# **CHAPTER I**

**INTRODUCTION**

## **1.1. Rationale**

Water is a fundamental resource essential for human survival, health, and development. Recognized as a basic human right by the United Nations, access to clean and safe drinking water is critical not only for sustaining life but also for ensuring overall well-being **[1]**. However, close to 2 billion people worldwide still lack access to safe drinking water services, leaving them vulnerable to various health risks associated with contaminated water sources **[2]**. Beyond its essential role in hydration, clean water is crucial for cooking, sanitation, and maintaining overall hygiene, all of which are integral aspects of daily life and public health **[3].**

Accessing clean water has grown more and more challenging in many places of the world. Many communities struggle to obtain clean water due to the increasing strain on accessible water sources caused by the growth of cities and population. As a result, water refilling stations are now an essential component of the solution, filling the gaps left by traditional water systems in meeting the growing demand for safe drinking water. Particularly in places where access to clean water isn't always assured, these stations provide a dependable and easily accessible choice. Water refilling stations reduce plastic waste and encourage environmental responsibility by offering a sustainable and safe substitute for bottled water through the use of cutting-edge filtering technology **[4]**.

Water refilling stations provide an economical and sustainable alternative to bottled water, allowing customers to refill their containers with purified water. Most stations utilize advanced filtration systems, such as reverse osmosis, ultraviolet (UV) sterilization, and activated carbon filtration. The process begins with reverse osmosis, where water is pushed through a semi-permeable membrane under high pressure. This step removes dissolved solids, heavy metals, bacteria, and other contaminants. As a result, it ensures water purity and reduces hardness, making it suitable for consumption. Next, the water undergoes ultraviolet (UV) sterilization, where it is exposed to UV light. This step effectively eliminates harmful microorganisms by disrupting their DNA, ensuring up to 99.99% pathogen elimination. UV sterilization is a chemical-free process that preserves the water’s natural properties. Finally, the water is processed through activated carbon filtration, which adsorbs impurities such as chlorine, volatile organic compounds (VOCs), and certain pesticides. This step enhances the taste and odor of the water while retaining beneficial minerals **[5]**. Together, these technologies ensure that the water dispensed meets stringent health and safety standards. Additionally, by offering a refillable, eco-friendly solution, these stations help combat the rising concerns about plastic waste and the environmental footprint of bottled water. The systematic purification process not only guarantees safe drinking water but also aligns with sustainable practices by reducing reliance on single-use plastics **[6]**.

Water refilling stations in Tagbilaran City, Bohol, have become a cornerstone of daily life, ensuring the community's access to safe and reliable drinking water. According to the Business Permit and Licensing Office (BPLO) of the local government, the data reveals a notable rise in the number of registered water refilling stations, growing from 28 in 2018 to 44 in 2023. This expansion mirrors the increasing reliance on these services and a corresponding surge in revenue, climbing from ₱7,239,998.00 in 2018 to an impressive ₱19,283,277.00 in 2023. Station 1 achieves an average of 2,100 containers sold per week, reflecting its ability to cater to a broader customer base. Station 2 records an average of 1,500 containers sold per week, showcasing steady demand. Station 3 reports an average of 1,700 containers sold per week. This rapid growth highlights the essential role these stations play in meeting the community's demand for clean water, driven by heightened public awareness about water safety and convenience. The upward trajectory also highlights the untapped opportunities for business innovation and expansion as the community's need for dependable water sources continues to grow.

Water refilling stations in Tagbilaran City typically offer a variety of water types to meet the diverse needs and preferences of consumers. The most common are purified water, alkaline water, and mineral water. Purified water goes through extensive filtration and reverse osmosis processes to remove impurities and contaminants, making it the most affordable option, usually priced at around ₱25 per 5-gallon container. Alkaline water, on the other hand, has a higher pH level and is believed to help neutralize acidity in the body, promoting better hydration and potential health benefits. Mineral water retains natural minerals like calcium, magnesium, and potassium, which are essential to overall health. Both alkaline and mineral water are often considered premium choices and are typically sold at ₱50 per 5-gallon container. This variety of options allows consumers to choose water products that best suit their lifestyle, health goals, and budget, further increasing the appeal and demand for refilling services.

The operations of water refilling stations in Tagbilaran City rely largely on manual processes to ensure a steady and reliable supply of clean water. Onsite workers handle the refilling of empty water containers brought in by customers or delivery personnel. Each refill is recorded by hand, tracking the number of containers filled and the volume of water dispensed. This manual inventory tracking is crucial for maintaining an up-to-date record of stock, enabling stations to anticipate and prepare for customer demand, whether for walk-in refills or deliveries.

Onsite workers also log the sizes of the containers filled, as this data helps calculate total sales and monitor remaining stock levels. By managing these details manually, stations can avoid stock shortages and ensure smooth operations. Once refilled, the containers are organized for either immediate pick-up by walk-in customers or delivery. For delivery orders, onsite workers prepare the appropriate number of filled containers to load onto delivery vehicles, streamlining the distribution process and minimizing delays. While this manual approach can be time-consuming, it remains a cornerstone of day-to-day operations, offering precise control over inventory and a personalized level of service.

Delivery workers play a key role in maintaining efficiency. They log the number of containers loaded into their vehicles and track each delivery as it happens. Delivery workers proactively visit neighborhoods to attract potential customers and focus on fulfilling specific delivery orders communicated by the station staff. Orders are typically placed and confirmed through text messages or emails, ensuring smooth coordination between customers and the stations.

Despite the essential role of water refilling stations in Tagbilaran City, Bohol, several operational challenges hinder their efficiency and customer satisfaction. Insights from interviews with local station owners and operators revealed key issues that impact daily operations.

One significant challenge lies in the inefficiencies faced by delivery workers, who often spend hours roaming neighborhoods in search of potential customers. This method not only wastes time but also creates uncertainty about sales, as drivers have no way to predict how much water they’ll sell on a given trip. Without a targeted customer base, drivers frequently cover long distances without guaranteed returns, making it difficult to sustain profitability. For customers, this results in inconsistent delivery schedules and, at times, missed orders, leaving them without access to clean water when they need it most.

Another common issue is the difficulty in locating exact delivery addresses. Drivers often rely on vague or incomplete information, which can lead to prolonged searches, wasted fuel, and delays in reaching customers. For example, a driver might struggle to find a specific home, delaying deliveries that could have otherwise been completed more efficiently. From the customer’s perspective, this adds to their frustration, particularly when deliveries arrive late or require them to wait longer than expected. These inefficiencies in managing delivery routes not only affect the stations' operations but also undermine customer trust and satisfaction.

Additionally, the reliance on manual processes for logging sales and inventory poses significant challenges. Onsite workers record refilled containers and sales for different types of water such as purified, alkaline, and mineral water, etc. This system is prone to human error, with records easily misplaced or inaccurately written. For customers, this can result in mistakes such as receiving incorrect orders. For station operators, manually tracking inventory for multiple water types becomes increasingly difficult, making it harder to identify which products are running low or are overstocked. Without an automated system, stations may face issues like overstocking certain types of water while running out of others, leading to additional costs or delays in meeting customer demand. These inefficiencies collectively affect the overall reliability and quality of service.

A platform is needed to address these operational challenges. The manual processes currently in place are time-consuming, prone to errors, and hinder efficiency in inventory management, sales tracking, and delivery operations. Without a streamlined system, stations struggle to meet demand, resulting in stock discrepancies and delayed deliveries. Delivery workers waste time searching for customers due to inefficient routing and incomplete information. Additionally, many residents find it difficult to access the services of water refilling stations as they often do not know the contact numbers of the stations or how to place orders. Determining delivery locations is further complicated by the lack of clear landmarks or detailed addresses, leading to delayed or missed deliveries and inconvenience for both customers and delivery workers. From the customer’s perspective, these inefficiencies result in inconsistent delivery schedules and delayed orders, which undermines trust and satisfaction. A platform that automates inventory tracking optimizes delivery routes and consolidates communication between stations, drivers, and customers can improve efficiency, reduce errors, and enhance service reliability. For customers, it would provide a more seamless experience by enabling order tracking, easy access to station’s contact information, timely deliveries, and convenient payment options. Additionally, it would provide valuable data insights, helping station owners make informed decisions about inventory, staffing, and pricing, ultimately improving profitability and customer satisfaction.

Despite the clear advantages of automation, many water refilling stations continue to rely on manual processes due to several key barriers. A major factor is the limited access to technology, particularly in smaller or locally owned stations that may not have the resources to invest in automated systems. High upfront costs of software, hardware, and training for staff can be prohibitive, especially for smaller businesses with tight budgets. Additionally, many station owners perceive the initial investment in digital tools as unnecessary, given that the existing manual methods, though time-consuming, are familiar and still function to some extent.

Another barrier is the resistance to change, which is common in industries that have relied on traditional methods for many years. Water refilling station operators may be hesitant to adapt new technologies due to concerns about disrupting their current operations or fear of training staff on unfamiliar systems. Furthermore, some station owners may doubt the long-term benefits of digital solutions, preferring to stick with manual processes that have worked for them in the past.

Given these challenges, there is a clear opportunity to develop a mobile platform like Hydrohub, which is designed to address inefficiencies in the current operations of water refilling stations. The development approach would focus on streamlining sales, orders, delivery, and stock management processes, transitioning from manual methods to automated solutions. Hydrohub's envisioned functionality includes tracking sales, inventory levels, customer orders, and delivery routes. By incorporating GPS tracking integrated with a two-dimensional scale map of Tagbilaran City, the platform aims to optimize delivery operations and resource allocation. These proposed features are expected to reduce errors associated with manual data entry and enable stations to better meet customer demands. The conceptualization and design of Hydrohub would prioritize real-time updates and a user-friendly interface to enhance operational efficiency and customer satisfaction.

The development of Hydrohub presents an opportunity to enhance decision-making for station owners by incorporating detailed analytics and insights derived from tracked data. The envisioned platform would generate reports on sales trends, inventory usage, and delivery performance, enabling owners to identify areas for improvement, optimize inventory levels, and plan staffing requirements. By analyzing peak demand periods and popular product types, such as purified or alkaline water, Hydrohub aims to help station owners make informed, data-driven decisions to improve profitability and customer satisfaction. To ensure accessibility, the design of Hydrohub prioritizes a simple and intuitive interface catering to users with varying levels of technical expertise. The platform would provide step-by-step guidance for onsite workers, delivery personnel, customers, and station owners, streamlining the use of its features.

In addition to operational improvements, Hydrohub's development envisions the incorporation of innovative digital promotional strategies to replace traditional marketing methods. By leveraging the platform’s reach and incorporating customer feedback tools, the concept aims to provide station owners with a deeper understanding of customer preferences, enabling them to tailor their services accordingly. These features are anticipated to enhance customer experiences while opening opportunities for stations to expand their market reach and increase revenue.

## **1.2. Objectives of the Study**

The primary objective of this study is to develop Hydrohub, a mobile platform aimed at streamlining sales, delivery, and inventory management for water refilling stations in Tagbilaran City, Bohol. The specific objectives of the study are as follows:

1. To assess the operations of water refilling stations in Tagbilaran City, Bohol, including sales, inventory management, and delivery operations.
2. To determine the tools, technologies, and algorithms necessary for optimizing operations.
3. To design and integrate the platform’s core functionalities.
4. To test the capabilities and functionalities of Hydrohub.

## **1.3. Scope and Limitations of the Study**

This study focuses on understanding the operations of water refilling stations in Tagbilaran City, Bohol. It aims to identify inefficiencies in their current sales, delivery, and inventory management processes. By analyzing these aspects, the study will provide a foundation for developing Hydrohub, a mobile platform designed to address the operational challenges faced by these stations.

The scope of the study includes an evaluation of how water refilling stations operate and how these practices impact both their efficiency and the satisfaction of their customers. To gather relevant data, the research will involve interviews with station owners to learn about their management challenges, interviews with customers and employees to gather feedback on their experiences, and an analysis of sales and inventory records to identify inefficiencies. These methods aim to provide an understanding of the challenges that stations face and how these challenges can be addressed through the proposed platform.

However, the study acknowledges several limitations. The geographic scope of the research will be confined to Tagbilaran City, which may limit the applicability of the findings to other regions. This geographic focus is necessary due to the nature of the research, as the operational dynamics and challenges faced by water refilling stations in other areas might differ. As such, the findings from this study will be contextualized within Tagbilaran City, and the long-term sustainability and scalability of Hydrohub beyond its initial development phase will not be evaluated. Additionally, the study will not cover employee scheduling or attendance tracking, focusing exclusively on sales, inventory, and delivery management. The product offerings considered will be limited to the water products available at the stations, focus on water refilling stations that have delivery services and pre-paid transactions is not available as trasactions are done after the delivery and refilling of container onsite. Furthermore, while the platform will feature route optimization for delivery, it will not incorporate real-time traffic conditions or other external factors that may affect delivery routes.

To assess the functionality and real-world effectiveness of Hydrohub, beta testing will be conducted at three strategically selected water refilling stations in Tagbilaran City. The stations chosen for testing include Paljar Water Refilling Station in Ubujan District, Aquafinity Water Refilling Station, and Samwa Water Refilling Station in Tiptip District. These stations were selected based on several criteria to ensure a diverse range of operational scenarios, enabling the platform to be tested in various contexts. The first criterion is the geographic location of the stations. By selecting stations in different districts, the study ensures that the platform is tested across varying geographic areas, each with unique delivery requirements and customer behaviors. Secondly, the size of the operations at each station was considered. The stations represent different operational scales, ranging from smaller, more localized businesses to larger operations with a wider customer base, allowing the study to assess how well Hydrohub adapts to different business capacities.

Another important criterion is the customer base served by each station. These stations cater to different types of customers, including residential, commercial, and institutional clients. This diversity ensures that the platform’s effectiveness is tested across a broad spectrum of customer needs and preferences. Finally, the technological readiness of the stations played a role in the selection process. By including stations with varying levels of technological infrastructure, the study will be able to evaluate how Hydrohub performs under different technological conditions, ensuring its adaptability to stations with different resources and capabilities.

The study will adhere to strict ethical considerations to ensure the integrity and fairness of the research process. Participants, including station owners, employees, and customers, will be fully informed about the purpose and scope of the study and their voluntary participation will be sought through signed consent forms. Privacy and confidentiality will be prioritized, with all personal and operational data collected being anonymized and securely stored. The research will also ensure equity and fairness by including participants from diverse operational setups and customer demographics, ensuring that the findings reflect a broad range of experiences.

# **CHAPTER II**

## **REVIEW OF RELATED LITERATURE AND STUDIES**

The researchers reviewed similar studies to understand the best practices and innovations used in such integrated solutions. This review of related studies is beneficial for the proponents in planning and designing Hydrohub.

A study by Ary Arvianto et al. focused on addressing logistical inefficiencies in the distribution of drinking water containers. The primary aim of their research was to optimize delivery routes to minimize transportation costs and delivery times while maximizing resource utilization **[7]**. This approach sought to enhance customer service by streamlining the distribution process and improving overall operational efficiency. To achieve their objectives, the study employed a structured methodology. First, the researchers collected essential data, including customer demand, delivery locations, vehicle capacities, and route constraints. This information was analyzed to identify inefficiencies in the existing delivery system. Next, they utilized two optimization algorithms, namely the Saving Matrix Algorithm and the Nearest-Neighbor Algorithm, to develop efficient routing strategies. These algorithms prioritized minimizing travel distances and combining routes to save time and costs. The system was then integrated into a Geographic Information System (GIS) framework, which provided spatial visualization of the optimized delivery routes. The GIS-based system allowed the researchers to conduct simulations and test their solutions in scenarios that mimicked real-world conditions. By comparing the results with traditional distribution practices, the study demonstrated significant improvements in delivery efficiency, cost reduction, and resource utilization. The findings of the study revealed that the use of Geographic Information Systems (GIS) and routing algorithms effectively enhanced service delivery and reduced operational inefficiencies. This integration of advanced technologies not only minimized transportation costs but also improved customer satisfaction by ensuring timely deliveries and efficient resource allocation. For Hydrohub, the study's approach offers valuable insights, particularly in integrating advanced routing technologies with Global Positioning System (GPS) capabilities. Unlike Geographic Information System (GIS), which focuses on spatial visualization, Global Positioning System (GPS) technology is better suited for real-time tracking and dynamic route adjustments, which are critical for Hydrohub's operations. By leveraging the Global Positioning System (GPS), Hydrohub can enable real-time delivery tracking, optimize routes dynamically, and provide live updates to both station owners and customers. This integration will ensure reduced delivery times, lower fuel consumption, and improved service reliability, making it highly beneficial for water refilling stations in Tagbilaran City.

