

COVER PAGE

DECLARATION

The project is scheduled to begin on [Start Date] and is expected to be completed by [End Date].

Approach

We will be utilizing the Waterfall to execute this project efficiently.

Signatories

<<Insert your name here>>

<<Project supervisor's name>>

I hereby acknowledge that I have thoroughly reviewed and understood the project objectives, deliverables, timeline, budget, and approach. I commit to providing the necessary support and resources to ensure the successful completion of this project within the specified timeframe and budget.

Signature: _____

Printed Name: _____

Date: _____

ACKNOWLEDGEMENT

<<Please write your acknowledgements here>>

<<Also be sure to add Title 2 and update the TOC>>

TABLE OF CONTENTS

INTRODUCTION	XX
INSTALLATION	XX
PREREQUISITES	XX
INSTALLATION STEPS	XX
USAGE.....	XXI
LOGIN INTERFACE	XXI
USER REGISTRATION	XXII
CUSTOMER LOGIN	XXVI
MERCHANT LOGIN	XXXI
DATA MANAGEMENT	XXXI
DATABASE INITIALIZATION	XXXI
MERCHANT BOTTLES TABLE	XXXI
SALES TABLE	XXXII
ORDERS TABLE	XXXII
FUNCTIONALITY	XXXII
CUSTOMER ORDERS.....	XXXII
MERCHANT ACTIONS	XXXII
<i>Tabs</i>	<i>xxxii</i>
TROUBLESHOOTING.....	XLVI
COMMON ISSUES	XLVI
LIMITATIONS	XLVII
SCALABILITY AND CONCURRENCY	XLVII
REFERENCES.....	ERROR! BOOKMARK NOT DEFINED.

ABSTRACT

The Water Dispensing Sales System is a robust and automated solution designed to facilitate the dispensing of water bottles while efficiently recording and managing sales data. This system enables the dispensing of water bottles of various sizes, capturing essential information such as bottle size, unit cost, customer details, merchant name, and transaction date. By accurately tracking sales data, the system provides valuable insights for inventory management, financial reporting, and customer analysis. With its user-friendly interface, this system streamlines the process of water bottle dispensing, ensuring seamless transactions and enhancing customer satisfaction. The Water Dispensing Sales System caters to a wide range of users, including water suppliers, retailers, and consumers, offering them a reliable and efficient solution for managing water sales transactions.

LIST OF FIGURES

Figure 1 First screen upon successful launch	xi
Figure 2 Registration window	xii
Figure 3 Error message if any of the fields is left blank	xiii
Figure 4 Error message for a weak password	xiii
Figure 5 Error message if passwords do not match	xxiv
Figure 6 Warning for a username that has already been taken.....	xxiv
Figure 7 Prompt for assertion	xxv
Figure 8 Success message for a successfully created account (merchant)	xxv
Figure 9 Success message for a successfully created account (customer).....	xxvi
Figure 10 Customer dashboard upon login.....	xxvii
Figure 11 Customer prompt to select a merchant username	xxvii
Figure 12 Merchant selected. This merchant does not have any bottles yet	xxviii
Figure 13 Merchant selected. This merchant has bottles. All the bottles he owns are cascaded on a dropdown list	xxviii
Figure 14 Bottle selected, and all its details are displayed	xxix
Figure 15 Success message for a successfully placed order.....	xxix
Figure 16 Placed orders display board	xxx
Figure 17 Display screen upon login – merchant	xxxi
Figure 18 Bottles tab (Enables the user to add bottles, delete or edit existing ones)	xxxiii
Figure 19 Dispenser interphase	xxxiv
Figure 20 List of all orders from customers.....	xxxv
Figure 21 Sample order ID selected. The details about the order are automatically displayed! Isn't this awesome?	xxxvi
Figure 22 Confirm the action to proceed.....	xxxvii
Figure 23 System detects an order fill. Prompt before dispense as "Order"	xxxviii
Figure 24 Dispense progress shown.....	xxxix
Figure 25 Confirmation message	xxxix
Figure 26 Action recorded in general transaction history.....	xl
Figure 27 Action recorded in orders history	xli
Figure 28 The customer receives a notification that his previous order has been filled!.....	xlii
Figure 29 Notification showing delivered orders	xliii
Figure 30 Cumulative sales history tab	xliv
Figure 31 Orders transaction history	xlv
Figure 32 Download receipts tab	xlvii

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

TOC - Table of Contents

DBMS - Database Management System

GUI - Graphical User Interface

ID - Identifier

UI - User Interface

DB - Database

Additional abbreviations:

App - Application

Tk - Tkinter

SQL - Structured Query Language

OOP - Object-Oriented Programming

PW - Password

Reg - Registration

Auth - Authentication

Config - Configuration

So in summary, the main ones are:

TOC, DBMS, GUI, ID - Used to refer to document sections and data components

UI, DB - Used when discussing application code structure

App, Tk, SQL, OOP - Related to programming concepts and tools

PW, Reg, Auth - For login, registration, authentication functions

Config - For setup like database connections

CHAPTER ONE: SYSTEM OVERVIEW

1.1 Research Objectives

1.1.1 General Objective

To develop an intelligent, fully automated water management system that connects customers and merchants on an integrated platform to enable seamless ordering and delivery of bottled water. The system aims to drive operational efficiency, minimize errors, reduce costs, improve customer experience and provide insightful analytics.

1.1.2 Specific Objectives

- i). Enable customers to conveniently search and browse through catalogs of registered merchants filtered by location, view available products like bottles, cans, jars with price and size attributes, validate account balance sufficiency and place orders online through a user-friendly interface.
- ii). Provide merchants with powerful tools and interfaces to efficiently manage pending orders received from customers across the platform, update order statuses in real-time as they get fulfilled, maintain dynamic inventory levels, record dispensing transactions and quantities, analyze sales metrics, generate detailed reports on demand and configure notification rules.
- iii). Develop a robust notifications engine integrated with the order, inventory and dispensing modules to automatically alert customers via in-app notifications on latest status of their orders, ensuring transparency and building trust. Merchants also receive alerts for low inventory levels to replenish stocks.
- iv). Design a relational database with normalized schemas to store customer profile data including order history, merchant product catalogues encompassing inventory levels, pricing models and availability, dispensing logs, granular transaction histories, sales and revenue reports as well as configurations in a structured MySQL database system accessed via optimized ORM models.
- v). Incorporate role-based access control through a sophisticated user management module to handle customer and merchant registrations, profile management, defining user roles and granular permissions along with administrative functions like system-wide configuration, scheduled data backups and restoration capabilities.
- vi). Implement intuitive and reactive graphical user interfaces with the Tkinter framework consisting of interactive dashboard panels, action buttons and tabs, dropdown selectors, notifications inbox and embedded instructions to guide end-users and enable seamless navigation across various workflows.

- vii). Integrate security best practices including input sanitization, password salting and hashing, multi-factor authentication, role-based authorization and other protocols to ensure safety of platform, data assets and transactions.

