TopLevel = false;

form.Size = panel.Size;

panel.Controls.Clear();

panel.Controls.Add(form);

form.Show();

}

private void bunifuFlatButton1\_Click(object sender, EventArgs e)

{

DASHBOARD.dashboardPanel dashboardPanel = new DASHBOARD.dashboardPanel();

nav(dashboardPanel, main);

}

private void bunifuFlatButton2\_Click(object sender, EventArgs e)

{

ACCOUNTS.accountsPanel accountsPanel = new ACCOUNTS.accountsPanel();

nav(accountsPanel, main);

}

private void bunifuFlatButton3\_Click(object sender, EventArgs e)

{

CUSTOMER.customerPanel customerPanel = new CUSTOMER.customerPanel();

nav(customerPanel, main);

}

private void bunifuFlatButton6\_Click(object sender, EventArgs e)

{

STOCKS.stocksMonitoringPanel stocksMonitoringPanel = new STOCKS.stocksMonitoringPanel();

nav(stocksMonitoringPanel, main);

}

private void bunifuFlatButton4\_Click(object sender, EventArgs e)

{

ORDERS.ordersPanel ordersPanel = new ORDERS.ordersPanel();

nav(ordersPanel, main);

}

**CUSTOMER  
addCustomerPanel**

private void addCustomerPanel\_Load(object sender, EventArgs e)

{

String date = DateTime.Now.ToString("yyyyMM");

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd1 = new SqlCommand();

cmd1.Connection = con;

con.Open();

cmd1.CommandText = "SELECT COUNT(\*) FROM customerInfo;";

int num1 = Convert.ToInt32(cmd1.ExecuteScalar());

int res = num1 + 1;

txtacc.Text = date + "-" + "00" + res;

}

private void bunifuFlatButton2\_Click(object sender, EventArgs e)

{

if (txtFname.Text.Equals("") || txtMname.Text.Equals("") || txtLname.Text.Equals("") || txtAddress.Text.Equals(""))

{

MessageBox.Show("Please Fill In all the fields");

}

else

{

DialogResult result = MessageBox.Show("Are you sure you want to save the data?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

String date = DateTime.Now.ToString("yyyy-MM-dd");

//String date1 = DateTime.Now.ToString("yyyy");

Random rand = new Random();

int number = rand.Next(0, 100);

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "Insert into customerInfo VALUES('" + txtacc.Text + "','" + txtFname.Text + "','" + txtMname.Text + "','" + txtLname.Text + "','" + txtAddress.Text + "','" + date + "','ACTIVE')";

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("CUSTOMER ADDED TO SYSTEM SUCCESFULLY");

this.Hide();

txtFname.Text = "";

txtLname.Text = "";

txtMname.Text = "";

txtAddress.Text = "";

}

}

}

**customerPanel**

public void CustomForm()

{

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd = new SqlCommand();

con.Open();

SqlDataAdapter sqlDATA = new SqlDataAdapter("SELECT customerInfo.customerID,CONCAT(customerInfo.customerFname, ' ' , customerInfo.customerMname , ' ' , customerInfo.customerLname) AS NAME, customerInfo.customerAddress,customerInfo.customerEntry,customerInfo.customerStatus FROM customerInfo WHERE customerInfo.customerStatus = 'ACTIVE';", con);

DataTable dtbl = new DataTable();

sqlDATA.Fill(dtbl);

dataTable.AutoGenerateColumns = false;

dataTable.DataSource = dtbl;

}

private void customerPanel\_Load(object sender, EventArgs e)

{

CustomForm();

}

}

}

**ORDERS**

**orderFormPanel**

public void TableForm()

{

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd = new SqlCommand();

con.Open();

SqlDataAdapter sqlDATA = new SqlDataAdapter("SELECT \* FROM stocksInfo WHERE stockStatus = 'AVAILABLE';", con);

DataTable dtbl = new DataTable();

sqlDATA.Fill(dtbl);

dataGridView1.AutoGenerateColumns = false;

dataGridView1.DataSource = dtbl;

}

private int CalculateColumnSum()

{

int sum = 0;

// Iterate over the rows of the DataGridView

foreach (DataGridViewRow row in dataTable.Rows)

{

// Check if the row is not a new row or a header row

if (!row.IsNewRow && row.Index != -1)

{

// Retrieve the value from the desired column (assuming it's the first column, change the index if needed)

int cellValue;

if (int.TryParse(row.Cells[4].Value?.ToString(), out cellValue))