“A Mobile Delivery Application Applying Fixed Radius Near Neighbors and Help-Karp Algorithms" by Reynaldo Castillo et al. introduced a mobile application aimed at improving delivery efficiency for small businesses. The researchers employed the Held-Karp algorithm to solve the Traveling Salesman Problem (TSP), determining the most efficient delivery routes for vehicles. Additionally, the Fixed-Radius Near Neighbors algorithm was utilized to identify nearby businesses, enabling localized delivery optimization and enhanced service for nearby customers **[8]**. The application’s functionality included detailed navigation instructions, which helped reduce delivery times and operational costs. To create the system, the researchers integrated the Held-Karp algorithm for route optimization by analyzing all possible delivery sequences and selecting the shortest path. This approach ensured maximum route efficiency, especially for scenarios involving multiple stops. Meanwhile, the Fixed-Radius Near Neighbors algorithm analyzed the geographic distribution of businesses and customers, focusing on those within a predefined radius for better route clustering. The application’s user interface was designed to provide real-time navigation, order updates, and feedback mechanisms, ensuring ease of use for both businesses and their customers. The study concluded that the application successfully enhanced delivery processes for small businesses, significantly reducing delivery times and increasing overall efficiency. User feedback highlighted the system’s reliability and user-friendliness, and the application met ISO 25010 quality standards, reflecting its effectiveness and user acceptance. Hydrohub can adopt these algorithms to improve its operational capabilities. By integrating the Held-Karp algorithm, Hydrohub can optimize delivery routes for water refilling stations, minimizing travel distances and reducing fuel costs. The Fixed-Radius Near Neighbors algorithm can be used to identify nearby customers, enabling stations to prioritize and efficiently cluster deliveries within a specific area. This approach will not only improve delivery speed but also enhance customer satisfaction by ensuring timely and reliable service. Together, these innovations will help Hydrohub streamline operations and maintain a high standard of service quality.

A study by Marco Antonio Narahaba and Francka Sakti Lee titled” Advances in water purification and filtration technologies. Journal of Water Quality” focused on addressing the operational inefficiencies faced by an industrial water depot in Jakarta, Indonesia. The depot struggled with manual, paper-based processes, which resulted in data loss and errors in inventory and sales management **[9]**. To solve these issues, the researchers developed a website-based inventory application using the Extreme Programming (XP) methodology. This study emphasizes flexibility and iterative development, which allowed the application to evolve with changing user needs. The development process followed the key phases of Extreme Programming (XP): planning, design, coding, and black-box testing. The system was designed to streamline inventory management by tracking incoming and outgoing transactions, managing customer records, and generating detailed transaction reports. It aimed to improve operational efficiency and data accuracy, ultimately enhancing the depot’s overall performance. The results of the study demonstrated the effectiveness of the Extreme Programming (XP) methodology in creating a flexible, adaptable system that met the evolving needs of the depot. The application successfully addressed the challenges of manual processes, improving both data management and operational efficiency. For Hydrohub, the Extreme Programming (XP) methodology's focus on flexibility and adaptability can be integrated to ensure that the platform can evolve to meet the changing needs of water refilling stations. By using Extreme Programming (XP), Hydrohub can quickly incorporate feedback, allowing for continuous improvement throughout the development process and ensuring that the platform remains responsive to the operational requirements of the stations.

Samantha Dominique Bucao et al. developed a computer-based inventory system to address the challenges of manual inventory management at Crystal-Clear Water Refilling Station. The goal of their study was to automate inventory tracking, streamline stock replenishment, and improve decision-making by providing detailed reports. The system featured real-time inventory tracking, automatic reordering when stock levels hit a predefined threshold, and a reporting feature for analyzing sales trends and stock movements **[10]**. The researchers created the system by first analyzing the inefficiencies of manual inventory tracking methods. They developed a centralized platform that provided real-time updates on inventory levels, enabling the automation of stock monitoring and replenishment. The system used an algorithm to automatically reorder stock based on a threshold level to prevent shortages. Additionally, the sales trend analysis algorithm was implemented to generate reports that highlighted sales patterns, helping station managers make informed decisions. The system was tested by being implemented at the water refilling station, where it was found to be effective in reducing errors, preventing stockouts, and improving operational efficiency. The findings concluded that automating the inventory management system significantly improved operational efficiency, reduced human errors, and increased customer satisfaction by maintaining optimal inventory levels and preventing delays in fulfilling customer orders. The study emphasized the importance of real-time tracking and automated reordering in ensuring better inventory management. For Hydrohub, this study is highly relevant, particularly with the integration of the Sales Trend Analysis Algorithm and the Threshold-based Reordering Algorithm. Hydrohub can adopt similar features to automate stock monitoring and replenishment, use sales trend analysis to optimize decision-making and ensure a more efficient and reliable inventory management system for water refilling stations.

The study "Information Systems Development and Technology Plan for NADAP Water Refilling Station" by Harold Lance A. Espinosa et al. explored the operational challenges faced by NADAP Water Refilling Station, which included manual inventory management, inefficient transaction processing, and the lack of a reliable system for tracking sales and generating reports **[11]**. The researchers aimed to develop an automated system to streamline these operations, enhance inventory tracking, and enable data-driven decision-making, ultimately improving the station's overall efficiency. The system utilized MySQL as the database management system to handle customer information, sales transactions, and inventory data. The database schema was structured to ensure efficient data retrieval and minimize redundancy. The backend system was developed using PHP, while the frontend employed HTML, CSS, and JavaScript to provide a user-friendly interface. Rigorous testing ensured the system's accuracy and user acceptance. The study demonstrated that the implemented system successfully addressed the station's operational inefficiencies. By automating inventory management and integrating sales tracking, the system helped NADAP Water Refilling Station maintain optimal stock levels, reduce errors, and make data-driven decisions. This ultimately led to improved operational efficiency, reduced delays, and enhanced customer satisfaction. For Hydrohub, this study is highly relevant. The integration of an automated system for sales and inventory management aligns with Hydrohub's goals of optimizing operations for water refilling stations. Additionally, the use of MySQL for database management and the structured system development approach provides valuable insights for ensuring data integrity and scalability in Hydrohub's mobile-based platform. By adopting similar strategies, Hydrohub can streamline operations for water refilling stations in Tagbilaran City, improving both efficiency and customer satisfaction. Since Hydrohub will be a mobile-based platform, PostgreSQL will be used instead of MySQL to ensure data management and accessibility across mobile devices. This integration will help Hydrohub deliver a seamless experience for users while improving operational processes and customer service.

The study "CloudWater: SaaS for Streamlining Water Refilling Operations" by Christian Baral focused on the development of a Software as a Service (SaaS) platform tailored for water refilling businesses. The research aimed to address inefficiencies in managing customer orders, inventory, and financial transactions. By leveraging cloud-based technology, the platform sought to provide a centralized and scalable solution to enhance operational efficiency and improve service delivery **[12]**. The study involved designing and developing the SaaS platform using a cloud-based architecture. The backend utilized Node.js for its event-driven, non-blocking nature, which ensured high performance and scalability. The front end was developed using Flutter, a cross-platform framework, to provide a seamless and responsive user interface across multiple devices. The database layer was implemented using PostgreSQL, which was chosen for its robustness in handling structured data and ensuring data integrity. Key features of the platform included real-time inventory tracking, automated customer order processing, and financial reporting capabilities. The development process followed an Agile methodology, emphasizing iterative development and regular stakeholder feedback to ensure the platform met user requirements effectively. The study highlighted the platform's success in addressing the operational challenges of water refilling businesses. The adoption of cloud-based technology-enabled businesses to access their data and manage operations remotely, improving flexibility and efficiency. The system's features, such as real-time tracking and automated processes, significantly reduced manual errors and enhanced decision-making capabilities, leading to better resource utilization and customer satisfaction. For Hydrohub, this study provides valuable insights into implementing solutions using technologies like Node.js and Flutter. These technologies align with Hydrohub's goal of creating a mobile-based platform for water refilling stations. By adopting similar strategies, Hydrohub can leverage cloud computing to enable real-time operations, streamline inventory and sales management, and improve overall efficiency for water refilling stations in Tagbilaran City.

Sajedeh Norozpour et al. conducted a study to develop a Water Refilling Station Management System (WRSMS) tailored for Havana, Cuba, where many water refilling stations still relied on manual methods for managing operations. The system aimed to streamline sales processes by incorporating digital transaction recording, automated receipt generation, and detailed financial reporting tools **[13]**. Additionally, WRSMS included features for monitoring customer purchasing trends, enabling station owners to make informed and strategic decisions in sales operations. The study involved purposive sampling for data collection and industry-standard usability metrics to evaluate the system. The researchers employed statistical methods, including t-tests, to assess user satisfaction and the system’s impact on operational efficiency. The findings revealed a significant improvement in sales data handling and a strong preference among employees for the WRSMS over manual methods. The study concluded that the system not only enhanced sales management processes but also improved user satisfaction and operational decision-making capabilities. For Hydrohub, this study underscores the importance of modernizing sales processes in water refilling stations. Hydrohub can integrate features inspired by WRSMS that focus on optimizing sales transactions and providing robust tools for tracking customer purchase histories. These capabilities, currently missing in Hydrohub, would allow station owners to access detailed sales reports, analyze purchasing trends, and make data-driven decisions. This tailored mobile-based integration would improve sales management efficiency and customer satisfaction, aligning with Hydrohub’s goal of enhancing operational performance for water refilling stations.

A study titled "Operational Optimization of a Potable Water Refilling Station*"* by Norma Muyot et al. focused on enhancing the efficiency of water refilling station operations by emphasizing resource utilization and systematic process improvements. The primary aim of the research was to identify and address operational inefficiencies, thereby boosting service delivery and customer satisfaction **[14]**. To achieve these objectives, the researchers employed a systematic methodology involving several key steps and innovative features. First, they conducted a comprehensive analysis of the existing workflows, identifying bottlenecks such as delays in order processing, inefficiencies in inventory management, and underutilized production resources. Using this analysis, they designed and implemented workflow optimization strategies, which included automation of manual processes, real-time inventory tracking, and predictive stock monitoring to prevent overstocking or stockouts. Additionally, they introduced demand forecasting tools that used historical sales data to anticipate customer needs and adjust production schedules accordingly. The study also leveraged data analytics and visualization tools to provide station managers with actionable insights into operational performance. These tools included dashboards that displayed key performance indicators (KPIs) such as delivery times, resource utilization rates, and customer satisfaction metrics in real-time. By integrating these features, the system enabled managers to make data-driven decisions and swiftly address any inefficiencies. The findings of the study demonstrated significant improvements in service delivery, with optimized workflows leading to reduced operational delays, enhanced customer satisfaction, and improved resource utilization. The research highlighted the critical role of structured operational strategies in achieving higher productivity and better service outcomes. For Hydrohub, the study's insights are invaluable, particularly in integrating features that enhance operational optimization. Hydrohub can adopt workflow management tools, real-time resource tracking systems, and predictive analytics to ensure efficient resource utilization and minimize service delays. By implementing similar advanced features, Hydrohub can empower water refilling stations to monitor and refine their processes dynamically, elevating the overall customer experience while optimizing operational efficiency.

The reviewed studies highlight various approaches to addressing operational inefficiencies in water refilling stations, focusing on improving inventory management, sales tracking, delivery logistics, and overall operational performance. These insights demonstrate the importance of integrating advanced technologies such as cloud-based solutions, mobile applications, database management systems, and predictive analytics to streamline processes and enhance customer satisfaction. For Hydrohub, the findings emphasize the potential of implementing tailored strategies such as real-time inventory tracking, automated sales management, and delivery route optimization. Leveraging PostgreSQL as the database, alongside Node.js and Flutter for platform development, ensures scalability, efficiency, and a seamless user experience in Hydrohub's mobile-based system. By integrating proven methodologies, including workflow optimization and customer-centric features, Hydrohub can significantly improve the efficiency and reliability of water refilling stations in Tagbilaran City, aligning with its goal of enhancing operational performance and customer satisfaction.

## **REFERENCES**

[1] United Nations. 2023. Access to safe drinking water: Fundamental human right. <https://www.un.org/en/ga/search/view_doc.asp?symbol=A/RES/64/292>

[2] World Health Organization. (2021). Drinking-water. https://www.who.int/news-room/fact-sheets/detail/drinking-wate

[3] World Health Organization. (2021). Drinking-water. https://www.who.int/news-room/fact-sheets/detail/drinking-water

[4] Johnson, A., & Williams, T. 2018. Water filtration systems: Effectiveness and technology applications. *Environmental Science Review*, 45(3), 212–228. http://surl.li/ehqcmf

[5] Lee, H. 2021. Advances in water purification and filtration technologies. *Journal of Water Quality*, 59(2), 134–142. http://surl.li/lykrgl

[6] O’Connor, S., & Lee, J. 2020. Sustainability of water refilling stations: Reducing plastic waste through eco-friendly practices. *Environmental Impact Journal*, 37(4), 97–105. https://tinyurl.com/bp5tv9rs

[7] ARVIANTO, et al., W. 2023. Optimizing the Drinking Water Container Distribution Using Vehicle Routing Problems with Pick-up and Delivery Approach (Case Study of Berkah RO Drinking Water Depot). <https://bit.ly/3Le0645>

[8] CASTILLO et al., E.D. 2019. EzDelivery: A Mobile Delivery Application Applying Fixed Radius Near Neighbors and Help-Karp Algorithms. <https://bit.ly/3XZfVDk>

[9] NARAHABA, M.A. AND LEE, F.S. 2024. Inventory Application Design at Industrial Water Depo Using Extreme Programming Method in Jakarta,

Indonesia. <https://bit.ly/3znW0nz>

[10] BUCAO, S.D., CARREON, A.B., LAUENGCO T.J., AND YOUNG, DR.M. 2020. Designing a Computer-Based Inventory System for the Crystal-Clear Water Refilling Station. [https://bit.ly/4eJAqdf](https://bit.ly/4eJAqdf" \t "_parent)

[11] ESPINOSA, H.L., MAAMBONG K.A., TAYLARAN L.M., BAGONGON J., FORMENTERA A.G., AND JADRAQUE G. 2022. Information Systems Development and Technology Plan for NADAP Water Refilling Station. <https://bit.ly/4cFjHWC>

[12] BARAL. 2019. CloudWater: A WEB AND MOBILE BASED SOFTWARE-AS-A-SERVICE FOR WATER REFILLING BUSINESSES. https://tinyurl.com/ywnxbxjr

[13] NOROZPOUR, S., MOMENZADEH, M., AND ABOLHASANI, A. 2021. Proposing a New System for Handling Business Data Systems With More Functionality and Usability.  <https://bit.ly/4bp7PqS>

[14] MUYOT, N. B., & SAFLOR, C. S. R. 2022. Operational Optimization of a Potable Water Refilling Station. https://tinyurl.com/j98xwyaf

# **CHAPTER III**

**METHODOLOGY**

## **3.1. Analysis**

In this chapter, we present the use case diagram and the use case narrative. The use case diagram identifies and explains the interactions between the actors and the system’s capabilities. The use case narrative describes the steps and actions that users must take to complete tasks. This chapter also discusses a list of functionalities.

The system involves five actors: the administrator, station owner, delivery worker, onsite worker, and customer. The Administrator is the owner of the app, who is a bookkeeper who wants to venture on a business app; the administrator is the one who oversees the entire platform, ensuring the proper registration of water refilling stations and their compliance with necessary documentary requirements. The Station Owner oversees the day-to-day operations of the water refilling station, including inventory, orders, and sales. Responsible for ensuring that water refilling services are provided efficiently through the app. The Onsite Worker Works at the water refilling station, managing inventory and ensuring stock is accurately recorded and orders are fulfilled correctly. The Delivery Worker is responsible for delivering water to customers as part of fulfilling the orders placed on the platform. The Customer places orders for water through the platform and interacts with the station for deliveries. In the use case diagram, there are six primary use cases: Manage account, Manage product, Manage inventory, Manage order, Manage walk-in sales, and Generate report.