1.2 Key Modules and Features

Customer Order Placement and Tracking Module

- i) Search and browse through merchant catalogs
- ii) View products like bottles, jars with attributes
- iii) Select items and place orders
- iv) Integrated payments acceptance
- v) Real-time order status tracking
- vi) Notifications on order updates

Merchant Order Management Module

- i). View and manage pending orders
- ii). Update order status as fulfilled
- iii). Dispensing transactions logs
- iv). Inventory management with alerts
- v). Sales and revenue analytics
- vi). Custom reports generation

Notifications Engine

- i). Order status alerts for customers
- ii). Low inventory alerts for merchants

User Management Module

- i). Customer and merchant signups
- ii). Profile management capabilities
- iii). Role definition and permissions

Administration Module

- i). System-wide configurations
- ii). User administration capabilities
- iii). Data backup schedules and restoration

Security Modules

- i). Authentication protocol,JWT
- ii). Password salting and hashing
- iii). Input validations through OWASP standards
- iv). Role-based access control

Relational Database Management System

- i). MySQL database server
- ii). Normalized database schema
- iii). ORM-based data access

Interactive User Interfaces

- i). Dashboard panels for visualizations
- ii). Action buttons and tabs
- iii). Guided workflow design
- iv). Notifications inbox
- v). Intuitive placements

CHAPTER TWO: HIGH LEVEL DESIGN

The system follows a modular architecture consisting of a Python backend application with a Tkinter GUI frontend interfacing with a MySQL database via the SQLAlchemy ORM.

Key components include:

Presentation Layer: Tkinter based GUI with login pages, customer and merchant dashboards, order management interfaces, dispenser controls, admin configurations etc.

Application Layer: Python modules and packages containing business logic, utilities, algorithms, notifications engine, access controls etc.

Data Access Layer: SQLAlchemy ORM models to abstract MySQL database tables like customers, orders, products, transactions etc.

Database Layer: MySQL relational database management system storing normalized tables.

Security Modules: Authentication, authorization, input validation and other protocols.

Inter-process Communication: REST APIs and SQLAlchemy interfaces to connect UI, business logic and database layers.

Key workflows include:

Customer order placement: Search merchants, select bottles, integrate payment confirmation, notifications.

Merchant order fulfillment: View orders, update status, record dispensing, inventory alerts.

Admin user management: Register users, define roles and permissions.

The presentation layer interacts with the application backend to process and render business logic on the GUI. The backend modules interface with the database via the ORM to complete data read/write operations.

CHAPTER THREE: TECHNICAL DOCUMENTATION

ER Diagrams

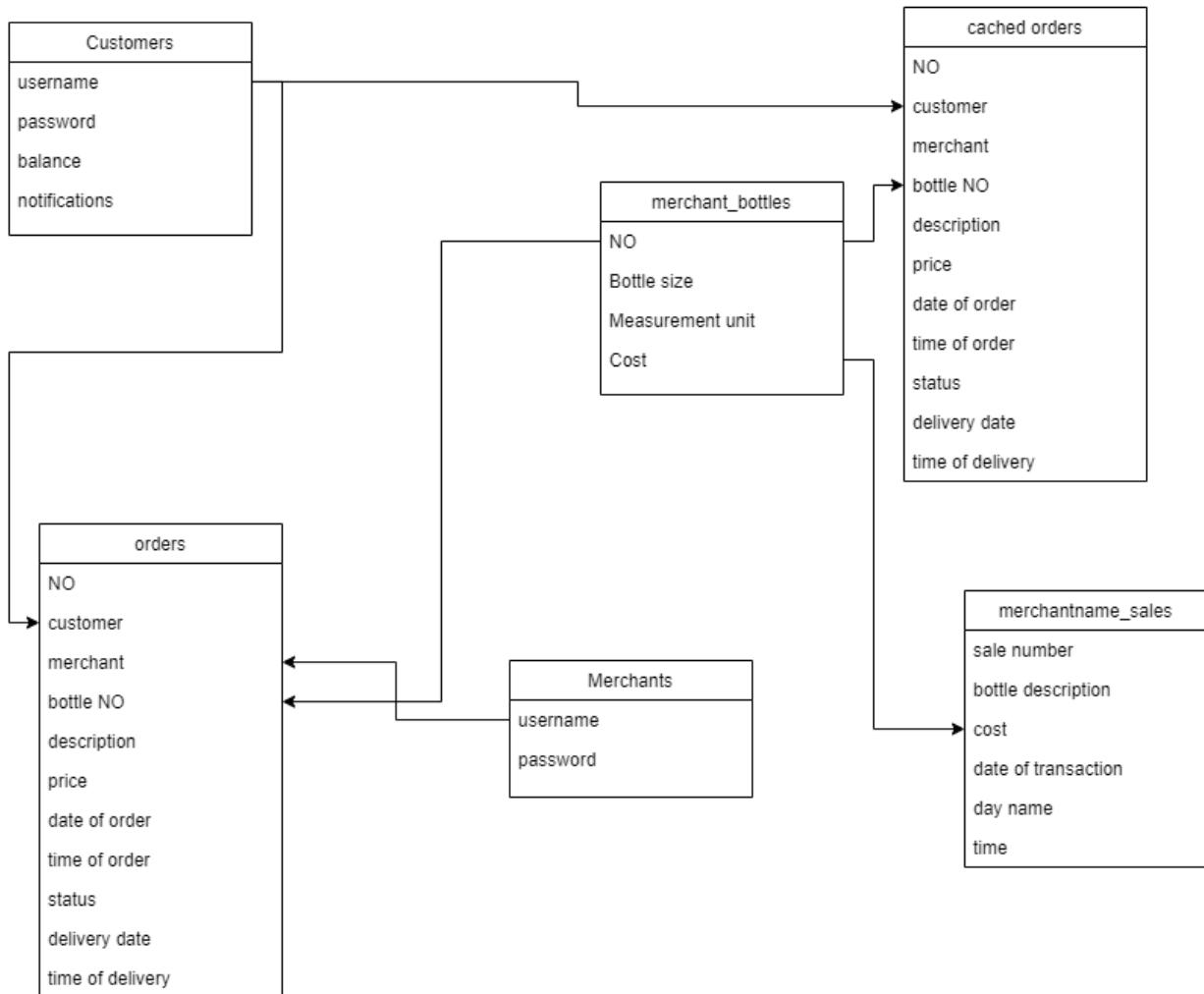


Figure 1 ER Diagram (Database schema)

Entities:

Customers

Attributes: *username, password, balance, cached orders*

Merchants

Attributes: *username, password, merchant_bottles*

Bottles

Attributes: *bottle size, measurement unit, cost*

Orders

Attributes: *customer, merchant, bottle, description, price, date of order, time of order, status, delivery date, time of delivery*

Merchantname_sales

Attributes: *sale number, bottle description, cost, date of transaction, day name, time*

Relationships:

- ❑ Customers place Orders with Merchants for specific Bottles
- ❑ Merchants sell Bottles and fulfill Orders placed by Customers
- ❑ Bottles are linked to Orders placed by Customers with Merchants
- ❑ Merchantname_sales capture transaction data related to Bottle sales by the Merchant
- ❑ The core entities are Customers, Merchants, Bottles and Orders. Customers and Merchants have a many-to-many relationship via Orders. Bottles are linked to Orders. Merchantname_sales stores sales transaction data related to the Merchant's bottle sales.

- ❑ Customers have a balance to pay for orders
- ❑ Merchants have inventory of bottles to sell
- ❑ Bottles have attributes like size, unit, cost
- ❑ Orders track details like bottle, price, dates, status etc.
- ❑ Merchantname_sales stores per-merchant sales data of bottles

CHAPTER FOUR: TEST DOCUMENTATION

In this section, I will highlight critical test points in the application.