{

// Add the value to the sum

sum += cellValue;

}

}

}

return sum;

}

public void CustomForm()

{

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd = new SqlCommand();

con.Open();

SqlDataAdapter sqlDATA = new SqlDataAdapter("SELECT orderStatusInfo.referenceID,orderStatusInfo.customerID,CONCAT(customerInfo.customerFname, ' ' , customerInfo.customerMname , ' ' , customerInfo.customerLname) AS NAME,orderStatusInfo.livestockID, orderStatusInfo.serviceFeeAdded,orderStatusInfo.dateOrder,orderStatusInfo.orderStatus FROM orderStatusInfo INNER JOIN customerInfo ON customerInfo.customerID = orderStatusInfo.customerID WHERE orderStatusInfo.customerID = '"+ cmb1.Text + "' AND orderStatus = 'PENDING PAYMENT' ", con);

DataTable dtbl = new DataTable();

sqlDATA.Fill(dtbl);

dataTable.AutoGenerateColumns = false;

dataTable.DataSource = dtbl;

}

private void orderFormPanel\_Load(object sender, EventArgs e)

{

TableForm();

bunifuCards1.Visible = false;

String date = DateTime.Now.ToString("yyyyMMdd");

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd1 = new SqlCommand();

cmd1.Connection = con;

con.Open();

cmd1.CommandText = "SELECT COUNT(\*) FROM orderSummaryInfo;";

int num1 = Convert.ToInt32(cmd1.ExecuteScalar());

int res = num1 + 1;

txtacc.Text = "REF-"+ date + "-" +"00"+ + res;

using (SqlConnection con1 = new SqlConnection())

{

con1.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

using (SqlDataAdapter sda = new SqlDataAdapter("SELECT customerID,CONCAT(customerFname,' ',customerMname,' ',customerLname) AS NAME FROM customerInfo", con1))

{

//Fill the DataTable with records from Table.

DataTable dt = new DataTable();

sda.Fill(dt);

//Insert the Default Item to DataTable.

DataRow row = dt.NewRow();

row[0] = 0;

row[1] = "Please select";

dt.Rows.InsertAt(row, 0);

//Assign DataTable as DataSource.

cmb1.DataSource = dt;

cmb1.DisplayMember = "customerID";

}

}

}

private void txtFname\_OnValueChanged(object sender, EventArgs e)

{

}

private void cmb1\_SelectedIndexChanged(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "SELECT customerID,customerFname,customerMname,customerLname,customerAddress FROM customerInfo where customerID = '" + cmb1.Text + "';";

SqlDataReader sdr = cmd.ExecuteReader();

while (sdr.Read())

{

txtFname.Text = Convert.ToString(sdr.GetValue(1).ToString());

CustomForm();

txtTotalPurhcase.Text = "0";

for (int i = 0; i < dataTable.Rows.Count; i++)

{

txtTotalPurhcase.Text = Convert.ToString(double.Parse(txtTotalPurhcase.Text) + double.Parse(dataTable.Rows[i].Cells[4].Value.ToString()));

}

}

con.Close();

}

private void bunifuFlatButton1\_Click(object sender, EventArgs e)

{

this.Hide();

}

private void cmb1\_SelectionChangeCommitted(object sender, EventArgs e)

{

}

private void cmb2\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void bunifuImageButton1\_Click(object sender, EventArgs e)

{

double num1 = Convert.ToDouble(txtTotal.Text);

double num2 = Convert.ToDouble(txtService.Text);

double totalFinal = num1 + num2;

txtTotalPay.Text = totalFinal.ToString();

}

private void bunifuFlatButton3\_Click(object sender, EventArgs e)

{

bunifuCards1.Visible = true;

}

private void bunifuImageButton2\_Click(object sender, EventArgs e)

{

bunifuCards1.Visible = false;

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if (e.ColumnIndex == 5)

{

bunifuCards1.Visible = true;

int v = e.RowIndex;

DataGridViewRow row = dataGridView1.Rows[v];

txtInventory.Text = row.Cells[0].Value.ToString();

txtWeight.Text = row.Cells[1].Value.ToString();

txtKilo.Text = row.Cells[3].Value.ToString();

double num1 = Convert.ToDouble(txtWeight.Text);

double num2 = Convert.ToDouble(txtKilo.Text);

double totalFinal = num1 \* num2;

txtTotal.Text = totalFinal.ToString();

bunifuCards1.Visible = false;