In Manage account, the administrator registers water refilling stations by creating and approving accounts for owners after verifying their compliance with documentary requirements like business permits, license to operate, Sanitary permits, and Water Analysis and Compliance. Once registered, station owners can view, update, and deactivate their own accounts. Station owners can also create, view, update, and deactivate the accounts of their employees (delivery workers and onsite workers) as well as manage the profile associated with their water refilling stations. Customers can create, view, update, and deactivate their own accounts, while onsite workers and delivery worker can view and update the accounts assigned to them by the station owners. The system allows for account deletion or deactivation based on necessity, such as when a user no longer requires access or when compliance is not met. Inactive accounts may be archived to preserve data integrity and for potential future reinstatement.

In the Manage Product, station owners can create, update, and archive products offered by the water refilling station. This allows them to maintain an accurate and updated list of available products, such as type of water and prices. The customers can view the available products from water refilling stations. This functionality simplifies product management for the station and enhances the customer experience by providing clear and reliable information about the station's offerings. The administrator can view product logs from water refilling stations, which only limits to the date of creation, name of the station, product name and price to monitor the products in the platform while maintaining data privacy.

In the Manage Inventory, inventory management for water refilling stations is optimized through automated tracking and logging, drawing on efficient methodologies and system designs observed in similar studies. The onsite worker is responsible for creating, updating, and viewing the inventory list, which includes tracking the number of refilled containers. The system automatically logs the date and time when containers are refilled or discarded and adjusts the inventory details according to the logs, thus ensuring real-time updates and accurate record-keeping. If containers are discarded due to issues such as leakage or expiration, the onsite worker manually log these instances, ensuring that the inventory data is consistently maintained. Similarly, the delivery worker manages a separate set of inventory data, focusing on the containers stored in the delivery vehicle. This includes tracking unsold containers that are returned to the station. Like the onsite worker, the delivery worker’s actions are automatically recorded, with the system capturing the date and time of inventory changes and adjusts the inventory according to logs. This ensures that the movement of stock between the station and customers is carefully tracked, contributing to more accurate inventory data. The station owner can view and monitor the inventory managed by both the onsite worker and delivery worker, ensuring a centralized and transparent system. The administrator can view inventory logs from water refilling stations which only limits to the date of creation, name of the station, product name and product amount to monitor the products in the platform while maintaining data privacy. This approach draws from advanced inventory systems observed in the studies, where real-time tracking and automated updates lead to more efficient operations and fewer errors. Through this centralized management, discrepancies can be easily identified and addressed, ensuring better service delivery and operational efficiency. The system's automated tracking and logging process ultimately contributes to improved resource management, reducing stock-outs and ensuring that water refilling stations operate smoothly and efficiently.

In Manage Order, customers can view a list of water refilling stations and create, update, and track their orders for both delivery and pick-up. When placing an order, customers are required to fill out a form that includes the number of 5 gallon containers as it is the standard requirement for delivery, order type (Pick-up or Delivery), the type of water (e.g., purified, mineral, or alkaline), the type of service (either On-the-Day or Pre-Order), the schedule, and the delivery location. Customers can select “On-the-Day” for same-day delivery, which is available only before the station’s cutoff time, or “Pre-Order” for deliveries scheduled at least a day in advance. Onsite workers are responsible for managing these orders. They can view, accept, or deny them. Orders may be denied if the station cannot accommodate them due to capacity constraints or other issues. If an “On-the-Day” order remains in a pending status for 15 minutes without any action from the onsite worker, the system will automatically cancel the order using an Order Denial Algorithm, notifying the customer and suggesting alternative stations where they may place an order. Once orders are accepted, they are added to the delivery or pick-up list, ensuring clear communication between the station and its customers. Onsite workers can confirm pick-up orders by reviewing details such as the water type (e.g., purified, mineral, or alkaline), quantity, and price. When it comes to payment, the onsite worker checks if the correct amount is provided. After confirming the amount, the order is accepted and moved to the sales list. Delivery personnel can view the delivery list, which includes customer details, delivery address, type of water (e.g., purified, mineral, or alkaline), and the number of containers ordered. The system generates an optimal delivery route on the 2-dimensional map of Tagbilaran City using Global Positioning System (GPS) for real time updates on where the delivery personnel are with Traveling Salesman Problem (TSP) based and Held-Karp algorithms, ensuring the most efficient sequence of deliveries. These algorithms minimize the time spent on the road and ensures deliveries are made in a logical and efficient order. As deliveries are made photo confirmation of the delivery is made by the delivery personnel for evidence and the system is updated in real-time, allowing customers to track the delivery status. Delivery personnel also confirm the payment method: if paid in cash, they verify that the correct amount is received before completing the delivery. For online payments, a picture of the payment proof is stored for confirmation. The delivery status is updated throughout the process, ensuring the customer is kept informed at every step of the delivery. The station owner can view and monitor the orders managed by both the onsite worker and delivery worker. The administrator can view order logs from water refilling stations which only limits to the date delivered, name of the station, product name, product amount, price amount, and customer name to monitor the products in the platform while maintaining data privacy.

In the Manage Walk-in Sales, onsite workers and delivery personnel manage and track transactions from customers who did not place orders through the app. This ensures seamless record-keeping and efficient handling of direct sales. The onsite worker is responsible for creating, updating, and viewing the walk-in sales list, which records transactions categorized as "On-site" (in-store purchases), number of containers, their prices, the types of water (e.g., purified, mineral, or alkaline, and container sizes, as customers can purchase water for different bottle sizes directly at the station. For each transaction, the onsite worker confirms payment. For cash payments, they verify the accuracy of the amount before confirming the sale. For online payments (GCash or Maya), the onsite worker reviews the amount and uploads a picture of the payment proof to confirm the transaction. Once confirmed, these transactions are recorded in the database. The delivery personnel are responsible for creating, updating, and viewing the walk-in sales list, which records transactions categorized as "Off-site" (deliveries that are not purchased in the app), number of containers, their prices, and the types of water (e.g., purified, mineral, or alkaline). They follow the same payment confirmation process as the onsite worker: verifying cash payments for accuracy or reviewing online payments with proof uploaded to confirm the transaction. The station owner and can view the walk-in sales list, gaining insight into all transactions not ordered through the app. The administrator can view walk-in sales logs from water refilling stations which only limits to the date of creation, name of the station, product name, price amount and product amount to monitor the products in the platform while maintaining data privacy. This visibility supports effective monitoring of walk-in sales activities, inventory levels, and financial records.

In Generate Report, the system offers a comprehensive reporting capability to provide insights into the operations and sales of the water refilling station. The station owner can access detailed sales reports in graphical formats, breaking down data by water type, container sizes, transaction categories, water consumption. These reports include information on sales across various locations, such as sales per barangay, peak sales periods, the most saleable container sizes, and feedbacks from deliveries. The system also generates reports based on daily, weekly, and monthly time frames, offering a flexible view of performance over time. All sales, including those made via delivery, onsite purchases, and payment like online and cash, are included in the reports. The system ensures that sales from both app-ordered transactions and walk-in or manual transactions are accounted for, and accurate monitoring of water consumption by adding the water levels from the containers refilled by the onsite workers for delivery and walk-in sales, providing a holistic view of revenue streams. In addition to sales, the system includes specialized reports for canceled orders and damaged inventory. These reports highlight trends or issues in cancellations and document instances of inventory loss due to damage, such as leaks or expired products. This feature allows station owners to identify areas for improvement and minimize future disruptions. Customers also benefit from the reporting feature, with the ability to generate personalized reports such as order history and delivery status updates. These ensure transparency and enable better customer experience. The Administrator has the capability to generate annual sales, inventory, and delivery reports with limited viewing consolidate data across all water refilling stations. This macro-level reporting supports strategic planning and decision-making for the entire network of stations and maintain data privacy.

Overall, the primary function of the system is to:

Administrator

* Create accounts for water refilling stations and Administrator.
* View, activate, and deactivate Station owner’s accounts.
* View and manage inventory lists, order lists, delivery lists, payment lists, sales product lists, and sales lists.
* Generate annual sales, inventory, delivery reports consolidating data across all water refilling stations.

Station Owner

* Create, view, update, and archive employee accounts.
* Manage the station profile, including creating, viewing, updating, and archiving details.
* Create, view, update, and archive products available in the water refilling station.
* View and manage inventory lists, order lists, delivery lists, payment lists, sales product lists, and sales lists.
* Generate sales reports visualized through graphs based on water types, container sizes, transaction categories, sales per location (barangay), payment methods, peak sales times, and water consumption.
* View and generate daily, weekly, and monthly sales reports.
* Generate feedback, operational performance, and water consumption level reports.

Delivery Personnel

* View and update their account information.
* Create, update, and view their inventory list, including the number of gallons deployed for delivery and returned to the station.
* Access the order list to generate optimized delivery routes and confirm deliveries.
* Manage sales lists and sales product lists for transactions not placed through the app, including the type of water, number of gallons, and price.

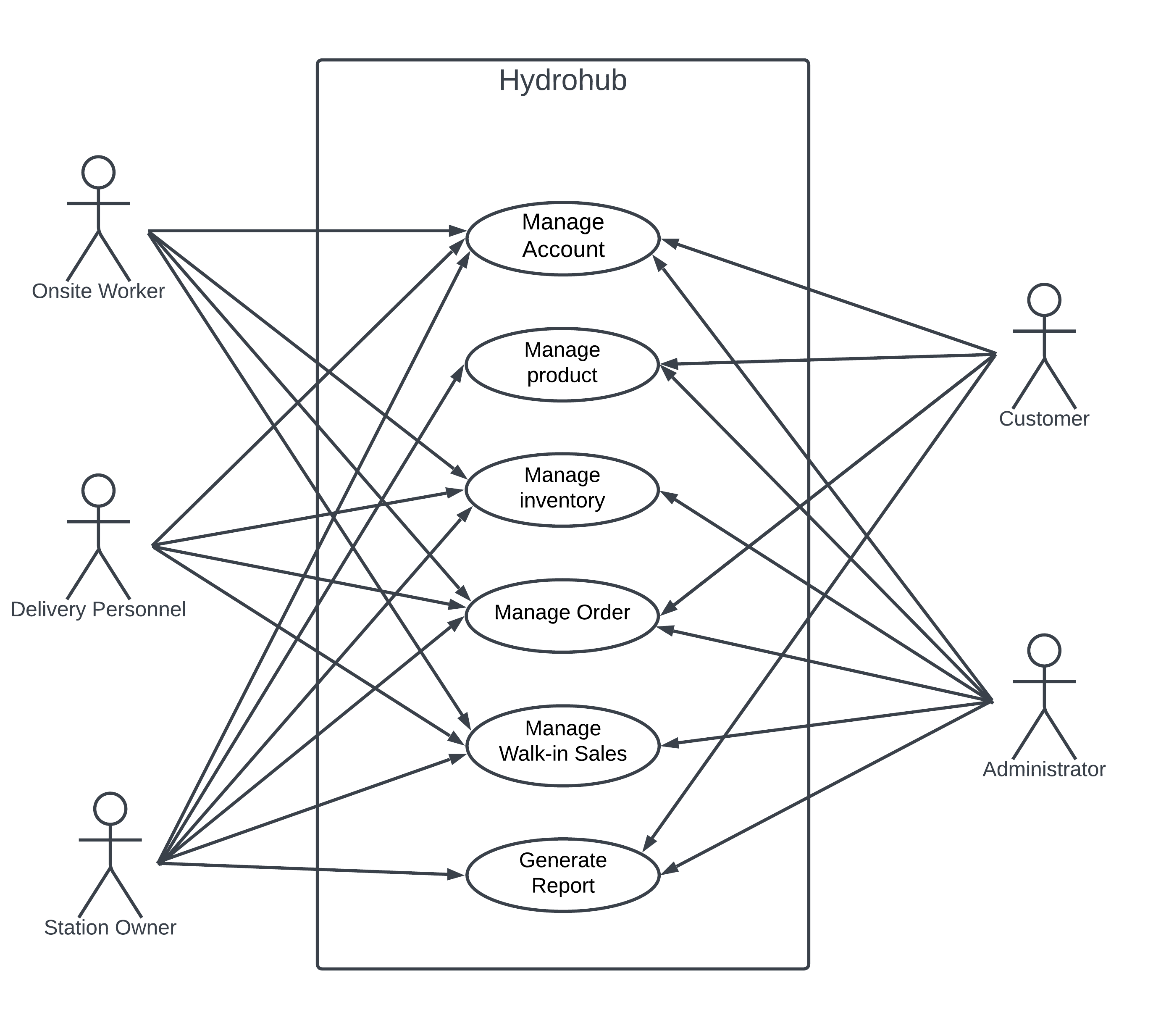
Onsite Worker

* View and update their account information.
* Manage inventory lists, including tracking the number of gallons refilled and discarded due to leakage or expiration.
* Create and manage delivery requests, specifying details such as the type of water, number of gallons, payment method, schedule, and location.
* Confirm and process payments, including verifying cash payments and uploading proof of online payments.
* Manage sales lists and sales product lists for walk-in transactions, including type of water, price, and transaction details.

Customer

* Create, view, update, and archive their account.
* Search for water refilling stations and explore their products and services.
* Place, update, and view orders for delivery or pick-up, specifying water type, number of gallons, payment method, schedule, and location.
* Provide feedback and ratings after delivery is confirmed.
* Generate order history and delivery status reports for tracking purposes.

## **3.1.1 Use Case Diagram**



**Use Case Diagram for Hydrohub: An All-in-One Solution for Managing Sales, Delivery, and Inventory at Water Refilling Station**

## **3.1.2. Use Case Narrative**

### **3.1.2.1. Manage Account (Administrator)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Account |
| **Actor** | Administrator |
| **Pre-condition** | Has an existing account |
| **Description** | Allows the administrator to view, add an account for water refilling stations and update their information. The Administrator can also update his/her profile information. It enables to update the status of the water refilling station owner (active or inactive). |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Enters username, password and clicks “log in" button. | **Step 2:** Displays homepage with sales data within the day of all stations and a menu bar with Accounts button, Report button, and Profile button. |
| **Step 3:** Clicks the "Accounts" button. | **Step 4.1:** If the Customer Account button is clicked, displays the lists of customer users. Return to Step 2  **Step 4.2:** If the Station Account button is clicked, displays the lists of users under the specified user type and its water refilling station. Go to Step 5.  **Step 4.3:** If Create account button is clicked, displays a form to be filled up with “Confirm” and “Cancel” button. Go to Step 9.  **Step 4.4:** If "Profile" button is clicked, it displays the profile information page containing the Update Profile button and Return button. Go to Step 11. |
| **Step 5:** Clicks a button. | **Step 6:** Displays the information of the selected user with Activate button and Deactivate button. |
| **Step 7:** Clicks a specified user. | **Step 8:** If Activate button is clicked, activates and displays the account as active giving permission to operate. Go to Step 4.1.  **Step 8.2:** If Deactivate button is clicked, deactivates and displays the account as inactivate restricting permission to operate. Go to Step 2 |
| **Step 9:** Fills up the form and clicks a button. | **Step 10.1:** If Confirm button is clicked, validates the inputted data, saves the data into the database, and displays a message; Generates a password then sends it via email to the user (Owner of the water refilling station, and the new administrator)). Go to Step 2.  **Step 10.2:** If Cancel button is clicked, Go to Step 2. |
| **Step 11:** Clicks a button. | **Step 12.1:** If Update Profile button is clicked, displays form that is needed to be filled up with Confirm button and Cancel button.  **Step 12.2:** If Return button is clicked, Go to Step 2. |
| **Step 13:** Fills up the form and clicks a button | **Step 14.1:** If Update button is clicked, validates the inputted data, saves the data into the database, and displays a message “Your profile has successfully updated”. Go to Step 2.  **Step 14.2:** If Cancel button is clicked. Go to Step 2.  4: Displays form that is needed to filled up with Update button and Cancel button. |
| **Alternative Path:**   * **Alternative Path 9:** If the specified user has an active transaction, an error message will be displayed "Deactivation is not allowed." Go to Step 2. * **Alternative Path 10.1:** If the administrator did not fill up the required fields for registration (water refilling station and new administrator) or filled invalid entries, a message will be displayed "Please fill up the required fields" or "Invalid entries." Go back to Step 9. * **Alternative Path 14.1:** If the user did not fill up the required fields for an update or filled invalid entries, an error message will be displayed "Please fill up the required fields" or "Invalid entries." Go back to Step 13. | |