Customer order placement

The customer starts by choosing a merchant. This selection cascades a list of all the bottles available on the merchant's bottle table. The bottles are indexed using bottle number, 'NO'.

A list of bottle numbers is shown in a drop-down list (combo box). Upon highlighting the bottle number all the other details such as bottle size, measurement unit and cost are displayed in a canvas.

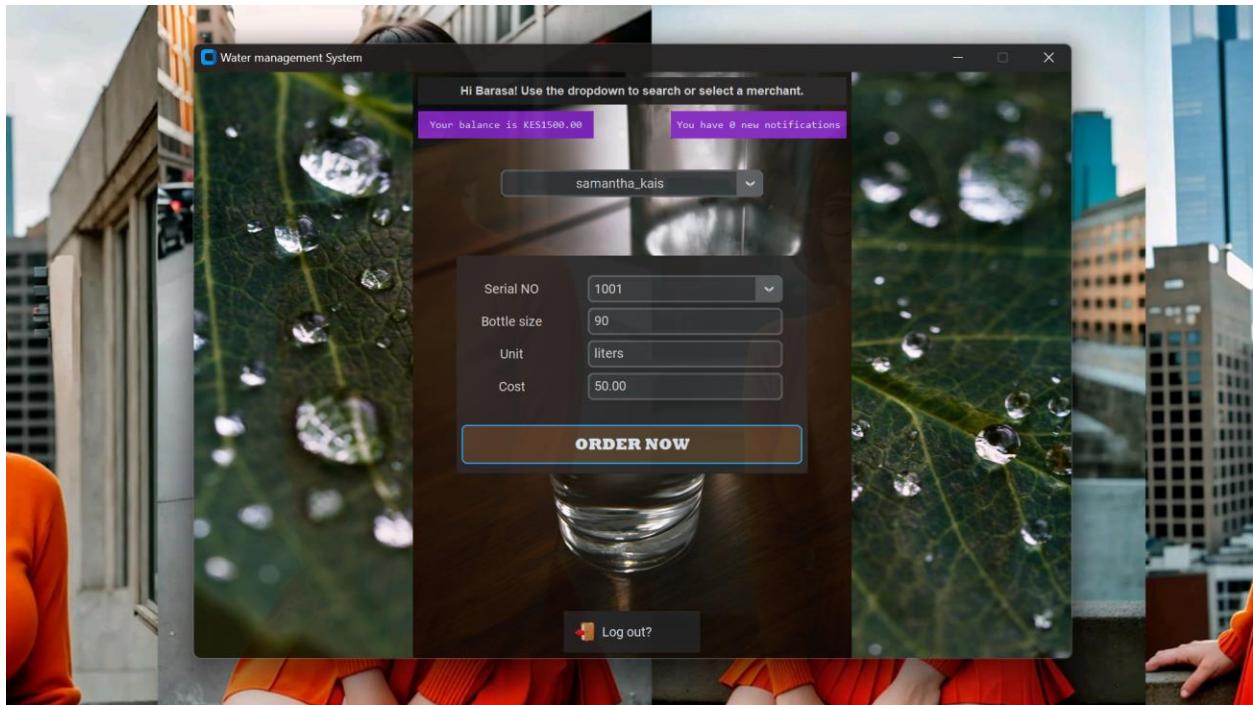


Figure 2 An order of cost within customer's bank account.

If the bank balance is greater than equal to the selected bottle cost, the customer is prompted to confirm and the order is sent to the merchant.

Otherwise, a message box appears showing that they do not have enough funds to make the order:

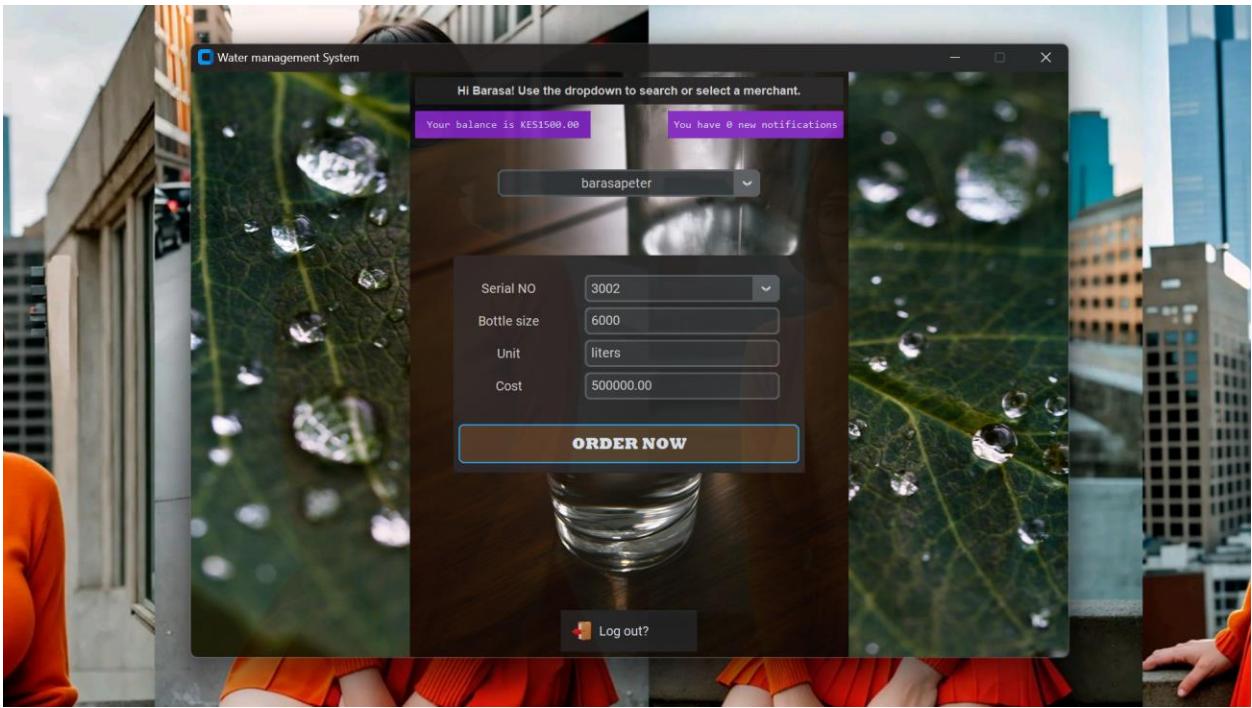


Figure 3 An order exceeding bank account balance.