}

}

private void bunifuFlatButton5\_Click(object sender, EventArgs e)

{

if (txtFname.Text.Equals("") || txtInventory.Text.Equals("") || txtWeight.Text.Equals("") || txtKilo.Text.Equals(""))

{

MessageBox.Show("Please Fill In all the fields");

}

else

{

DialogResult result = MessageBox.Show("Proceed to order?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

String date = DateTime.Now.ToString("yyyy-MM-dd");

//String date1 = DateTime.Now.ToString("yyyy");

Random rand = new Random();

int number = rand.Next(0, 100);

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd = new SqlCommand();

SqlCommand cmd1 = new SqlCommand();

cmd.Connection = con;

cmd1.Connection = con;

con.Open();

cmd.CommandText = "Insert into orderStatusInfo VALUES('" + txtacc.Text + "','" + cmb1.Text + "','" + txtInventory.Text + "','" + txtTotalPay.Text + "','" + date + "','PENDING PAYMENT')";

cmd1.CommandText = "Update stocksInfo set stockStatus = 'RELEASE' WHERE livestockID = '" + txtInventory.Text + "';";

cmd.ExecuteNonQuery();

cmd1.ExecuteNonQuery();

con.Close();

MessageBox.Show("ADDED to ORDER LIST");

TableForm();

CustomForm();

txtTotalPurhcase.Text = "0";

for (int i = 0; i < dataTable.Rows.Count; i++)

{

txtTotalPurhcase.Text = Convert.ToString(double.Parse(txtTotalPurhcase.Text) + double.Parse(dataTable.Rows[i].Cells[4].Value.ToString()));

}

txtFname.Text = "";

txtInventory.Text = "";

txtWeight.Text = "";

txtKilo.Text = "";

}

}

}

private void txtService\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

double num1 = Convert.ToDouble(txtTotal.Text);

double num2 = Convert.ToDouble(txtService.Text);

double totalFinal = num1 + num2;

txtTotalPay.Text = totalFinal.ToString();

}

}

private void bunifuFlatButton2\_Click(object sender, EventArgs e)

{

DialogResult result = MessageBox.Show("Proceed wiht this purhcase?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

String date = DateTime.Now.ToString("yyyy-MM-dd");

Random rand = new Random();

int number = rand.Next(0, 100);

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd = new SqlCommand();

SqlCommand cmd1 = new SqlCommand();

cmd.Connection = con;

cmd1.Connection = con;

con.Open();

cmd.CommandText = "Insert into orderSummaryInfo VALUES('" + txtacc.Text + "','" + cmb1.Text + "','" + txtTotalPurhcase.Text + "','" + date + "','PENDING PAYMENT')";

cmd1.CommandText = "Update orderStatusInfo set orderStatus = 'WAITING' WHERE customerID = '" + cmb1.Text + "' AND referenceID = '"+txtacc.Text+"'";

cmd.ExecuteNonQuery();

cmd1.ExecuteNonQuery();

con.Close();

MessageBox.Show("ADDED to ORDER LIST");

CustomForm();

txtFname.Text = "";

txtInventory.Text = "";

txtWeight.Text = "";

txtKilo.Text = "";

}

}

private void dataTable\_CellStateChanged(object sender, DataGridViewCellStateChangedEventArgs e)

{

txtTotalPurhcase.Text = (from DataGridViewRow row in dataTable.Rows where row.Cells[4].FormattedValue.ToString() != string.Empty select Convert.ToInt32(row.Cells[4].FormattedValue)).Sum().ToString();

}

private void dataTable\_CellValueChanged(object sender, DataGridViewCellEventArgs e)

{

txtTotalPurhcase.Text = (from DataGridViewRow row in dataTable.Rows where row.Cells[4].FormattedValue.ToString() != string.Empty select Convert.ToInt32(row.Cells[4].FormattedValue)).Sum().ToString();

}

}

}

**ordersPanel**

public void CustomForm()

{

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd = new SqlCommand();

con.Open();

SqlDataAdapter sqlDATA = new SqlDataAdapter("SELECT orderSummaryInfo.referenceID,orderSummaryInfo.customerID,CONCAT(customerInfo.customerFname, ' ' , customerInfo.customerMname , ' ' , customerInfo.customerLname) AS NAME, orderSummaryInfo.totalPayment,orderSummaryInfo.purchaseDate,orderSummaryInfo.orderStatus FROM orderSummaryInfo INNER JOIN customerInfo ON customerInfo.customerID = orderSummaryInfo.customerID", con);