### **3.1.2.2. Manage Account (Station Owner)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Account |
| **Actor** | Station Owner |
| **Pre-condition** | * Account is created by the Administrator |
| **Description** | Allows the Station owner to create, view, update, and archive employee accounts (onsite worker, delivery personnel), their profile information, station’s profile information, products, and promotions. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** opens hydrohub application | **Step 2:** Displays homepage with sales data within the day and a menu bar Employee button, Station button, Product button, Logs button, Report button, and Profile button. |
| **Step 3:** Clicks a button | **Step 4.1:** If Employee button is clicked, extracts and displays list of employees with their user types, New Employee button, Remove button and Return button. Go to Step 5.  **Step 4.2:** If Station button is clicked, extracts and displays station’s information, and displays Update button and Return button. Go to Step 13.  **Step 4.3**: If Product button is clicked, extracts and displays station’s product list.  **Step 4.4:** If Logs button is clicked, extracts and displays logs.  **Step 4.5:** If Report button is clicked, extracts and display reports.  **Step 4.6:** If the Profile button is clicked, extract and displays owner’s information, and displays Update button and return button. Go to Step 17. |
| **Step 5:** Clicks a button. | **Step 6.1:** If New Employee button is clicked, displays a form to be filled up with Create Account button and Cancel button.  **Step 6.2:** If Remove button is clicked, asks user to click employee information to remove and displays X button for each employee and Cancel Button. Go to Step 9.  **Step 6.3:** If Return button is clicked, Go to Step 2. |
| **Step 7:** Fills up the form and clicks a button | **Step 8.1:** If Create account button is clicked, validates the inputted data, saves the data into the database, and displays a message, "Employee profile is successfully added." Go back to Step 2.  **Step 8.2:** If Cancel button is clicked, Go to Step 2. |
| **Step 9:** Clicks a button | **Step 10.1:** If X button is clicked, highlights the employee chosen and displays “Are you sure” message with Yes button and No button. Go to Step 11.  **Step 10.2:** If Cancel button is clicked, Go to Step 2. |
| **Step 11:** Clicks a button | **Step 12.1:** If Yes button is clicked, validates the data, archives the data in the database, and displays an "Employee profile is successfully archived." Message. Go back to Step 2.  **Step 12.2:** If No button is clicked, Go to Step 4.1. |
| **Step 13:** Clicks a button | **Step 14.1:** Displays a form with the existing information of the station to be updated with “Confirm” and “Cancel” button.  **Step 14.2:** If Return button is clicked, Go to Step 2. |
| **Step 15:** Fills up the form and clicks a button. | **Step 16.1:** If Confirm button is clicked, validates the inputted data, saves the data into the database, and displays a message, "Station profile is successfully updated." Go back to Step 2.  **Step 16.2:** If Cancel button is clicked, Go to Step 2. |
| **Step 17:** Clicks a button. | **Step 18:** If Update button is clicked, displays form that is needed to filled up with Confirm button and Cancel button.  **Step 18.2:** If Return button is clicked, Go to Step 2. |
| **Step 19:** Fills up the form and clicks a button | **Step 20.1:** If Confirm button is clicked, validates the inputted data, saves the data into the database, and displays a message “Your profile has successfully updated”. Go to Step 2.  **Step 20.2:** If Cancel button is clicked. Go to Step 2. |
| **Alternative Path:**   * **Alternative Path 4.1:** If no data is available for the employee list, display a message "No data available for the selected criteria.". * **Alternative Path 4.2:** If no data is available for the station list, form values are blank. * **Alternative Path 4.3:** If no data is available for the promotion list, display a message "No data available for the selected criteria.". * **Alternative Path 4.6:** If no data is available for the profile, form values are blank * **Alternative Path 8.1:** If the user did not fill up the required fields or filled invalid entries, display a message, "Please fill up the required fields" or "Invalid entries." Go back to Step 7. * **Alternative Path 9.1:** If the user chose an employee with restricting values like delivery is currently going, “Employee cannot be removed”. Go back to Step 8. * **Alternative Path 16.1:** If the user did not fill up the required fields or filled invalid entries, display a message, "Please fill up the required fields" or "Invalid entries." Go back to Step 15. * **Alternative Path 20.1:** If the user did not fill up the required fields or filled invalid entries, display a message, "Please fill up the required fields" or "Invalid entries." Go back to Step 19. | |

### **3.1.2.3. Manage Account – Sign Up (Customer)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Account – Sign up |
| **Actor** | Customer |
| **Pre-condition** | Customer must have a valid contact number. |
| **Description** | Enables the customer to create an account by registering the necessary credentials. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Clicks the "Sign Up" button. | **Step 2:** Displays the form needed by the customer to be filled up. |
| **Step 3:** Fills up the form; clicks the "Sign Up" button. | **Step 4:** Validates the inputted data, saves the data into the database, and displays a message. The system will send a generated One-Time Pin (OTP) via SMS. Display a form to be filled up for the OTP. |
| **Step 5:** Fill-up a form for OTP, inputs a valid 6-digit OTP. | **Step 6:** Validates the inputted data; the page is redirected to the customer homepage. |
| **Alternative Path:**   * **Alternative Path 4**: If the user did not fill up the required fields for registration or filled invalid entries, a message will be displayed "Please fill up the required fields" or "Invalid entries." Go back to Step 2. * **Alternative Path 6**: If the user inputs invalid OTP, display an error message "Incorrect SMS code". Go back to Step 5. | |

### **3.1.2.4. Manage Account (Customer)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Account |
| **Actor** | Customer |
| **Pre-condition** | Customer must have an existing account. |
| **Description** | Enables the customer to create an account by registering the necessary credentials. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Opens Hydrohub app | **Step 2:** Displays available stations and prices with a menu bar that consists, Order button, Profile button, and Report button. Also displays a search bar |
| **Step 3:** Clicks the Profile button. | **Step 4:** Displays the Profile Information page containing an edit profile button and return button. |
| **Step 5:** Clicks a button | **Step 6.1:** If Edit Profile button is clicked, displays a form to be filled up with Confirm button and Cancel button.  **Step 6.2:** If Return button is clicked, Go to Step 2. |
| **Step 11:** Fills up the form and clicks a button | **Step 12.1:** If Confirm button is clicked, validates the inputted data, saves the data into the database, and displays a message, "Your profile has successfully updated." Go back to Step 8.  **Step 12.2:** If Cancel button is clicked, Go to Step 2. |
| **Alternative Path:**   * **Alternative Path 12.1**: If the user did not fill up the required fields for updating the profile or filled invalid entries, a message will be displayed "Please fill up the required fields" or "Invalid entries." Go back to Step 2. | |

### **3.1.2.5. Manage Account (Onsite Worker & Delivery Personnel)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Account |
| **Actor** | Onsite Worker & Delivery Personnel |
| **Pre-condition** | * Pre-registration from the station owner has been approved. * Has received account credentials from the station owner. |
| **Description** | Enables the onsite worker and delivery personnel to update and view their profile information. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Opens Hydrohub app. | **Step 2:** Displays Homepage with a menu bar that consists of Stock button, Order button, Walk-in button, and Profile button |
| **Step 3:** Clicks Profile button. | **Step 4:** Displays the Profile information of the user containing the Update Profile button and return button. |
| **Step 5:** Clicks a button; | **Step 6.1:** If Update Profile button is clicked, displays a form to be filled up with Confirm button and Cancel button.  **Step 6.2:** If Return button is clicked, Go to Step 2 |
| **Step 7:** Fills up the form and clicks a button | **Step 8.1:** If Confirm button is clicked, validates the inputted data, saves the data into the database, and displays a message, " Your profile is successfully updated." Go back to Step 4.  **Step 8.2:** If Cancel button is clicked, Go to step 4 |
| **Alternative Path:**   * **Alternative Path 2:** If the user did not fill up the required fields for registration or filled invalid entries, a message will be displayed "Please fill up the required fields" or "Incorrect Username or Password." Go back to Step 1. * **Alternative Path 8:** If the user did not fill up the required fields for an update or filled invalid entries, an error message will be displayed "Please fill up the required fields" or "Invalid entries." Go back to Step 2. | |

### **3.1.2.6. Manage Products (Administrator)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Products |
| **Actor** | Administrator |
| **Pre-condition** | * Administrator has an existing and active account. * Water refilling stations have products. |
| **Description** | Enables the administrator to create, view, update, archives products available in all of the water refilling station. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Enters username, password and clicks “log in" button. | **Step 2:** Displays Homepage with a menu bar that consists of Inventory button, Product button, Order button, Sales button, Report button, and Profile button |
| **Step 3:** Clicks Product button. | **Step 4:** Displays the list of products available of all water refilling stations. |
| **Alternative Path:**   * **Alternative Path 4.1:** If no inventory items are available, display a message "No inventory items logged." Go back to Step 2. | |

### **3.1.2.7. Manage Products (Station Owner)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Products |
| **Actor** | Station Owner |
| **Pre-condition** | * Station owner has an existing and active account. * Water refilling stations have products. |
| **Description** | Enables the station owner to create, view, update, archives products available in the water refilling station. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Enters username, password and clicks “log in" button. | **Step 2:** Displays Homepage with a menu bar that consists of Stock button, Product button, Order button, Sales button, Logs button, and Profile button |
| **Step 3:** Clicks Product button. | **Step 4:** Displays the list of products available at the water refilling station with New Product button, Update Button next to each product, and Return button. |
| **Step 5:** Clicks a button. | **Step 6.1:** If New Product button is clicked, displays a form to be filled up with Confirm button and Cancel button. Go to Step 7.  **Step 6.2:** If Update button is clicked, displays a form with existing information of the product clicked to be edited. Go to Step  **Step 6.3:** If Return button is clicked, go to Step 2 |
| **Step 7:** Fills up the form and clicks a button | **Step 8.1:** If Confirm button is clicked, validates the inputted data, saves the data into the database, and displays a message, "Product is successfully added." Go back to Step 4.  **Step 8.2:** If Cancel button is clicked, Go to step 2 |
| **Step 9:** Fills up the form and clicks a button | **Step 10.1:** If Confirm button is clicked, validates the inputted data, saves the data into the database, and displays a message, "Product is successfully updated." Go back to Step 4.  **Step 10.2:** If Cancel button is clicked, Go to step 2 |
| **Alternative Path:**   * **Alternative Path 8.2:** If the user did not fill up the required fields for registration or filled invalid entries, a message will be displayed "Please fill up the required fields" or "Incorrect Username or Password." Go back to Step 1. * **Alternative Path 10.1:** If the user did not fill up the required fields for an update or filled invalid entries, an error message will be displayed "Please fill up the required fields" or "Invalid entries." Go back to Step 2. | |

### **3.1.2.8. Manage Products (Customer)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Products |
| **Actor** | Customer |
| **Pre-condition** | * Customer has an existing and active account. * Products are available at the app. |
| **Description** | Enables the customers to view products available of water refilling stations in the platform. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Enters the app | **Step 2:** Displays available stations and prices with a menu bar that consists, Order button, Profile button, and Report button. Also displays a search bar |
| **Step 3:** Scrolls the products and clicks a product or search bar. | **Step 4.1:** If product is clicked, displays the information of the product and water refilling station containing and other products available at the water refilling station with order button and back button. Go to Step 5.  **Step 4.2:** If Search bar is clicked, ask the user to input name of water refilling station. Go to Step |
| **Step 5:** Clicks a button | **Step 6.1:** If Order Button is clicked. displays a form to be filled up with Confirm button and Cancel button.  **Step 6.2:** If back button is clicked, Go to Step 2 |
| **Step 7:** Fills up the search bar and clicks a search button. | **Step 8.1:** Displays the water refilling station and its products. |
| **Step 9:** Scrolls the products and clicks a product. | **Step 10:** If product is clicked, displays the information of the product and water refilling station containing and other products available at the water refilling station with order button and back button. |
| **Alternative Path:**   * **Alternative Path 7:** If the user searched for a water refilling station that is not registered in the platform, a message is displayed “No results found”. Go to Step 2 | |

### **3.1.2.9. Manage Inventory (Administrator)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Inventory |
| **Actor** | Administrator |
| **Pre-condition** | Administrator has existing accounts. |
| **Description** | Allows the administrator to view the inventory lists managed by both the onsite worker and the delivery personnel. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Clicks inventory button | **Step 2:** Displays List of Inventory like the products logged in the inventory |
| **Alternative Paths**   * **Alternative Path 2.1:** If no inventory items are available, display a message "No inventory items logged." Go back to Step 2. | |

### **3.1.2.10. Manage Inventory (Station Owner)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Inventory |
| **Actor** | Station Owner |
| **Pre-condition** | Owner has existing accounts. |
| **Description** | Allows the station owner to view the inventory lists managed by both the onsite worker and the delivery personnel. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Clicks Logs button | **Step 2:** Displays options for which consist of Inventory button, Order button, and Walk-In button, and Report Button. |
| **Step 3:** Clicks inventory button | **Step 4:** Displays List of Inventory like the products logged in the inventory |
| **Alternative Paths**   * **Alternative Path 4:** If no inventory items are available, display a message "No inventory items logged." Go back to Step 2. | |

### **3.1.2.11. Manage Inventory (Onsite Worker)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Inventory |
| **Actor** | Onsite Worker |
| **Pre-condition** | Onsite worker has existing account. |
| **Description** | Allows the onsite to create, update the inventory by logging refilled containers. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Clicks the "Stocks" button. | **Step 2:** Displays the inventory list with a Add New button and Discard button. |
| **Step 3:** Clicks a button | **Step 4.1:** If Add New is clicked, displays a form to add new stock details with Confirm button and Cancel button. Go to Step 5.    **Step 4.2:** If Discard button is clicked, displays a form to add new stock details with Confirm button and Cancel button. Go to Step 7. |
| **Step 5:** Fills up the form and clicks button. | **Step 6.1:** Validates the inputted data, saves the data into the database, adds the existing inventory of refilled containers, automatically adds the time and dated and displays a message “inventory is successfully added”. Go to Step 2.  **Step 6.2:** If Cancel button is clicked. Go to Step 2 |
| **Step 7:** Fills up the form and clicks button. | **Step 8.1:** Validates the inputted data, saves the data into the database, deducts the existing inventory of refilled containers, automatically adds the time and dated and displays a message “inventory is successfully deducted”. Go to Step 2.  **Step 8.2:** If Cancel button is clicked. Go to Step 2 |
| **Alternative Paths**   * **Alternative Path 6.1:** If the required fields are not filled or invalid entries are made, display a message "Please fill up the required fields" or "Invalid entries." Go back to Step 5. * **Alternative Path 8.1:** If the required fields are not filled or invalid entries are made, display a message "Please fill up the required fields" or "Invalid entries." Go back to Step 7. | |

### **3.1.2.12. Manage Inventory (Delivery Personnel)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Stocks |
| **Actor** | Delivery Personnel |
| **Pre-condition** | Delivery Personnel has existing account. |
| **Description** | Allows the delivery personnel to log deployed products that are loaded in the vehicle and returned products that are not sold after the departure. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Clicks the "Stocks" button. | **Step 2:** Displays the delivery inventory list with Add Deployed products button, Add Returned products button, and Edit buttons each log. |
| **Step 3:** Clicks a button | **Step 4.1:** If Deployed button is clicked, displays a form to be filled up with Add products button and Cancel button. Go to Step 5  **Step 4.2:**  If Add Returned button is clicked, displays a form to be filled up with “Add Stocks” and “Cancel”. Go to Step 7  **Step 4.3:** If Edit button is clicked, highlights clicked log and displays form to be edit up with Edit button and Cancel button. |
| **Step 5:** Fills up the form and clicks button. | **Step 6.1:** If Add products button is clicked, validates the inputted data, saves the data into the database, deducts the existing inventory of refilled gallons, automatically stores the time and date, and displays a message Inventory item successfully added”. Go to Step 2.  **Step 6.2:** If Cancel button is clicked, Go to Step 2 |
| **Step 7:** Fills up the form and clicks button. | **Step 8.1:** If Add Stocks button is clicked, validates the inputted data, saves the data into the database, adds the returned containers to the existing stocks of refilled containers, automatically stores the time and date, and displays a message “Inventory item successfully added/updated”. Go to Step 2.  **Step 8.2:** If Cancel button is clicked, Go to step 2. |
| **Step 9:** Fills up the form and clicks button. | **Step 10.1:** If Edit button is clicked, validates the inputted data, changes the data in the database, adds the existing stocks of refilled containers, automatically stores the time and date, and displays a message “Inventory item successfully added/updated”. Go to Step 2.  **Step 10.2:** If Cancel button is clicked, Go to Step 2. |
| **Alternative Path:**   * **Alternative Path 6:** If the required fields are not filled or invalid entries are made, display a message "Please fill up the required fields" or "Invalid entries." Go back to Step 5. * **Alternative Path 8:** If the required fields are not filled or invalid entries are made, display a message "Please fill up the required fields" or "Invalid entries." Go back to Step 7. * **Alternative Path 10:** If the required fields are not filled or invalid entries are made, display a message "Please fill up the required fields" or "Invalid entries." Go back to Step 9. | |