Here is the warning message displayed:

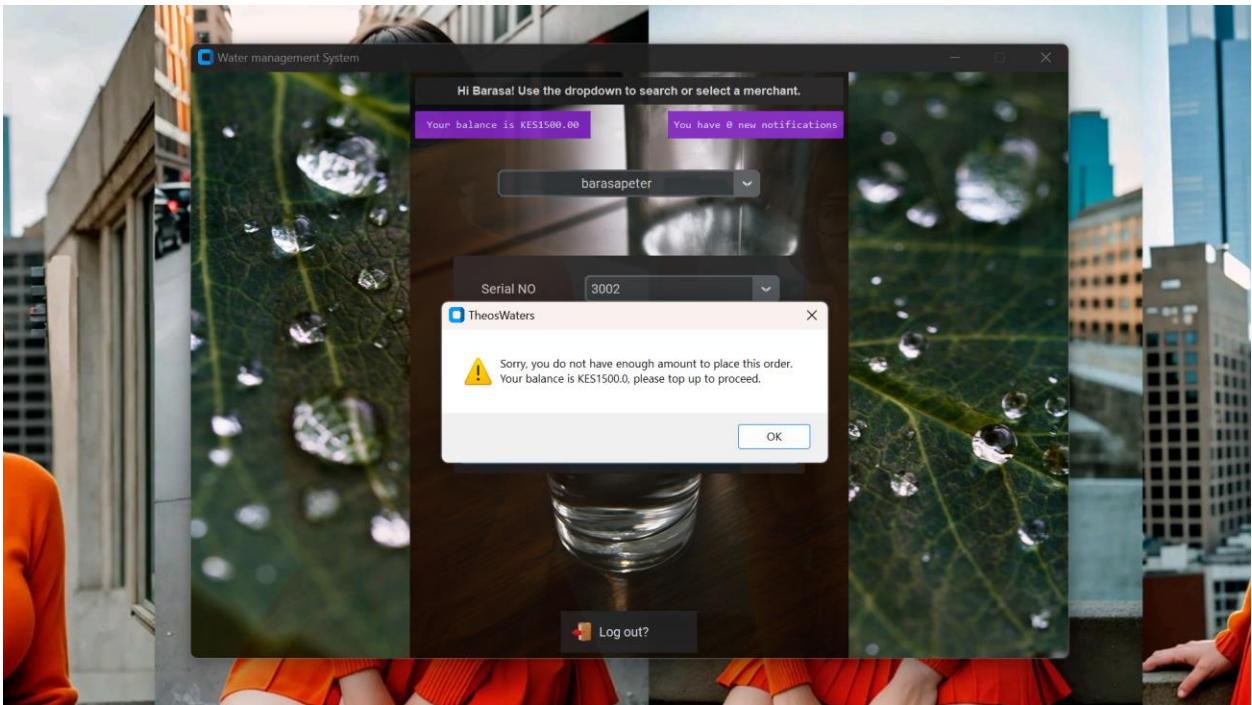


Figure 4 Warning: Insufficient funds to make the purchase.

If the user confirms the order (of course within their bank account balance), a pending order is created and the request is sent to the merchant, see below:

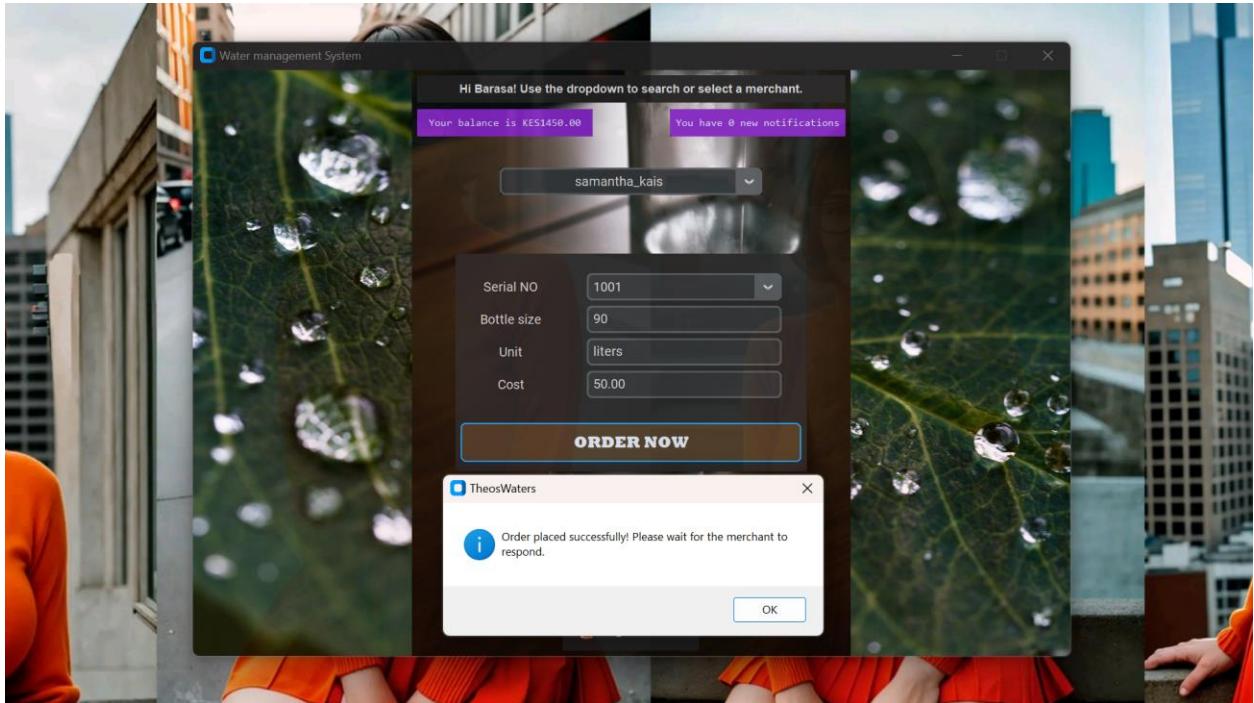
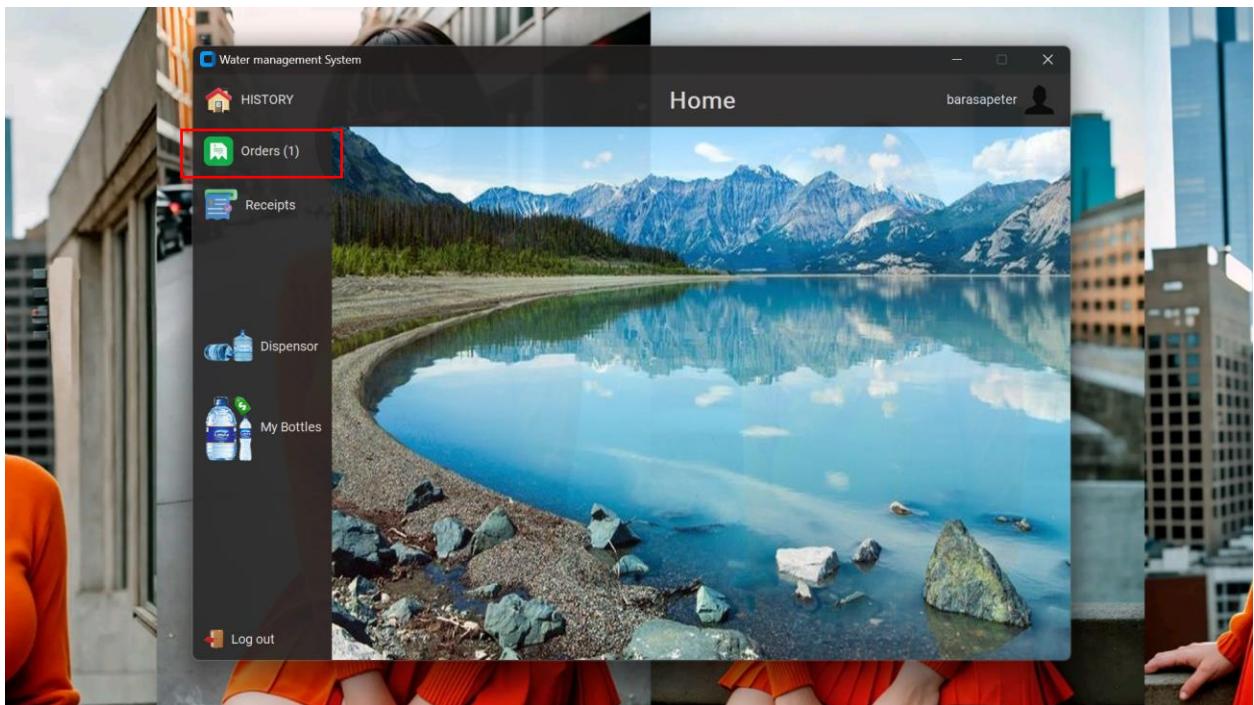


Figure 5 A confirmation message: Order placed successfully.