DataTable dtbl = new DataTable();

sqlDATA.Fill(dtbl);

dataTable.AutoGenerateColumns = false;

dataTable.DataSource = dtbl;

}

private void ordersPanel\_Load(object sender, EventArgs e)

{

CustomForm();

}

}

}

**STOCKS**

**addStockPanel**

private void bunifuFlatButton2\_Click(object sender, EventArgs e)

{

if (txtType.Text.Equals("") || txtMname.Text.Equals("") || txtkilos.Text.Equals(""))

{

MessageBox.Show("Please Fill In all the fields");

}

else

{

DialogResult result = MessageBox.Show("Are you sure you want to save the data?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

String date = DateTime.Now.ToString("yyyy-MM-dd");

//String date1 = DateTime.Now.ToString("yyyy");

Random rand = new Random();

int number = rand.Next(0, 100);

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "Insert into stocksInfo VALUES('" + txtGen.selectedValue + "','" + txtMname.Text + "','" + txtType.Text + "','" + txtkilos.Text + "','" + date + "','AVAILABLE')";

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("STOCKS ADDED TO SYSTEM SUCCESFULLY");

this.Hide();

txtType.Text = "";

txtMname.Text = "";

txtkilos.Text = "";

}

}

}

}

}

**stockMonitoringPanel**

public void CustomForm()

{

String date = DateTime.Now.ToString("yyyy-MM-dd");

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd = new SqlCommand();

con.Open();

SqlDataAdapter sqlDATA = new SqlDataAdapter("SELECT \* FROM stocksInfo WHERE stockStatus = 'AVAILABLE' AND dateEntry = '"+date+"'", con);

DataTable dtbl = new DataTable();

sqlDATA.Fill(dtbl);

dataTable.AutoGenerateColumns = false;

dataTable.DataSource = dtbl;

}

public void Logs()

{

String date = DateTime.Now.ToString("yyyy-MM-dd");

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd = new SqlCommand();

con.Open();

SqlDataAdapter sqlDATA = new SqlDataAdapter("SELECT \* FROM stocksInfo", con);

DataTable dtbl = new DataTable();

sqlDATA.Fill(dtbl);

dataTable.AutoGenerateColumns = false;

dataTable.DataSource = dtbl;

}

public void DailyOut()

{

String date = DateTime.Now.ToString("yyyy-MM-dd");

SqlConnection con = new SqlConnection();

con.ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();

SqlCommand cmd = new SqlCommand();

con.Open();

SqlDataAdapter sqlDATA = new SqlDataAdapter("SELECT \* FROM stocksInfo WHERE stockStatus = 'RELEASE'", con);