### **3.1.2.13. Manage Order (Administrator)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Order |
| **Actor** | Administrator |
| **Pre-condition** | Station Owner has existing accounts. |
| **Description** | Allows the administrator to view the delivery and order lists by both the customer and water refilling stations and feedbacks made by the customer. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Clicks the Sales button. | |  | | --- | | **Step 2:** Displays options for which consist of Order button and Sales button. | |
| **Step 3:** Clicks Order button | **Step 4:** Displays list Orders button, and Feedback button |

### **3.1.2.14. Manage Order (Station Owner)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Order |
| **Actor** | Station Owner |
| **Pre-condition** | Station Owner has existing accounts. |
| **Description** | Allows the station owner to view the delivery and order lists by both the customer, onsite worker, and the delivery man and feedbacks made by the customer. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Clicks the "Logs" button. | |  | | --- | | **Step 2:** Displays options for which consist of Stocks button, Order button, Sales button. | |
| **Step 3:** Clicks Order button | **Step 4:** Displays list Orders button, and Feedback button |
| **Step 5:** Clicks a button | **Step 6.1:** If Orders button is clicked, displays list of orders  **Step 6.2:** If Feedback button is clicked, displays list of feedbacks |
| **Alternative Paths**   * **Alternative Path 6.1:** If no orders are available, the system displays a message "No orders available." Owner is returned to Step 4 to select another category. * **Alternative Path 6.2:** If no feedbacks are available, the system displays a message "No feedbacks available." Owner is returned to Step 4 to select another category. | |

### **3.1.2.15. Manage Order (Onsite Worker)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Order |
| **Actor** | Onsite Worker |
| **Pre-condition** | * The Onsite Worker has an existing account. |
| **Description** | Allows the Onsite Worker accept or deny order requests, including specifying the number of containers, type of water, and type of order (delivery or pick up), schedule, and location. Also, Onsite workers can confirm pick up orders based on the order details which consists of the number of containers, type of water, and payment method |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Clicks the "Order " button. | |  | | --- | |  |  |  | | --- | | **Step 2:** Displays Order request button, Pick-up list button and Return button. | |
| **Step 3:** Clicks a button | **Step 4.1:** If Order request button is clicked, display list of pending orders (On-the-Day and Pre-order) waiting to be accepted o denied with each order with Accept button, Deny button, and Return button. Go to Step 5  **Step 4.2:** If Pick-up list button is clicked, displays list of accepted pick-up orders to be confirmed and completed with an option for payment (Cash, Online Payment) to be filled up, confirm button and return button. Go to Step 9.  **Step 4.3:** If Return button is clicked, go to Step 2 |
| **Step 5:** Clicks a button | **Step 6.1:** Accept button is clicked, removes order request from the list and moves it to its corresponding list either Pick-up or Delivery. Changes the status of the order from pending to accepted. Go to step 4.1  **Step 6.2:** If Deny button is clicked, displays a form to cite the reason for denial of order with confirm button and cancel button. Go to Step 7  **Step 6.3:** If Return button is clicked, go to Step 2 |
| **Step 7:** Fills up form and clicks a button. | **Step 8.1:** If Confirm button is clicked, removes order request list and notifies the customer of the denied order with its reasons. Changes order from pending to denied. Go to Step 4.1  **Step 8.2:** If Cancel button is clicked, Go to Step 4.1  **Step 8.3:** If Return button is clicked, go to Step 2 |
| **Step 9:** Clicks an option | **Step 10.1:** If Cash option is clicked, asks the user to click the confirm button or cancel button after reviewing the payment. Go to Step 11.  **Step 10.2:** If Online payment option is clicked, display Gcash and Maya option. Go to step 13  **Step 10.3:** If Return button is clicked, go to Step 2 |
| **Step 11:** Clicks a button | **Step 12.1:** If Confirm button is clicked, order is completed and saved in the database, and changed the status of the order to accepted to completed. Inventory is deducted according to the containers exchanged.  **Step 12.2:** If Cancel button is clicked, go to Step 4.2 |
| **Step 13:** Clicks an option | **Step 14.1:** If Gcash button is clicked, photo confirmation from the Gcash app is asked with Confirm button and Cancel button, Go to Step 15.  **Step 14.2:** If Maya button is clicked, photo confirmation from the Maya app is asked with Confirm button and Cancel button, Go to Step 17 |
| **Step 15:** Uploads picture and clicks button | **Step 16:** Verifies photo and saves order in the database and the inventory is deducted according to the containers exchanged. Changes the status of the order from accepted to completed. Go to Step 2 |
| **Step 17:** Uploads picture and clicks button | **Step 18:** Verifies photo and saves order in the database and the and the inventory is deducted according to the containers exchanged. Changes the status of the order from accepted to completed. Go to Step 2 |
| **Alternative Paths**   * **Alternative Path 4.1:** If no existing orders are found when viewing orders, the system displays "No orders available." The Onsite Worker is returned to Step 2 to create a new order or take other actions. * **Alternative Path 4.2:** If no existing orders are found when viewing orders, the system displays "No orders available." The Onsite Worker is returned to Step 2 to create a new order or take other actions. * **Alternative Path 8.**1: If the required fields are not filled or invalid entries are made, display a message "Please fill up the required fields" or "Invalid entries." Go back to Step 7. * **Alternative Path 16**: If the required fields are not filled or invalid entries are made, display a message "Please fill up the required fields" or "Invalid entries." Go back to Step 15. * **Alternative Path 18**: If the required fields are not filled or invalid entries are made, display a message "Please fill up the required fields" or "Invalid entries." Go back to Step 16. | |

### **3.1.2.16. Manage Order (Delivery Man)**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Manage Order | |
| **Actor** | Delivery Man | |
| **Pre-condition** | * The Delivery Man has access to the order management system. * An account has been created for the Delivery Man by the owner. | |
| **Description** | Allows the Delivery Man to manage delivery orders by handling the list of orders, generating delivery routes, and confirming deliveries. | |
| **Typical Course of Event** |  | |
| **Actor Action** | **System response** | |
| **Step 1:** Clicks the " Orders" button. | |  | | --- | |  |  |  | | --- | | **Step 2**: Displays the list of delivery orders and “Start Delivery” button. | | |
| **Step 3**: Clicks “Start Delivery” button. | **Step 4:** Generates delivery routes using Near neighbor and Held-Karp algorithms for efficient delivery and displays 2-dimensional scale of the map of Tagbilaran city with pin icons to be clicked in specified areas for delivery and street guides to follow. | |
| **Step 5**: Clicks” Pin” icon. | **Step 6:** Displays a form with the current details of the order, an area to input photo with photo icon, payment option, and “Confirm Delivery” button. | |
| **Step 7:** Clicks a payment option | | **Step 8.1:** If Cash option is clicked, asks the user to click the confirm button or cancel button after reviewing the payment. Go to Step 19.  **Step 8.2:** If Online payment option is clicked, display Gcash and Maya option. Go to step 11. |
| **Step 9:** Clicks a button | | **Step 10.1:** If Confirm button is clicked, order is completed and saved in the database, and changed the status of the order to accepted to completed. Vehicle inventory is deducted according to the containers exchanged.  **Step 10.2:** If Cancel button is clicked, go to Step 4.2 |
| **Step 11:** Clicks an option | | **Step 12.1:** If Gcash button is clicked, photo confirmation from the Gcash app is asked with Confirm button and Cancel button. Redirects to camera app, go to Step 13.  **Step 12.2:** If Maya button is clicked, photo confirmation from the Maya app is asked with Confirm button and Cancel button. Redirects to camera app, go to Step 13. |
| **Step 13:** Takes and upload picture, and clicks button | | **Step 14:** Verifies photo and asks for the photo confirmation of the delivery. Redirects to the camera app. Go to Step 15. |
| **Step 15:** Takes and upload picture, and clicks button | **Step 16:** Redirects back to Hydrohub app and inputs the pictures in the form. | |
| **Step 17:** Clicks “Complete Delivery ". | **Step 18:** Validates the inputted data, saves the order sale to the database, displays a confirmation message, removes current pin in the map and highlights next destination. Vehicle inventory is deducted according to the containers exchanged and changes the status of the delivery from accepted to completed. Return to step 4 | |
| **Alternative Paths**   * **Alternative Path 3**: If there is no list generated, “No orders to deliver” and returns to homepage. * **Alternative Path 18**: If photos is not inputted, display a message "Please fill up the required fields" Go back to Step 6. | | |

### **3.1.2.17. Manage Order (Customer)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Order |
| **Actor** | Customer |
| **Pre-condition** | Customer has an active account and access and chose a water refilling station |
| **Description** | Allows the customer to create, update, and view delivery orders and feedbacks. This includes specifying the number of gallons, type of deliver, type of water, payment method in Manage Payment, schedule, and location. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Clicks the new order button of a product from the station’s page or the order button in the menu bar | **Step 2.1:** If New Order button is selected from the water refilling station chosen, displays On the Day button, Pre-Order button and return button. Go to Step 3  **Step 2.2:** If Orders button is selected from the menu bar, displays list of orders and return button. Go to Step 9. |
| **Step 3:** Clicks a button | **Step 4.1:** If On the day button is clicked, displays a form to input details for the new order, including number of gallons, type of water, payment method in the Manage Payment, and location. The form has Submit and Cancel button. Go to Step 5  **Step 4.2:** If Pre-Order button is clicked, displays a form to input details for the new order, including number of gallons, type of water, payment method in the Manage Payment, schedule, and location. The form has Submit button and Cancel button. Go to Step 7.  **Step 4.3:** If Return button is clicked, go to Step 2 |
| **Step 5:** Fills out the form and clicks a button | **Step 6.1:** If Submit button is clicked, displays a summary of the order details with a Confirm Order button and Cancel button. Go to Step 9.  **Step 6.2:** If Cancel button is clicked. Go to Step 2. |
| **Step 7:** Clicks a button | **Step 8.1:** If Confirm order is clicked. Validates the inputted data, saves the new order to the database, and displays a confirmation message. Returns to Step 2.  **Step 8.2:** If Cancel button is clicked, Go to Step 2. |
| **Step 9:** Clicks a button. | **Step 10:** If an Order is clicked, displays details of the selected order with options to Update button, Cancel Order button, and Add Feedback button.  **Step 10.3:** If Return button is clicked, go to Step 2 |
| **Step 11:** Clicks a button." | **Step 12.1:** If "Update" is clicked, displays a form with current details of the selected order for updating with Confirm button and Cancel button. Go to Step 13.  **Step 12.2:** If "Cancel Order" is clicked, displays form with questions asking for the reason for cancellation with Confirm button and Cancel button. Go to Step 15.  **Step 12.3:** If Add Feedback button is clicked, displays a form to be filled up with Confirm button and Cancel button. Go to Step 17. |
| **Step 13:** Makes necessary changes and clicks a button | **Step 14.1:** If Confirm button is clicked, validates the inputted data, saves the feedback in the database, and displays a message confirming the update. Returns to Step 2.  **Step 14.2:** If Cancel button is clicked, Go to Step 2. |
| **Step 15:** Fills up the form and clicks button. | **Step 16.1:** If Confirm button is clicked, validates the inputted data, saves the feedback in the database, and displays a message confirming the cancellation. Returns to Step 2.  **Step 16.2:** If Cancel button is clicked, Go to Step 2. |
| **Step 17:** Fills up the form and clicks button. | **Step 18.1:** If Confirm button is clicked, validates the inputted data, saves the feedback in the database, and displays a message confirming the feedback. Returns to Step 2.  **Step 18.2:** If Cancel button is clicked, Go to Step 2. |

|  |
| --- |
| **Alternative Paths**   * **Alternative Path 8.1:** If the required fields are not filled or invalid entries are made when creating or updating an order, the system displays a message "Please fill up the required fields" or "Invalid entries." The Customer is prompted to correct the errors and resubmit. Go to step 7. * **Alternative Path 12.1:** If the order is completed “Update button is not displayed” * **Alternative Path 12.2:** If the order is completed “Cancel button is not displayed” * **Alternative Path 12.3:** If the order is not completed “Add Feedback button is not displayed” * **Alternative Path 14.1:** If the order cannot be updated because it was already been processed, the system displays "Update is not allowed for processed requests." The Customer is returned to Step 2 to view existing orders or create a new one. * **Alternative Path 16.1:** If the order cannot be cancelled because it has already been processed the system displays "Update is not allowed for   processed requests." The Customer is returned to Step 2 to view existing orders.   * **Alternative Path 18.1:** If the required fields are not filled or invalid entries are made when creating or updating an order, the system displays a message "Please fill up the required fields" or "Invalid entries." The Customer is prompted to correct the errors and resubmit. Go to step 17. |

### **3.1.2.18. Manage Walk-in Sales (Adminstrator)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Walk in-Sale |
| **Actor** | Administrator |
| **Pre-condition** | Administrator has existing accounts and access to then walk-in sales management system. |
| **Description** | Allows Administrator to view, manage, and monitor all sales made onsite and offsite that are not ordered in the app |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Clicks the "Sales" button. | |  | | --- | | **Step 2:** Displays options for which consist of Stocks button, Order button, Walk-in sales button. | |
| Step 3: Clicks Walk-in Sales button | **Step 4:** Displays list of Sales logs |
| **Alternative Paths**   * **Alternative Path 4:** If no orders are available, the system displays a message "No sales available." Station Owner is returned to Step 2 to select another category. | |

### **3.1.2.19. Manage Walk-in Sales (Station Owner)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Walk in-Sale |
| **Actor** | Station Owner |
| **Pre-condition** | Station Owner has existing accounts and access to then walk-in sales management system. |
| **Description** | Allows Station Owner to view, manage, and monitor sales made onsite and offsite that are not ordered in the app |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Clicks the "Logs" button. | |  | | --- | | **Step 2:** Displays options for which consist of Stocks button, Order button, Walk-in sales button. | |
| Step 3: Clicks Walk-in Sales button | **Step 4:** Displays list of Sales logs |
| **Alternative Paths**   * **Alternative Path 4:** If no orders are available, the system displays a message "No sales available." Station Owner is returned to Step 2 to select another category. | |

### **3.1.2.20. Manage Sales (Delivery Personnel and Onsite Worker)**

|  |  |
| --- | --- |
| **Use Case Name** | Manage Sale |
| **Actor** | Delivery Personnel and Onsite Worker |
| **Pre-condition** | * The Delivery Personnel and Onsite Worker has an active account. * Transaction has been made with the customer. |
| **Description** | Allows the Delivery Personnel and Onsite worker to view, update, and manage sales data associated with their deliveries that are not scheduled, including logging sales and handling payments from the Manage Payment. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Clicks the "Walk-in" button. | **Step 2:** Displays the list of walk-in sales associated with the Delivery Man, including order details and Add New Sales button and Update button. |
| **Step 3:** Clicks a button. | **Step 4.1:** If Add New Button is clicked, displays a form to add new sales details with payment options, confirm button and Cancel button. Go to Step 5.  **Step 4.2:** If Update button is clicked, displays a form of the existing data to update stock details with Confirm button and Cancel button. Go to Step 9. |
| **Step 5:** Fills up form and clicks an option | **Step 6.1:** If Cash option is clicked, asks the user to click the confirm button or cancel button after reviewing the payment. Go to Step 7.  **Step 6.2:** If Online payment option is clicked, display Gcash and Maya option. Go to step 13 |
| **Step 7:** Clicks a button | **Step 8.1:** If Confirm button is clicked, order is completed and saved in the database. If order is made by the delivery personnel it is off site while orders made by the onsite worker is onsite. Go to Step 2.  **Step 8.2:** If Cancel button is clicked, go to Step 2 |
| **Step 9:** Clicks an option | **Step 10.1:** If Gcash button is clicked, photo confirmation from the Gcash app is asked with Confirm button and Cancel button, Go to Step 15.  **Step 10.2:** If Maya button is clicked, photo confirmation from the Maya app is asked with Confirm button and Cancel button, Go to Step 11 |
| **Step 11:** Uploads picture and clicks button | **Step 12:** Verifies photo and saves order in the database. If order is made by the delivery personnel it is off site while orders made by the onsite worker is onsite. Go to Step 2 |

|  |
| --- |
| **Alternative Paths:**   * **Alternative Path 6.1:** If the required fields for updating the order status are not filled or invalid entries are made, display a message "Please fill up the required fields" or "Invalid entries." Return to Step 5. * **Alternative Path 12:** If the required fields for updating the order status are not filled or invalid entries are made, display a message "Please fill up the required fields" or "Invalid entries." Return to Step 7. |