The amount is automatically deducted after an order has been placed.

On the merchant's side, a notification message is sent:



When a merchant dispenses water, they have an option to fill as 'orders'. In this action, the merchant selects an order from a list of pending orders:

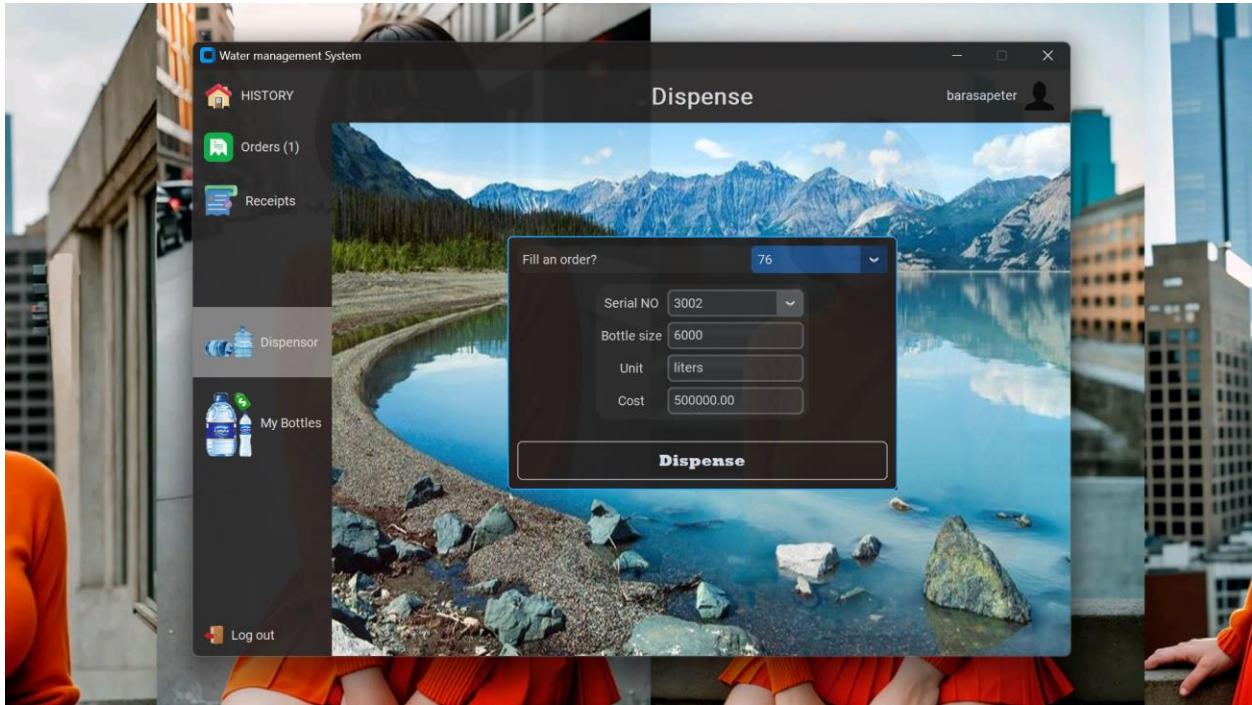


Figure 6 Fill as "order".

Water is dispensed when the Dispense button is clicked. This action decrements the orders number by one:

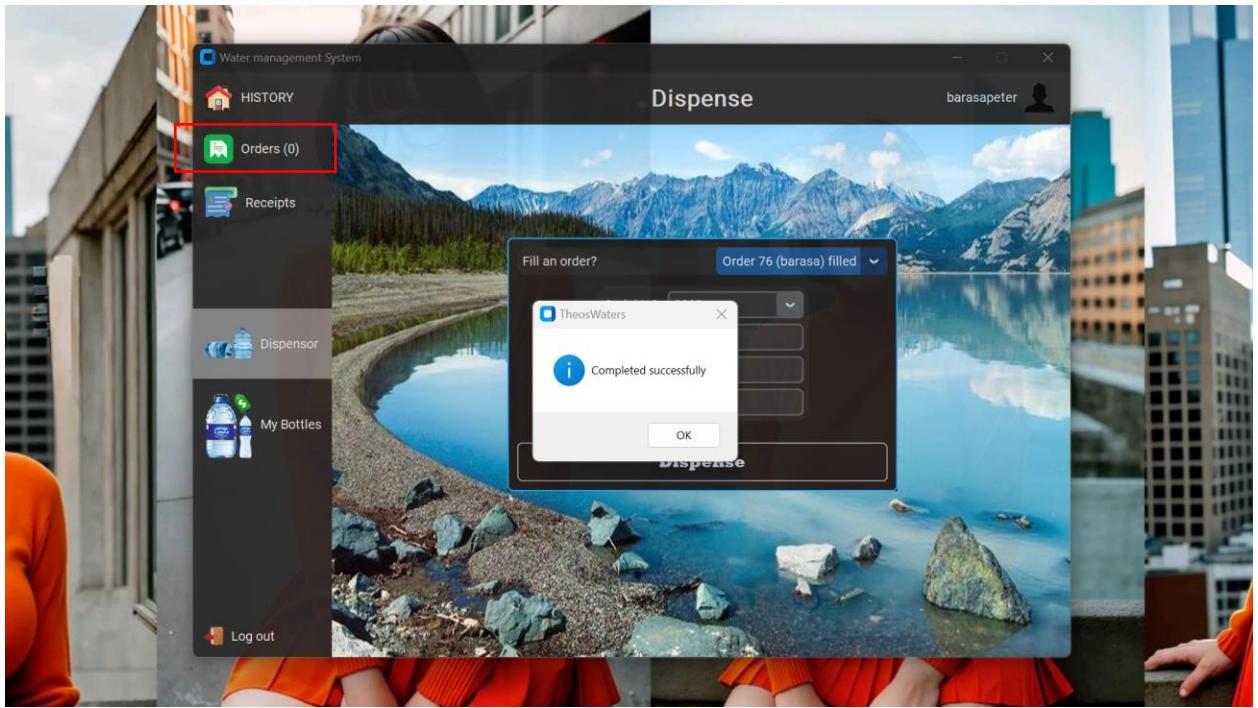


Figure 7 Order numbers decremented.

CHAPTER FIVE: USER MANUAL

Introduction

The Water Management System is a software application developed to manage water-related activities, such as ordering and dispensing water. It is built using Python with Tkinter for the user interface and MySQL as the primary DBMS. The system aims to streamline the process of water ordering and delivery for both customers and merchants, ensuring efficient management and improved user experience.

Installation

Prerequisites

Python: This project needs Python installed. You can download Python from the official Python website ([python.org](https://www.python.org)).

Tkinter: Tkinter is the standard GUI toolkit for Python. It is usually included with Python installations. If not, you can install it using the package manager or by following the official Tkinter installation instructions.