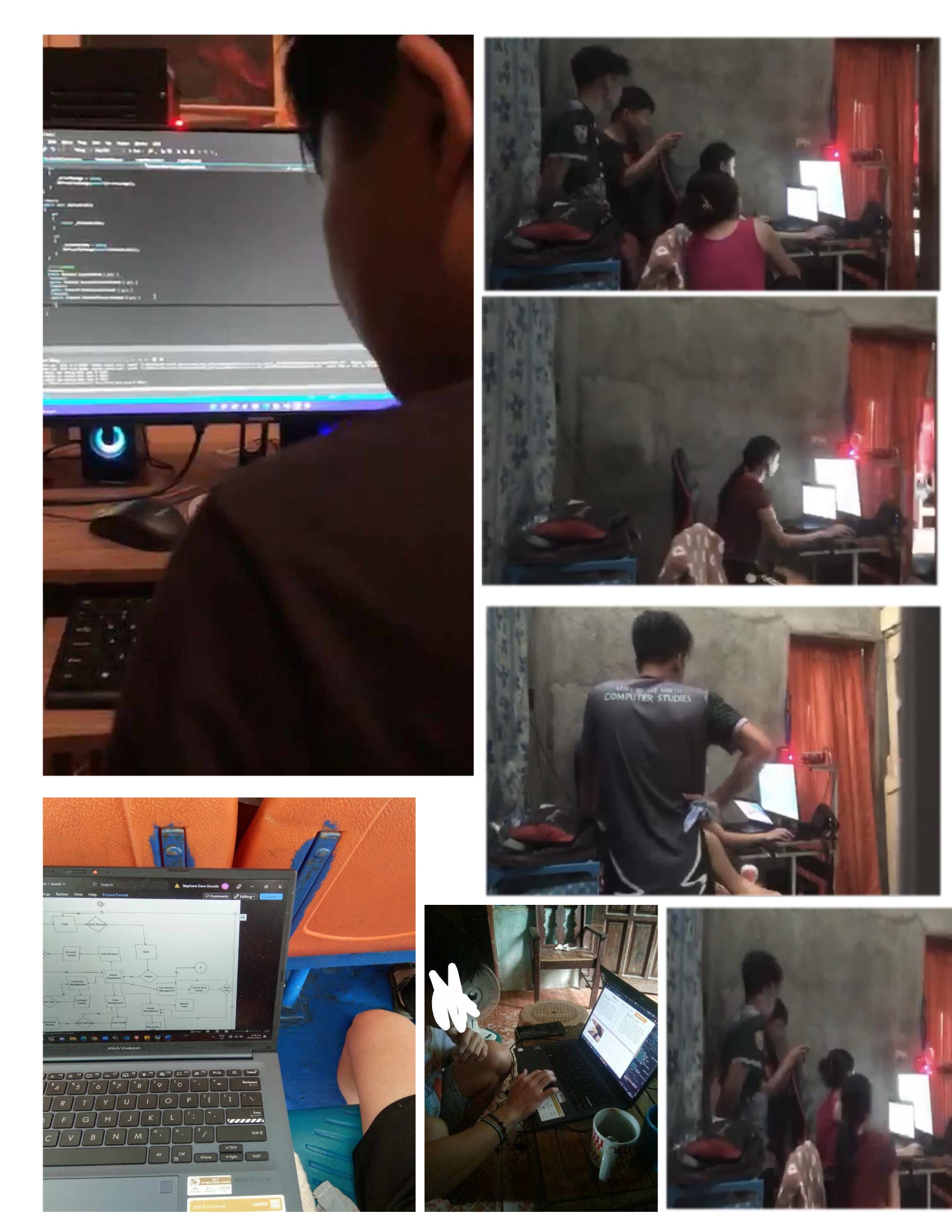
DataTable dtbl = new DataTable();

sqlDATA.Fill(dtbl);

dataTable.AutoGenerateColumns = false;

dataTable.DataSource = dtbl;

}

**APPENDIX C – DOCUMENTARY**

**REFERENCES**

*A Complete Guide to Sales Monitoring Systems (With Tips)*. (n.d.). Indeed Career Guide. Retrieved September 22, 2022, from <https://www.indeed.com/career-advice/career-development/sales-monitoring-system>

*A Complete Guide to Sales Monitoring Systems (With Tips)*. (n.d.). Indeed Career Guide. Retrieved September 22, 2022, from <https://www.indeed.com/career-advice/career-development/sales-monitoring-system>

*Attendance Monitoring in the Philippines: Significant Improvement of Punctuality and Performance of Employees*. (2018, June). ELID. <https://elid.com.ph/attendance-monitoring-system-philippines/>

Brown, R. (2021, September 17). *What is Sales Monitoring?* CloudApper - Automate Manual Business Operations Without the Technical Burden. Retrieved September 22, 2022, from <https://www.cloudapper.com/sales-force-management/what-sales-monitoring/>

Bianca, A. (2016, October 26). Sales & Inventory Monitoring Systems. Small Business - Chron.com. Retrieved September 22, 2022, from <https://smallbusiness.chron.com/sales-inventory-monitoring-systems-18660.html>

*Chapter 2 Review of Related Literature In this chapter - Free Scholarship Essay Examples*. (2019, August 13). Transect Science. Retrieved September 22, 2022, from <https://transectscience.org/chapter-2-review-of-related-literature-in-this-chapter/>

Kahn, S. (2021, November 8). *Monitoring Sales-Benefits of Sales Monitoring System to Improve Sales Performance-NogenTech*. Story Telling Co. Retrieved September 22, 2022, from <https://storytellingco.com/monitoring-sales-benefits-of-sales-monitoring-system-to-improve-sales-performance-nogentech/>

Phillips, C. (2016, October 26). *What Types of Inventory Should Be Kept in a Warehouse?* Small Business - Chron.com. Retrieved September 22, 2022, from <https://smallbusiness.chron.com/types-inventory-should-kept-warehouse-14426.html>

**CURRICULUM VITAE**