### **3.1.2.21. Generate Report (Station Owner)**

|  |  |
| --- | --- |
| **Use Case Name** | Generate Report |
| **Actor** | Station owner |
| **Pre-condition** | * Station owner has access to the reporting system. * Station owner has an existing account. |
| **Description** | Allows the station owner to generate various reports such as sales, operations, and transaction reports. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Clicks the "Reports" button. | **Step 2:** Displays Stocks button, Orders button, Payments button, Sales button, Delivery button, Water Consumption. |
| **Step 3:** Clicks a button | **Step 4.1:** If "Stocks" is selected, displays graphs and data for stock management, including discarded or damaged items with filters used for displaying specific data according to time frame.  **Step 4.2:** If "Payments" is selected, displays graphs for payment trends and methods used (e.g., cash, online payments) with filters used for displaying specific data according to time frame.  **Step 4.3:** If "Sales" is selected, displays graphs for sales data categorized by water type, container size, and location with filters used for displaying specific data according to time frame.  **Step 4.4:** If "Delivery" is selected, displays graphs for delivery performance, including fulfillment rates and locations served with filters used for displaying specific data according to time frame.  **Step 4.4:** If "Water Consumption" is selected, displays graphs for Water Consumption with filters used for displaying specific data according to time frame. |
| **Alternative Paths:**   * **Alternative Path 4.1:** If no data is available for the selected report criteria, display a message "No data available for the selected criteria.". Return to step 2. * **Alternative Path 4.2:** If no data is available for the selected report criteria, display a message "No data available for the selected criteria.". Return to step 2. * **Alternative Path 4.3:** If no data is available for the selected report criteria, display a message "No data available for the selected criteria.". Return to step 2. * **Alternative Path 4.4:** If no data is available for the selected report criteria, display a message "No data available for the selected criteria.". Return to step 2. * **Alternative Path 4.5:** If no data is available for the selected report criteria, display a message "No data available for the selected criteria.". Return to step 2. | |

### **3.1.2.22. Generate Report (Customer)**

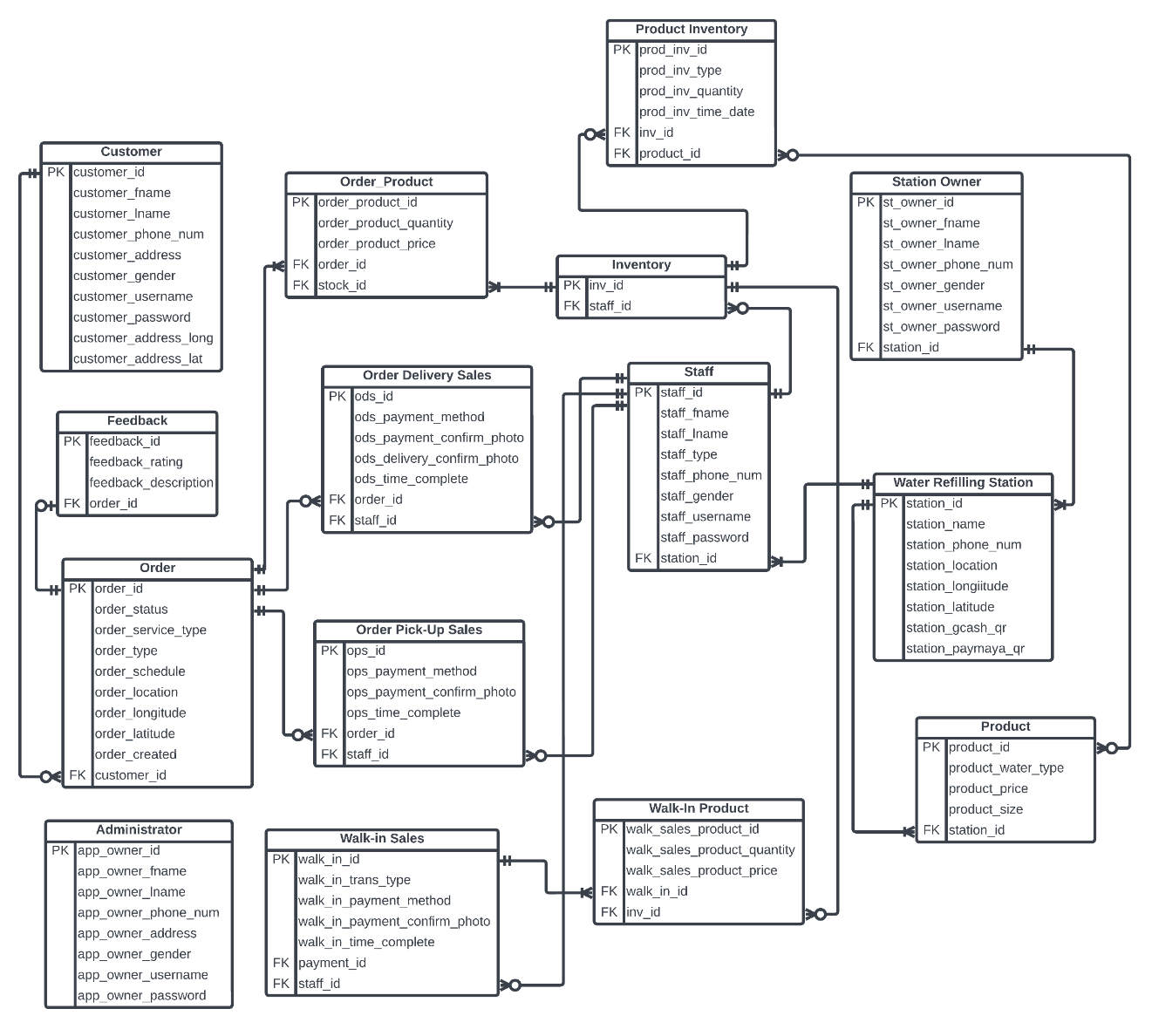
|  |  |
| --- | --- |
| **Use Case Name** | Generate Report |
| **Actor** | Customer |
| **Pre-condition** | Customer has access to the reporting system. |
| **Description** | Allows the customer to generate and view order history and delivery status reports. |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Clicks the "Reports" button. | **Step 2:** Displays options for different types of reports (e.g., Order History, Delivery Status). |
| **Step 3:** Selects the desired report type. | **Step 4:** Displays filters and criteria for generating the report. |
| **Step 5:** Applies filters and criteria, then clicks "Generate Report". | **Step 6:** Processes the request and generates the report, displaying it on the screen. |
| **Alternative Paths:**   * **Alternative Path 6:** If no data is available for the selected report criteria, display a message "No data available for the selected criteria." Go back to Step 4. | |

### **3.1.2.23. Generate Report (Administrator)**

|  |  |
| --- | --- |
| **Use Case Name** | Generate Report |
| **Actor** | Administrator |
| **Pre-condition** | * Administrator has access to the reporting system. * Administrator has an existing account. |
| **Description** | Allows the Administrator to generate annual sales reports for all water refilling stations or specific ones |
| **Typical Course of Event** |  |
| **Actor Action** | **System response** |
| **Step 1:** Clicks the "Reports" button. | **Step 2:** Displays Stocks button, Orders button, Payments button, Sales button, Delivery button, Water Consumption. |
| **Step 3:** Clicks a button | **Step 4.1:** If "Stocks" is selected, displays graphs and data for stock management, including discarded or damaged items of all water refilling stations with filters used for displaying specific data according to time frame and station.  **Step 4.2:** If "Payments" is selected, displays graphs for payment trends and methods used (e.g., cash, online payments) with filters used for displaying specific data according to time frame.  **Step 4.3:** If "Sales" is selected, displays graphs for sales data categorized by water type, container size, and location of all water refilling stations with filters used for displaying specific data according to time frame and station.  **Step 4.4:** If "Delivery" is selected, displays graphs for delivery performance, including fulfillment rates and locations served with filters used for displaying specific data according to time frame and station.  **Step 4.5:** If "Water Consumption" is selected, displays graphs for Water Consumption of all water refilling stations with filters used for displaying specific data according to time frame and station. |
| **Alternative Paths:**   * **Alternative Path 4.1:** If no data is available for the selected report criteria, display a message "No data available for the selected criteria.". Return to step 2. * **Alternative Path 4.2:** If no data is available for the selected report criteria, display a message "No data available for the selected criteria.". Return to step 2. * **Alternative Path 4.3:** If no data is available for the selected report criteria, display a message "No data available for the selected criteria.". Return to step 2. * **Alternative Path 4.4:** If no data is available for the selected report criteria, display a message "No data available for the selected criteria.". Return to step 2. * **Alternative Path 4.5:** If no data is available for the selected report criteria, display a message "No data available for the selected criteria.". Return to step 2. | |

## **3.2. Design**

**3.2.1. Relational Database**



## **3.2.2. File Structure**

### **TABLE 1. CUSTOMER**

This table contains the personal information of the customer.

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Type** | **Description** |
| customer\_id | int(9) | Customer’s id and the primary key of the table. |
| customer\_fname | varchar(30) | Customer’s first name |
| customer\_lname | varchar(30) | Customer’s last name |
| customer\_phone\_num | int(11) | Customer’s Active phone number |
| customer\_address | varchar(100) | Customer’s address for delivery. |
| customer\_gender | enum(‘Male’,’Female’) | Customer’s gender |
| customer\_username | varchar(50) | Customer’s username |
| customer\_password | varchar(255) | Customer’s password |
| customer\_address\_long | varchar(255) | Customer’s address longitude for precise location on map |
| customer\_address\_lat | varchar(255) | Customer’s address latitude for precise location on map |

### **TABLE 2. FEEDBACK**

This table contains the feedback details of the delivery and service created by the customer.

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Type** | **Description** |
| feedback\_id | int(9) | Feedback ID and the primary key of the table. |
| feedback\_rating | int(5) | The star rating of the delivery and service. |
| feedback\_description | varchar(255) | The comments of the customer about the delivery and service. |
| order\_id | int(9) | Order ID, foreign key |

### **TABLE 3. ORDER**

This table contains the order details of the scheduled by the customer.

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Type** | **Description** |
| order\_id | int(9) | Order ID and the primary key of the table. |
| order\_status | enum(‘Accepted’,’On Route’, ‘Completed’, ‘Cancelled’) | The status of the delivery. |
| order\_service\_type | enum(‘On the Day’, ‘Pre-Order’) | The order’s type of service in terms of scheduling the order |
| order\_type | enum (‘Delivery’, ‘Pick-Up’) | The type of order on what way to get the refilled container. |
| order\_schedule | DATETIME | The schedule of the delivery. |
| order\_location | varchar(255) | The location of the delivery. |
| customer\_id | int(9) | Customer’s ID, foreign key |
| order\_longitude | varchar(255) | order’s address longitude for precise location on map |
| order\_latitude | varchar(255) | order’s address latitude for precise location on map |
| order\_created | DATETIME | The time and date the order is created |

### **TABLE 4. ADMINISTRATOR**

This table contains the personal information of the Administrator

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Type** | **Description** |
| admin\_id | int(9) | Administrator’s id and the primary key of the table. |
| admin\_fname | varchar(30) | Administrator’s first name |
| admin\_lname | varchar(30) | Administrator’s last name |
| admin\_phone\_num | int(11) | Administrator’s Active phone number |
| admin\_address | varchar(255) | Administrator’s address. |
| admin\_gender | enum(‘Male’, ’Female’) | Administrator’s gender |
| admin\_username | varchar(50) | Administrator’s username |
| admin\_password | varchar(255) | Administrator’s password |

### **TABLE 5. ORDER\_PRODUCT**

This table contains the information of the products associated with the orders scheduled by the customer.

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Type** | **Description** |
| order\_product\_id | int(9) | Ordered product ID and the primary key of the table. |
| order\_product\_quantity | int(3) | The number of gallons ordered through the app |
| order\_product\_price | float(3,2) | Total price of the products ordered |
| order\_id | int(9) | Order ID, foreign ID |
| stock\_id | int(9) | Stock ID, foreign ID |

### **TABLE 6. ORDER DELIVERY SALES**

This table contains the orders used by the delivery man to generate routes and confirm deliveries.

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Type** | **Description** |
| ods\_id | int(9) | Order delivery sales ID and the primary key of the table. |
| ods\_payment\_method | enum (‘Cash’, ‘GCash’, ‘Maya’) | The payment method used to pay the order |
| ods\_payment\_confirm\_photo | varchar(255) | Contains the photo evidence of payment |
| ods\_delivery\_confirm\_photo | varchar(255) | Contains the photo evidence of delivery. |
| ods\_time\_complete | DATETIME | The order delivery is completed |
| order\_id | int(9) | Order ID, foreign key |
| staff\_id | int(9) | Staff’s ID, foreign key |

### **TABLE 7. ORDER PICK UP SALES**

This table contains the orders used by the onsite worker to confirm pick up orders.

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Type** | **Description** |
| ops\_id | int(9) | Order delivery sales ID and the primary key of the table. |
| ops\_payment\_method | enum (‘Cash’, ‘GCash’, ‘Maya’) | The payment method used to pay the order. |
| ops\_payment\_confirm\_photo | varchar(255) | Contains the photo evidence of payment |
| ops\_time\_complete | DATETIME | the time and date the pick-up order is completed |
| order\_id | int(9) | Order ID, foreign key |
| staff\_id | int(9) | Staff’s ID, foreign key |

### **TABLE 8. WALK-IN SALES**

This table contains the information of the sales of the water refilling station sold through onsite and offsite transactions.

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Type** | **Description** |
| walk\_in\_id | int(9) | Walk in sales ID and the primary key of the table. |
| walk\_in\_trans\_type | enum(‘Onsite’,’Offsite’) | The type of sales transaction |
| walk\_in\_payment\_method | enum (‘Cash’, ‘GCash’, ‘Maya’) | The payment method used to pay the walk-in sales. |
| walk\_in\_payment\_confirm\_photo | varchar(255) | Contains the photo evidence of payment |
| walk\_in\_payment | float(4,2) | Payment of the transaction amount made |
| staff\_id | int(9) | Staff ID, foreign key |

### **TABLE 9. PRODUCT INVENTORY**

This table contains product details that is associated with the inventory list

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Type** | **Description** |
| prod\_inv\_id | int(9) | Product Inventory ID and the primary key of the table. |
| prod\_inv\_type | enum(‘Refilled’, ’Deployed’, ‘Returned’, ‘Discarded’) | Contains the type of logging in the inventory. |
| prodi\_inv\_quantity | int(4) | the number of containers being logged in the inventory. |
| prod\_inv\_time\_date | DATETIME | The date and time when it is logged in the inventory. |
| staff\_id | int(9) | Staff ID, foreign key. |
| product\_id | int(9) | Product ID, foreign key |

### **TABLE 10. INVENTORY**

This table contains the stock details logged by the delivery man and onsite worker

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Type** | **Description** |
| inv\_id | int(9) | Inventory ID and the primary key of the table. |
| staff\_id | int(9) | Staff ID, foreign key. |

### **TABLE 11. STAFF**

This table contains the personal information of the staffs which are the delivery man and the onsite worker.

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Type** | **Description** |
| staff\_id | int(9) | Staff ID and the primary key of the table. |
| staff\_fname | varchar(30) | Staff’s first name |
| staff\_lname | varchar(30) | Staff’s last name |
| staff\_type | enum(‘Onsite’, ‘Delivery’) | The type of staff |
| staff\_phone\_num | int(11) | Staff’s phone number |
| staff\_gender | enum(‘Male’, ‘Female’) | Staff’s gender |
| staff\_username | varchar(50) | Staff’s username |
| staff\_password | varchar(255) | Staff’s password |
| station\_id | int(9) | Station ID, foreign key |

### **TABLE 12. WALK-IN PRODUCT**

This table contains the information of the products associated by the walk-in of the onsite worker and the delivery man sold through onsite and offsite transactions.