Customtkinter: Customtkinter is a third – party library that is used to create modern interphases. It provides a huge range of customization tools that I have used throughout the project.

TTKBOOTSTRAP: This is another third – party Python library I have used . It provides a huge range of customizable widgets some of which I have used in the project.

MySQL: Install MySQL and ensure it is running on your system. You can download MySQL from the official MySQL website.

There are some other few third – party libraries I have used as well. All of them can be installed easily by running pip install -r requirements.txt. You will run this command on the terminal right in the folder containing the project files.

Installation Steps

Clone the Water Management System repository from the provided source or extract the project files to a local directory.

Open a terminal or command prompt and navigate to the project directory.

Install the required dependencies by running the command: pip install -r requirements.txt.

Create a MySQL database and ensure it is running.

Update the database connection settings in the system configuration file.

Run the main application file using Python: python main.py.

Usage

Login Interface

When the system is launched, a login window is displayed, prompting the user to enter their username and password.

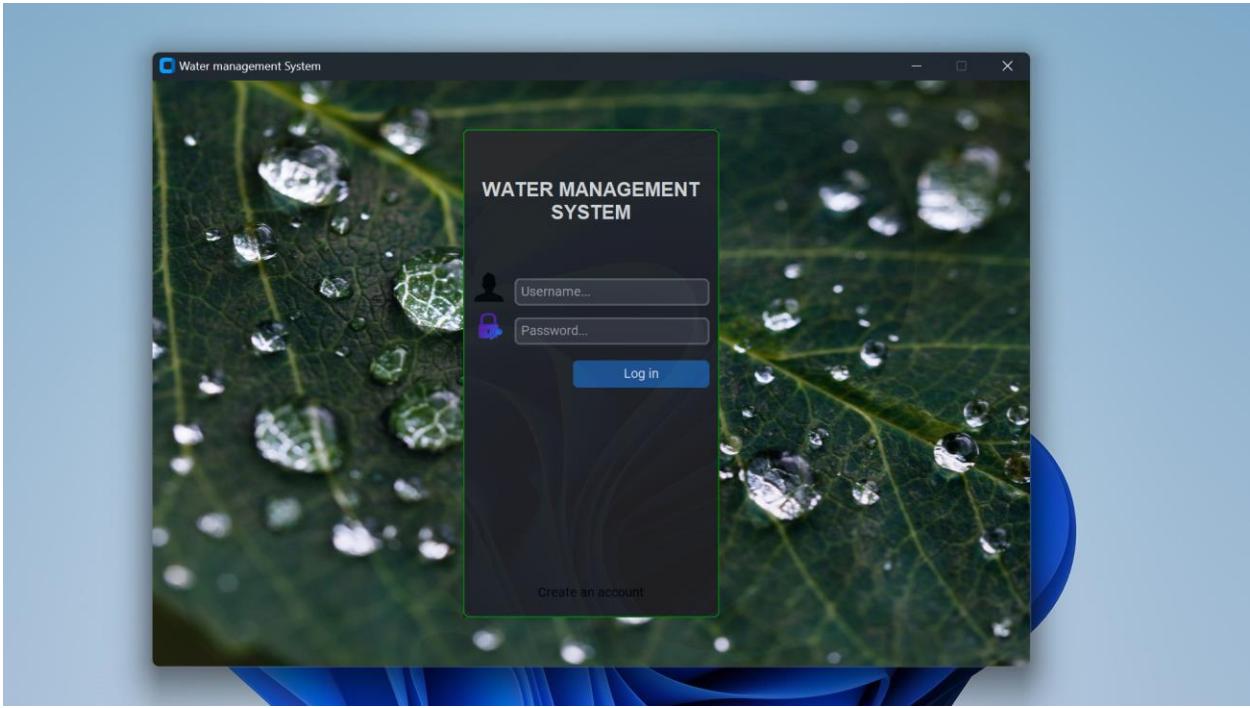


Figure 8 First screen upon successful launch

User Registration

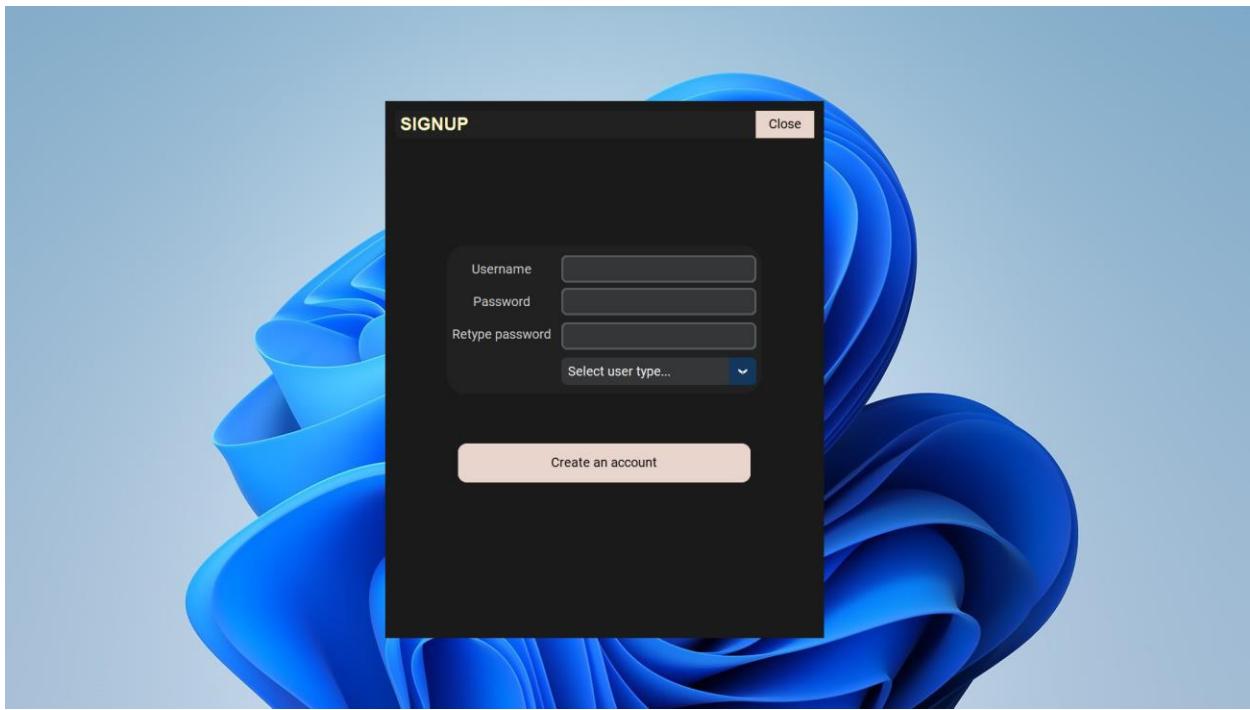


Figure 9 Registration window

If a user does not have an account, they can click on the "Sign Up" button to open a registration window.

In the registration window, the user is required to enter a username and password.

The password must meet the specified strength rules, including a minimum length of 8 characters, at least one number, one lowercase letter, one uppercase letter, and one special character.

The user is also prompted to select their user type: Merchant or Customer.

An account is created if all fields have been entered correctly and the username does not already exist in the system's database.

If there are any errors, appropriate error messages are displayed as shown in the figures below.