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Type** | **Description** |
| walk\_in \_id | int(9) | Walk-in Product ID and the primary key of the table. |
| walk\_in\_quantity | int(3) | The number of containers sold through onsite and offsite transactions |
| walk\_in\_price | float(3,2) | Total price of the products sold |
| walk\_in\_id | int(9) | Walk-In Sales ID, foreign ID |
| product\_id | int(9) | Product ID, foreign ID |

### **TABLE 13. STATION OWNER**

This table contains the personal information of the station owner.

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Type** | **Description** |
| st\_owner\_id | int(9) | Station Owner ID and the primary key of the table. |
| st\_owner\_fname | varchar(30) | Station Owner’s first name |
| st\_owner\_lname | varchar(30) | Station Owner’s last name |
| st\_owner\_phone\_num | int(11) | Station Owner’s phone number |
| st\_owner\_gender | enum(‘Male’, ‘Female’) | Station Owner’s gender |
| st\_owner\_username | varchar(50) | Station Owner’s username |
| st\_owner\_password | varchar(255) | Station Owner’s password |

### **TABLE 14. WATER REFILLING STATION**

This table contains the profile information of the water refilling station.

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Type** | **Description** |
| station\_id | int(9) | Station ID and the primary key of the table. |
| station\_name | varchar(50) | Water Refilling Station’s name |
| station\_phone\_num | int(11) | Water Refilling Station’s phone number |
| station\_location | varchar(100) | Water Refilling Station’s address |
| station\_longitude | varchar(255) | Station’s address longitude for precise location on map |
| station\_longitude | varchar(255) | Station’s address latitude for precise location on map |
| station\_paymaya\_acc | varchar(255) | Water Refilling Station’s  Paymaya account |
| station\_gcash\_qr | varchar(255) | The quick response (QR) code linked with the water refilling station’s GCash account |
| station\_paymaya\_qr | varchar(255) | The quick response (QR) code linked with the water refilling station’s PayMaya account |

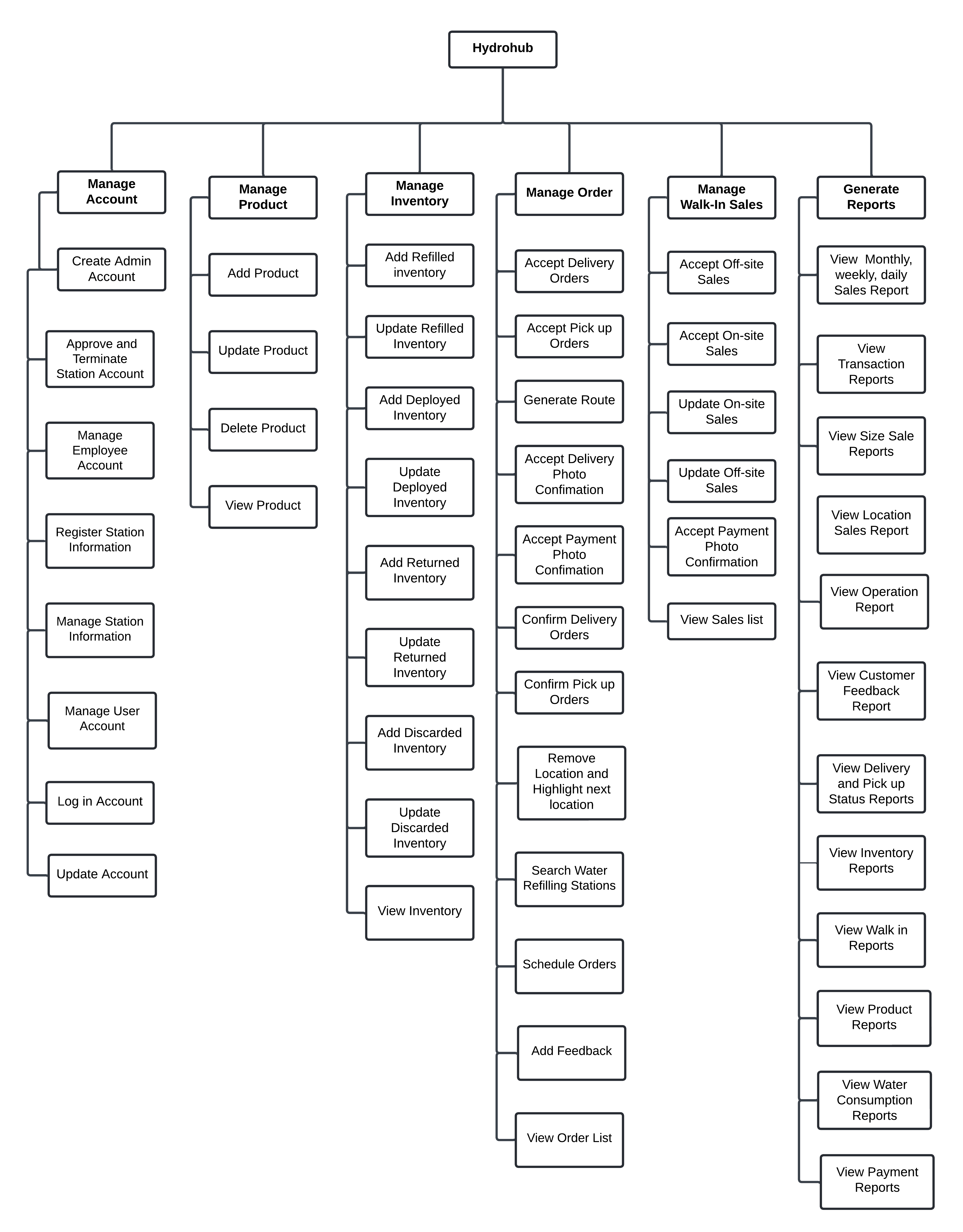
### **TABLE 15. PRODUCT**

This table contains the product information the water refilling station sells.

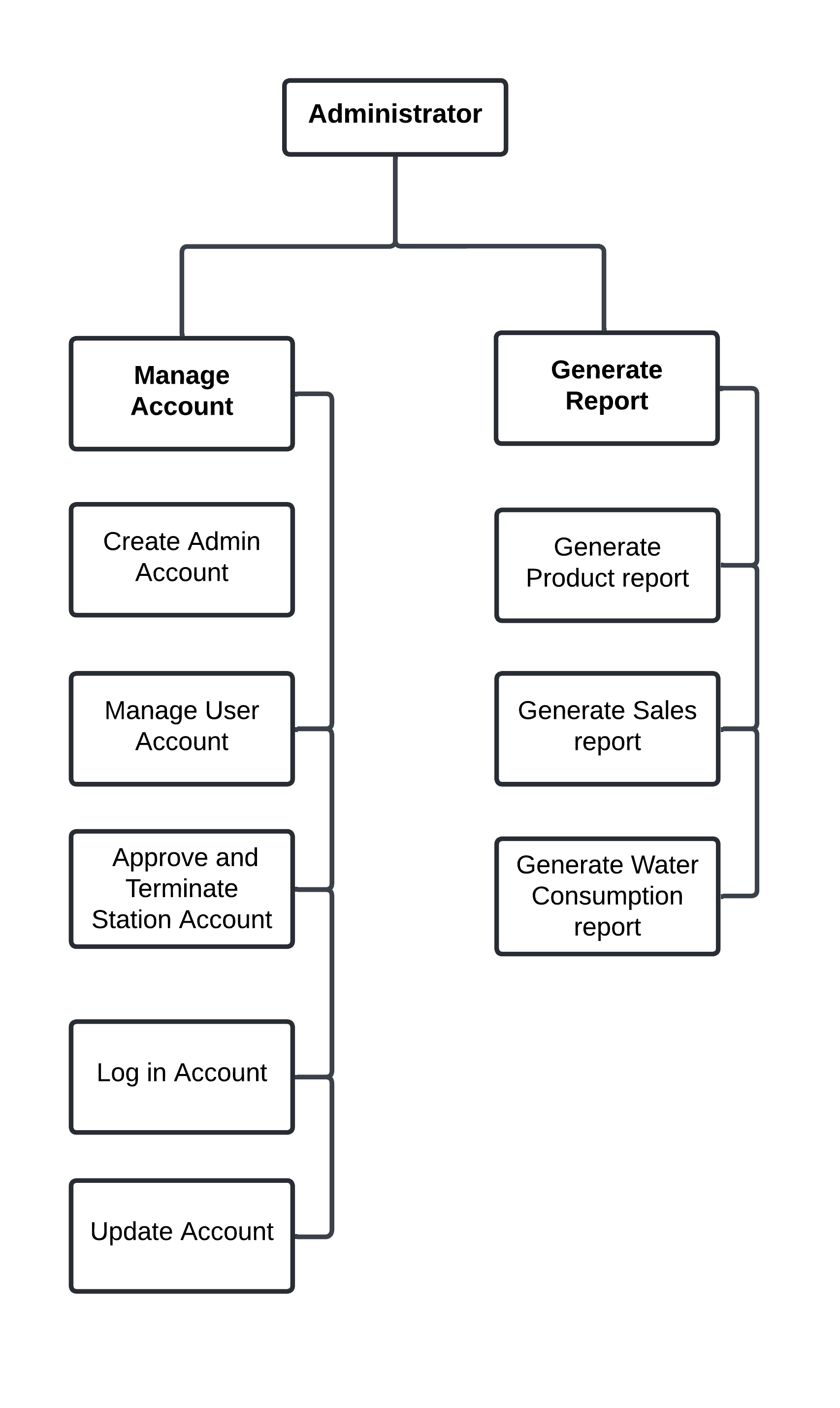
|  |  |  |
| --- | --- | --- |
| **Field Type** | **Type** | **Description** |
| product\_id | int(9) | Product ID and the primary key of the table. |
| product\_water\_type | varchar(30) | Product’s type of water. |
| product\_price | float(3,2) | Product’s price |
| product\_size | varchar(20) | Product’s size of container |
| station\_id | int(9) | Station ID, foreign ID |