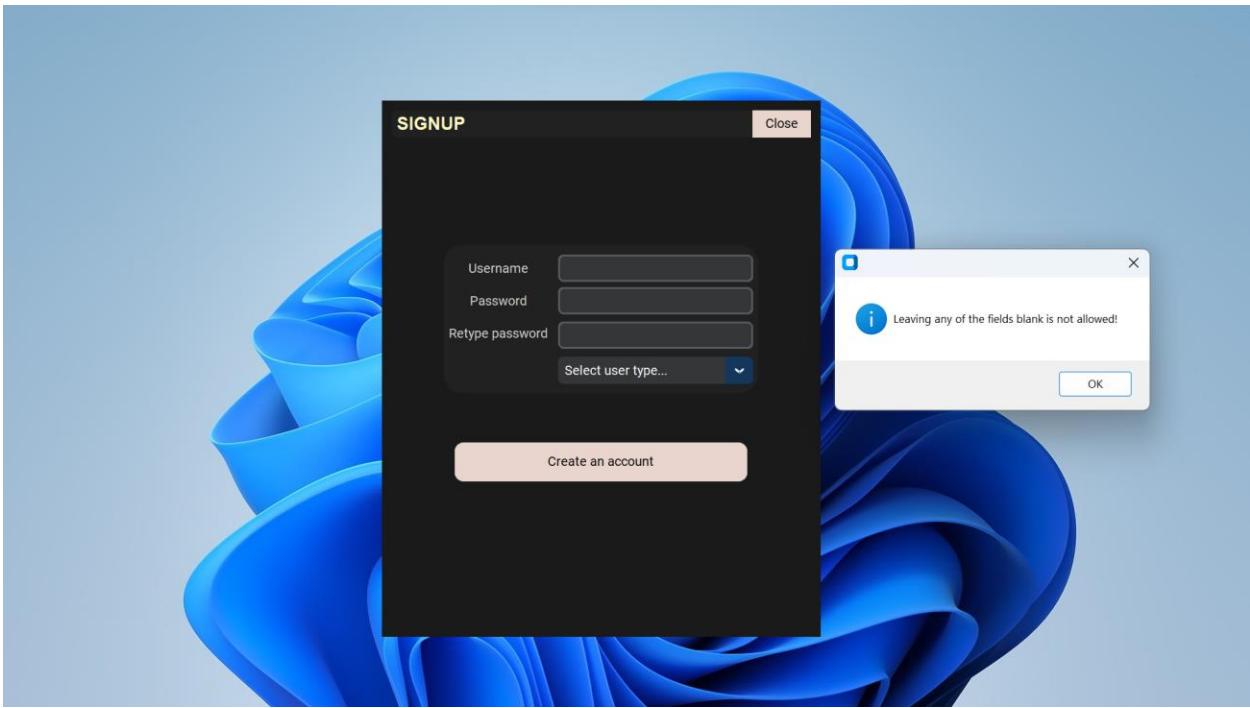


Figure 10 Error message if any of the fields is left blank

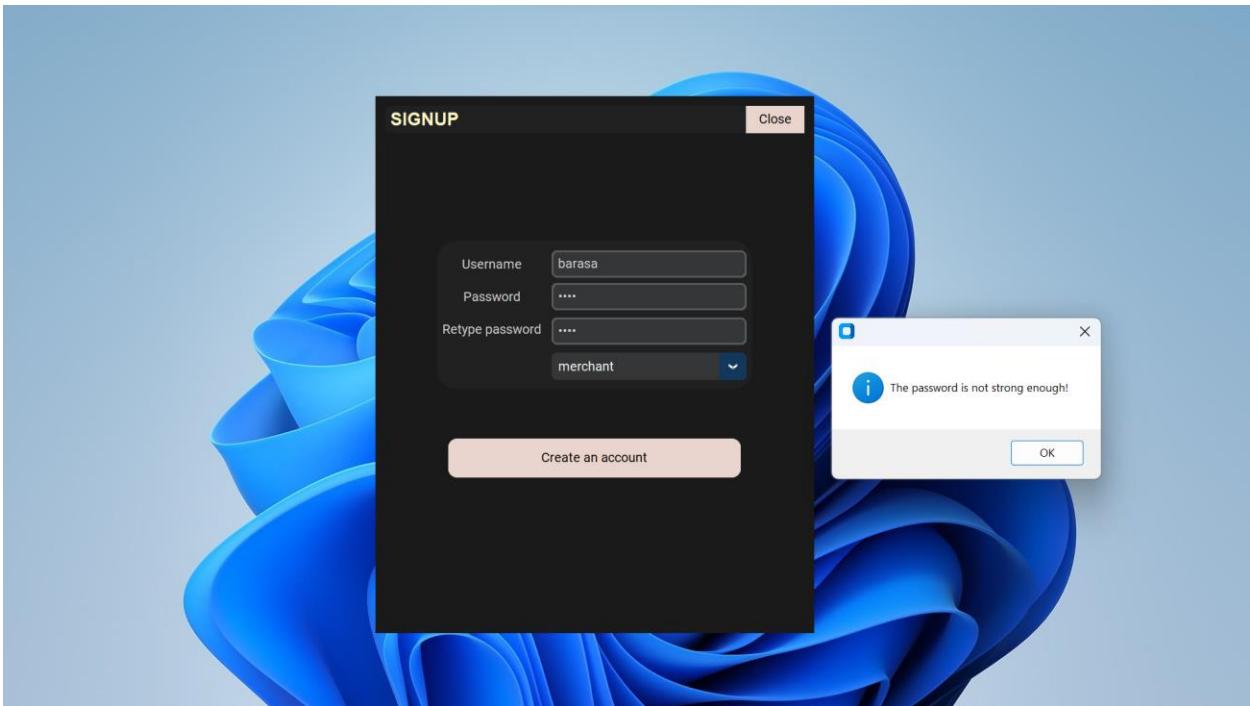


Figure 11 Error message for a weak password

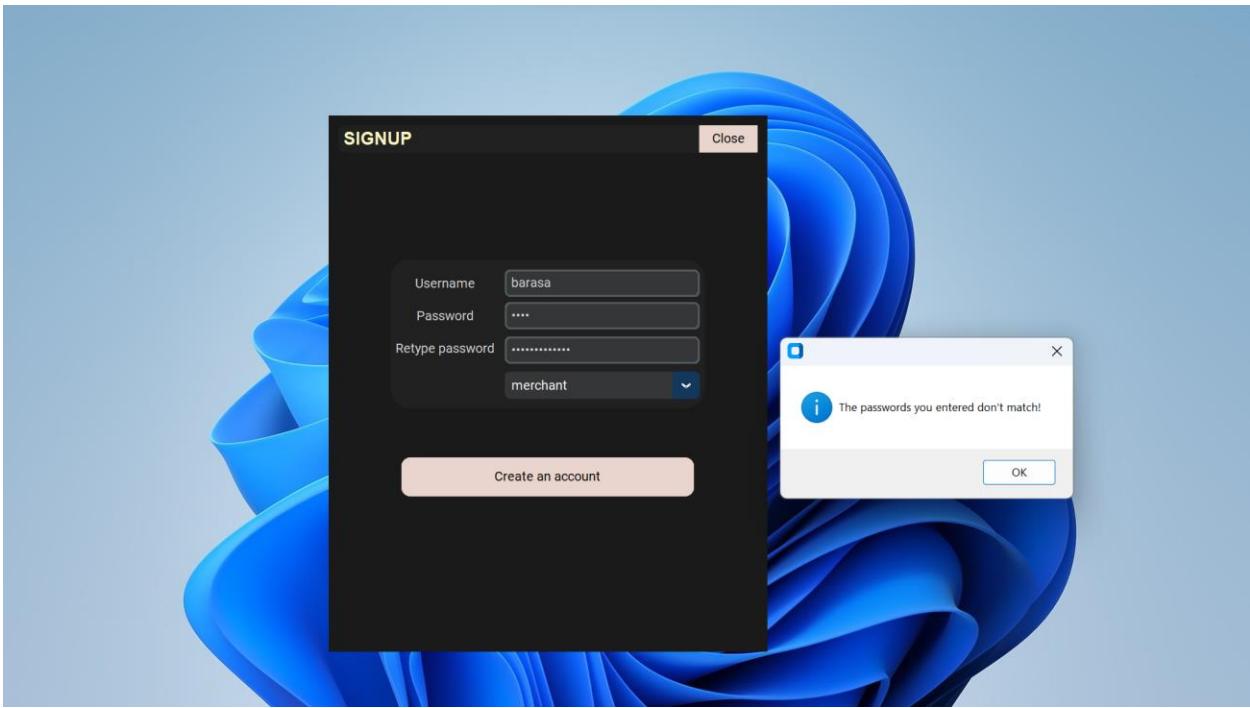