## **3.2.3. Program Hierarchy**

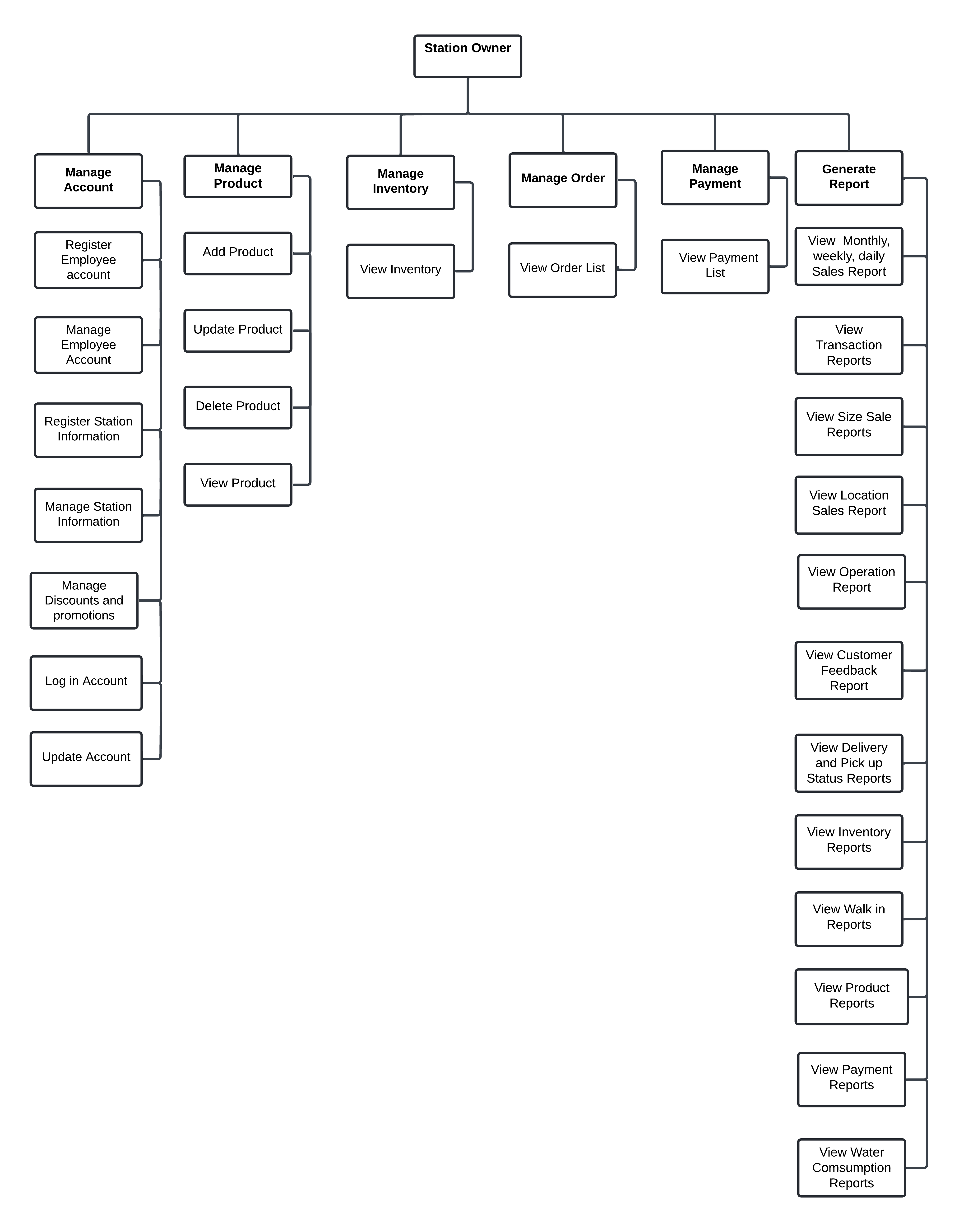
### **3.2.3.1. Top level**



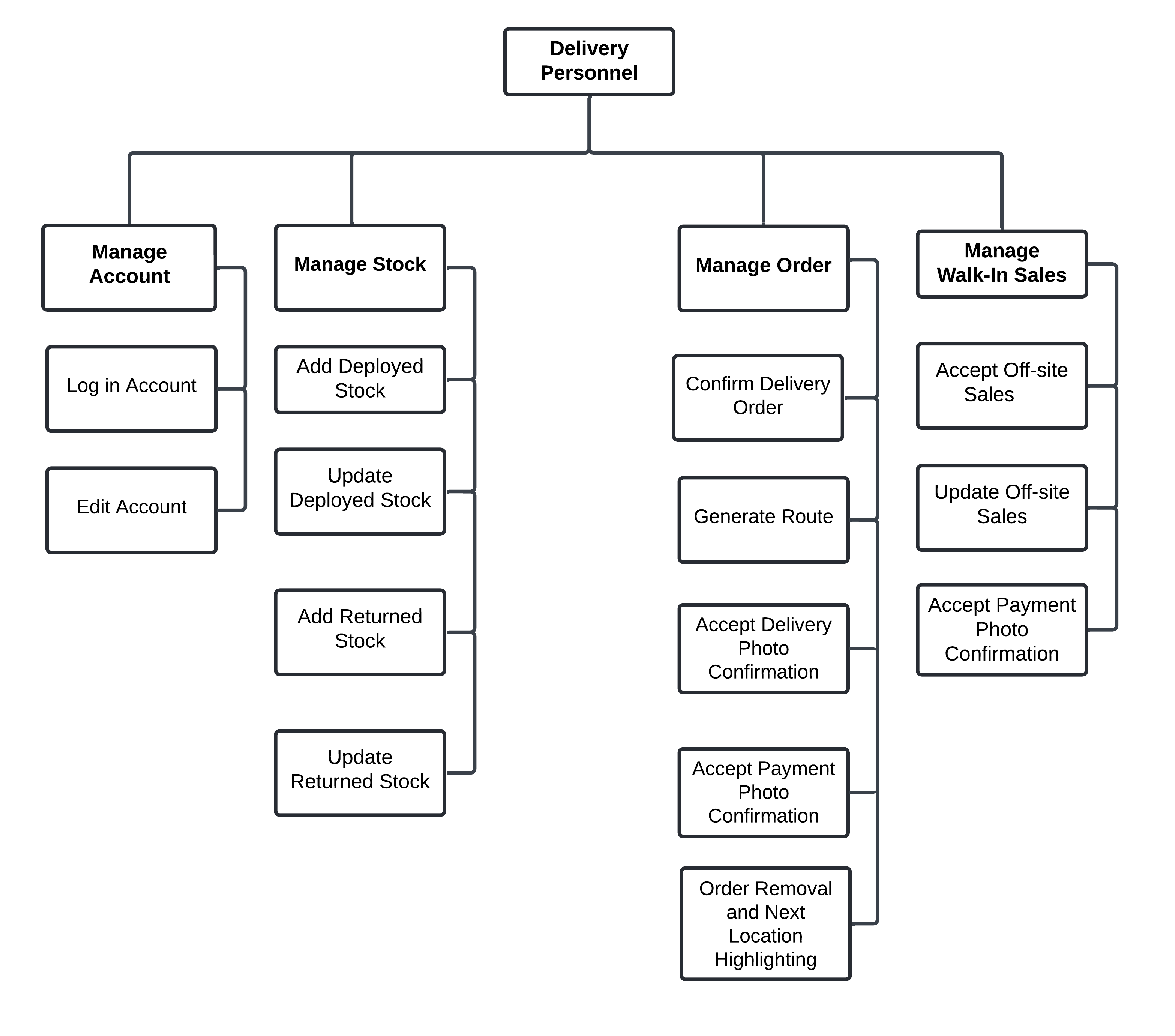
### **3.2.3.2 Administrator**

****

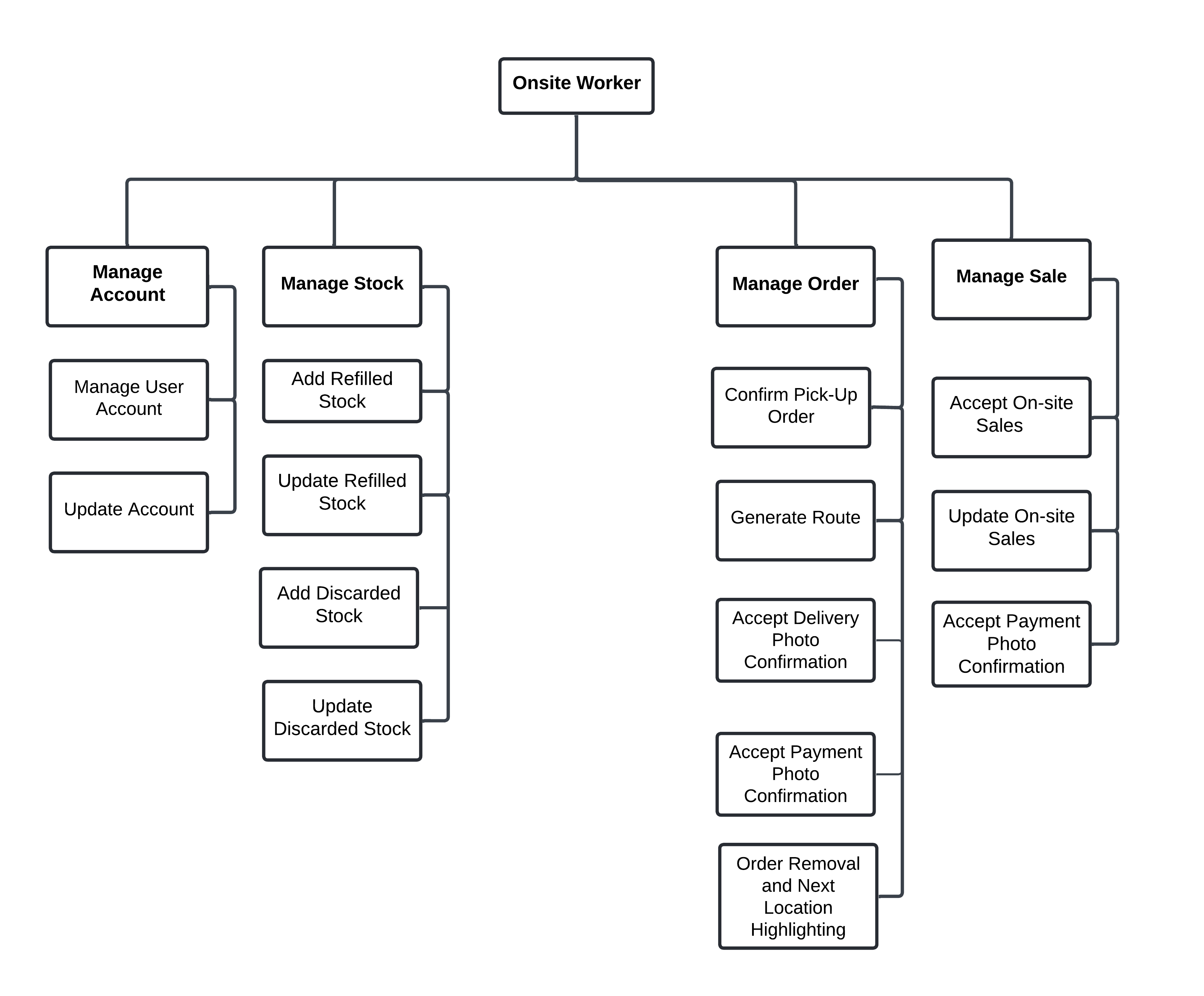
### **3.2.3.3 Station Owner**

****

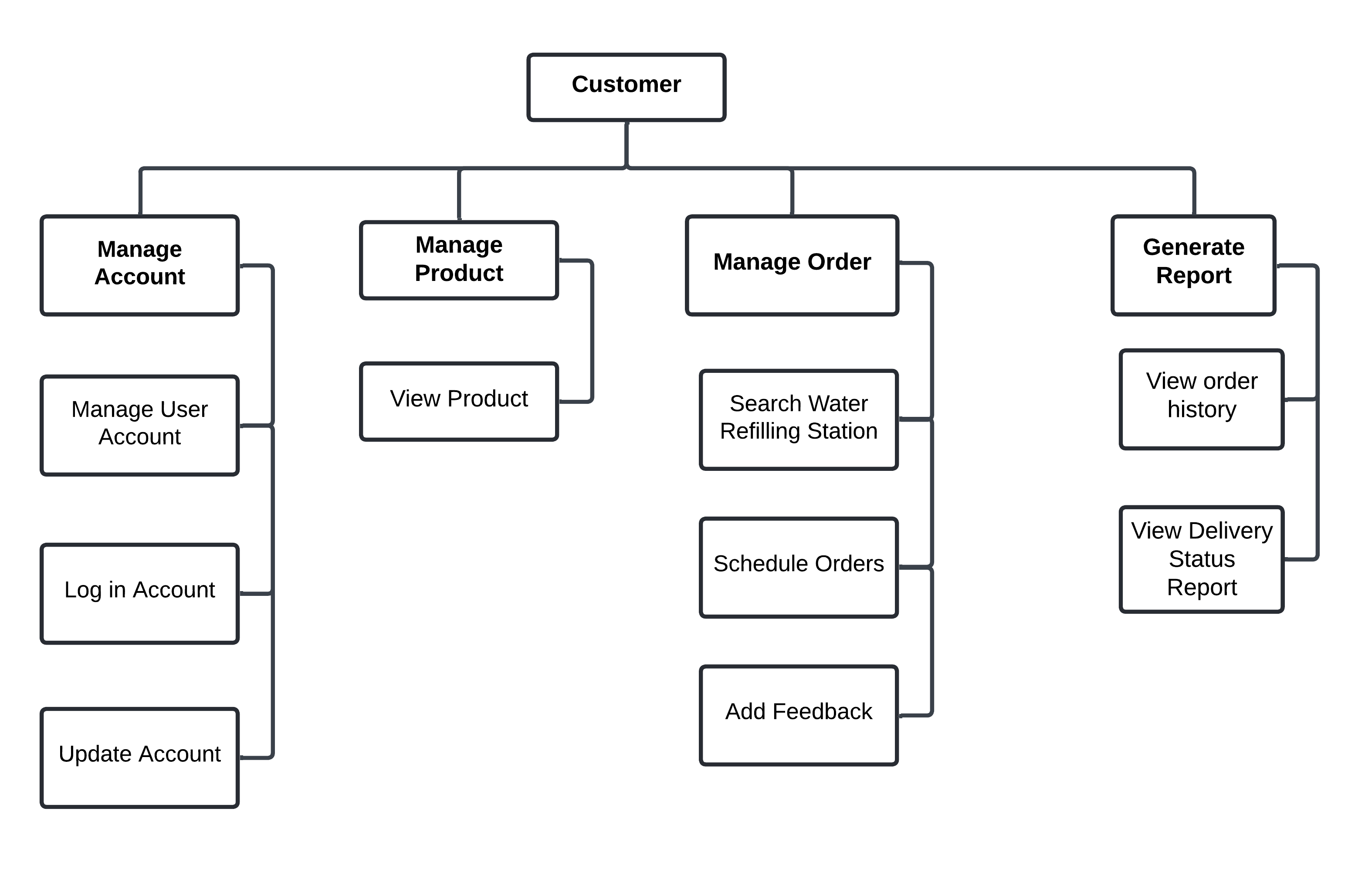
### **3.2.3.4 Delivery Personnel**



### **3.2.3.5 Onsite Worker**



### **3.2.3.6 Customer**



## **3.3 Development and Testing**

Development and Testing This chapter discuss the hardware and software used during the development and testing processes of the HydroHub: An All-in-One Solution for Sales, Inventory, and Delivery at Water Refilling Stations.

## **3.3.1 Development**

During the development phase, the researchers used a laptop with a 12th Gen Intel(R) Core(TM) i5-12450H processor, 8GB RAM, NVIDIA GeForce RTX 4050 GPU, and a 64-bit operating system to build and test the system.

Windows 11 Pro 64-bit was used as the operating system, and the system was developed using the following technologies: Flutter, Node.js, PostgreSQL, Postman, and Visual Studio Code. These frameworks and tools were chosen for both frontend and backend development.

Flutter is a free, open-source UI toolkit used for building natively compiled applications for mobile and web from a single codebase. Node.js is utilized for backend operations due to its event-driven, non-blocking nature, ensuring high performance and scalability. PostgreSQL serves as the database management system (DBMS), providing structured data storage with high reliability. Postman is used for API testing to ensure seamless communication between the frontend and backend.

Visual Studio Code (VS Code), a powerful and lightweight code editor developed by Microsoft, is used as the integrated development environment (IDE). It provides essential debugging, version control, and extension support for efficient software development.

## **3.3.2 Testing**

The testing phase aims to evaluate the system’s functionality, usability, and overall effectiveness in meeting user requirements. Testing is conducted with active participation from water refilling station owners, operational staff, and customers, ensuring a comprehensive assessment of the mobile application.

This study takes place in Tagbilaran City, Bohol, where participants are selected using a purposive sampling technique to ensure balanced representation from all user groups. The study involves three (3) water refilling station owners, six (6) staff members, and three (3) customers, totaling 12 participants. This selection process ensures that feedback is collected from key stakeholders directly involved in refilling station operations and service usage.

The selection criteria for participants include the following: water refilling station owners must have at least (2) years of experience in the industry, staff members must have worked in sales, inventory, or delivery roles at that specific station, and customers must be individuals who have purchased water refills but have not previously used a mobile ordering platform. If any participant withdraws during testing, a replacement is selected based on the same criteria. Participants are excluded from the study based on several criteria to maintain the accuracy and relevance of the testing process. These include water refilling station owners with less than two (2) years of experience in the industry and staff members who are not directly involved in sales, inventory, or delivery operations. Customers who have previously used any mobile or digital platforms to place water refill orders are also excluded, as their prior experience may influence the objectivity of their feedback. Additionally, individuals who are not affiliated with any water refilling station in Tagbilaran City, as well as those who are unwilling or unable to provide informed consent, are not eligible to participate. Persons with visual, motor, or cognitive impairments that significantly hinder interaction with mobile applications—unless the study specifically includes accessibility testing—are also excluded. Participants may be removed from the study if they fail to engage meaningfully with the testing process, provide incomplete responses, or do not meet the minimum participation requirements. Lastly, individuals currently involved in other usability or mobile application research studies are excluded to avoid potential bias in the results.The testing phase lasts for one (1) week, during which participants engage with the system and provide feedback. Before testing begins, participants are asked for their consent to be part of the study. Researchers explain the purpose of the study, the testing process, and how collected data will be used. Participants have the opportunity to ask questions and express any concerns. The testing session lasts between 30 minutes to 1 hour, and participation is entirely voluntary.

Participants in this study have the right to withdraw at any point during the research process without any penalty or consequence. Withdrawal may occur for any reason, including discomfort, schedule conflicts, or a change of willingness to participate. Participants may inform the researchers verbally or in writing of their intention to withdraw, and all data associated with their participation will be excluded from the final analysis if requested. Additionally, participants who show signs of distress during interviews or who fail to meet minimum participation requirements (e.g., incomplete responses or disengagement) may also be respectfully withdrawn by the researchers to maintain the integrity of the data collection process. Ensuring voluntary participation and the right to withdraw upholds the ethical standards of informed consent and respect for participant autonomy. Before using the system, they sign an informed consent form, which is available in English and Cebuano. To protect privacy, the participants’ names, signatures, and personal details are kept confidential throughout data collection, analysis, and reporting.

The testing process is conducted within water refilling stations, where owners and staff typically operate. The locations are selected to ensure a realistic testing environment, reflecting actual operational conditions. The required tools, such as mobile devices and a stable internet connection, are provided to support testing. Researchers also ensure that the testing environment remains private and secure to maintain confidentiality. No monetary transactions or business operations are conducted during the testing process.

The user functionality test is conducted by allowing participants to interact with the system and assess each feature’s performance. Testers follow predefined test cases and compare actual outcomes with expected results. A "PASS" remark signifies that a feature functions correctly, while a "FAIL" remark indicates that a functionality issue exists and requires refinement. Participants are encouraged to provide feedback and suggest improvements. The testing process does not require participants to provide signatures or names.

The user usability test evaluates the ease of use and learnability of Hydrohub’s interface for station owners, delivery personnel, and customers. The usability assessment consists of two key aspects: Software Design and User-Friendliness. Software Design testing involves assessing various interface elements crucial for user experience. The layout of the mobile application is examined to ensure intuitive navigation, while the color scheme is evaluated for visual coherence. Text clarity, including labels and instructions, is analyzed for readability. Buttons, menus, and other interactive elements are tested for accessibility and responsiveness. Additionally, multimedia components, such as images and icons, are reviewed for relevance and user engagement.

For User-Friendliness, testers evaluate how efficiently users can interact with the platform. The readability of text, including font size and style, is assessed for clarity. The navigation between different sections of the application is tested for smooth transitions. The ease of selecting options from menus and dashboards is also examined. Lastly, the application’s responsiveness across various mobile devices, including Android and iOS smartphones, is tested to ensure a consistent and user-friendly experience across platforms. Through these usability testing processes, researchers refine both Software Design and User-Friendliness, enhancing the overall user experience.

The usability test results are analyzed using a Likert scale assessment sheet. Researchers apply statistical procedures, including the weighted mean and grand mean, to interpret collected data. The weighted mean is used to determine the overall usability rating based on each criterion. The 5-point Likert scale is structured as follows:

|  |  |  |
| --- | --- | --- |
| **Value** | **Description** | **Range** |
| 5 | Excellent | 4.21 - 5.00 |
| 4 | Very Good | 3.41 - 4.20 |
| 3 | Average | 2.61 - 3.40 |
| 2 | Fair | 1.81 - 2.60 |

|  |  |  |
| --- | --- | --- |
| 1 | Poor | 1.00 - 1.80 |

***Table 1.*** *Likert Scale*

Points Range Equivalent Descriptive Meaning 5 4.21 - 5.00 E Excellent 4 3.41 - 4.20 VG Very Good 3 2.61 - 3.40 G Good 2 1.81 - 2.60 F Fair 1 1.00 - 1.80 P Poor

The usability ratings help assess the system’s overall user satisfaction and effectiveness. The calculated weighted mean and grand mean provide insight into Hydrohub’s performance based on user feedback.

The findings of the functionality and usability tests indicate overall positive performance, with testers expressing satisfaction with the platform. Each functionality test receives a "PASS" remark, confirming that the features meet system requirements.