Figure 12 Error message if passwords do not match

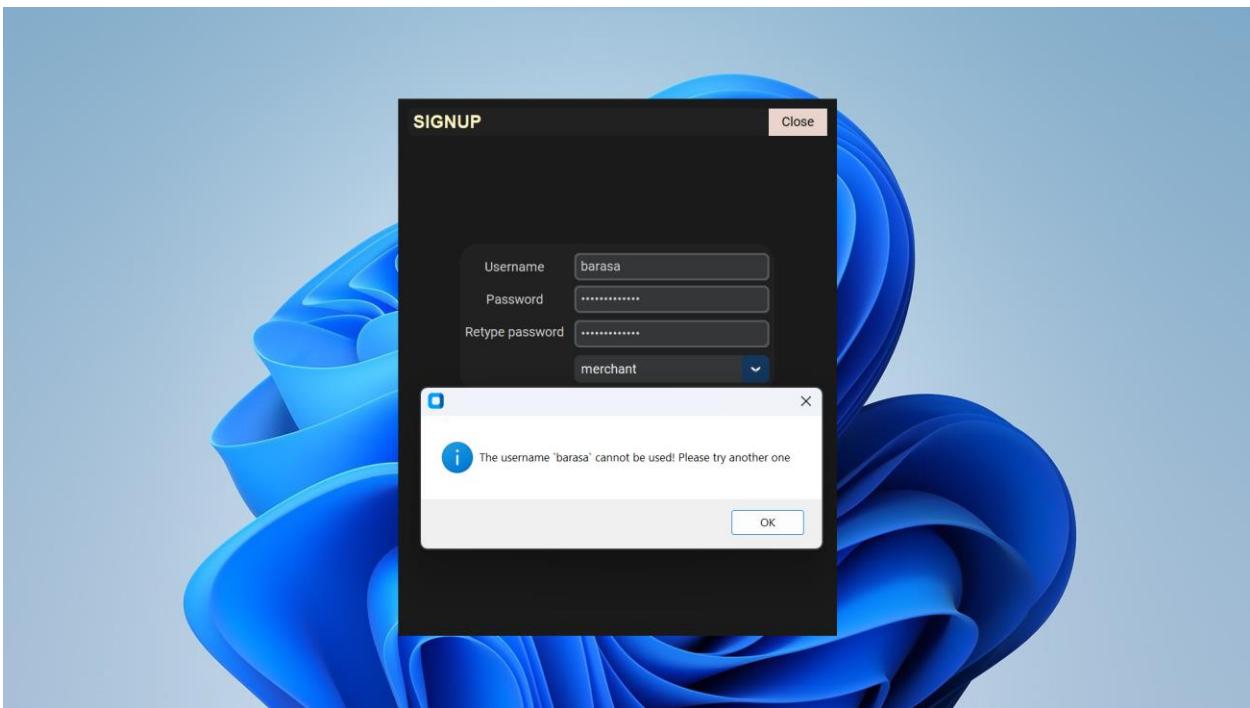


Figure 13 Warning for a username that has already been taken

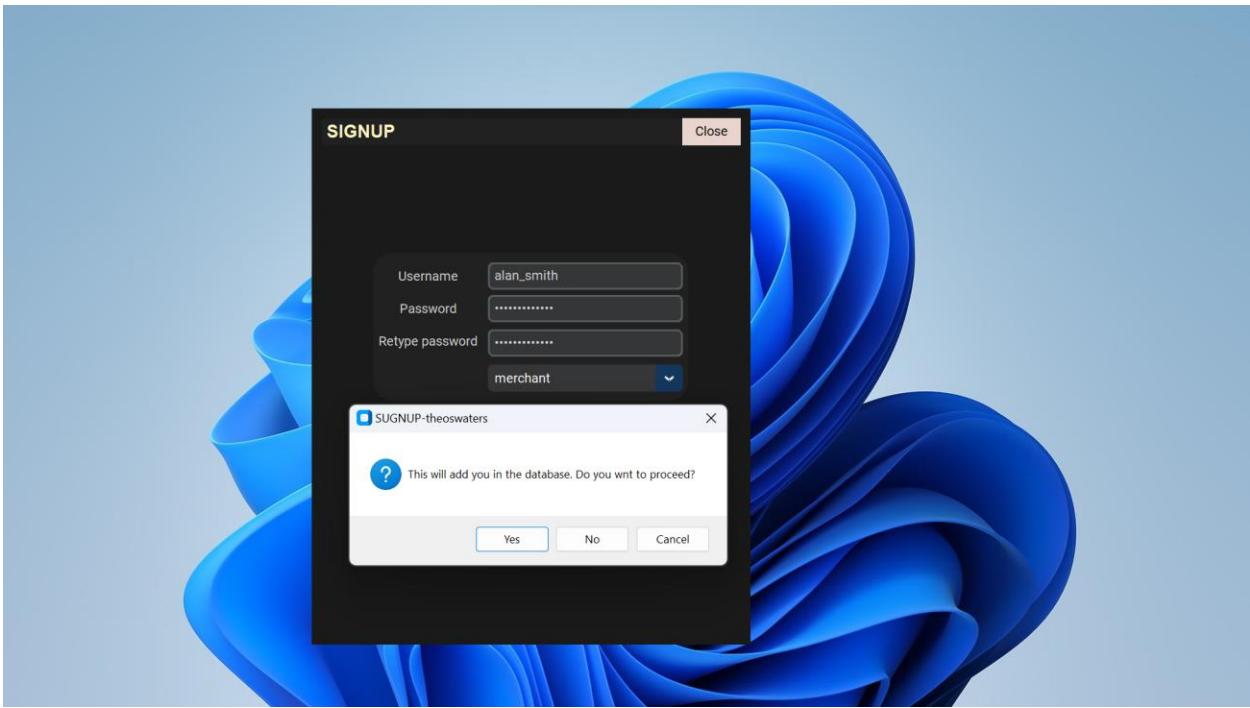


Figure 14 Prompt for assertion

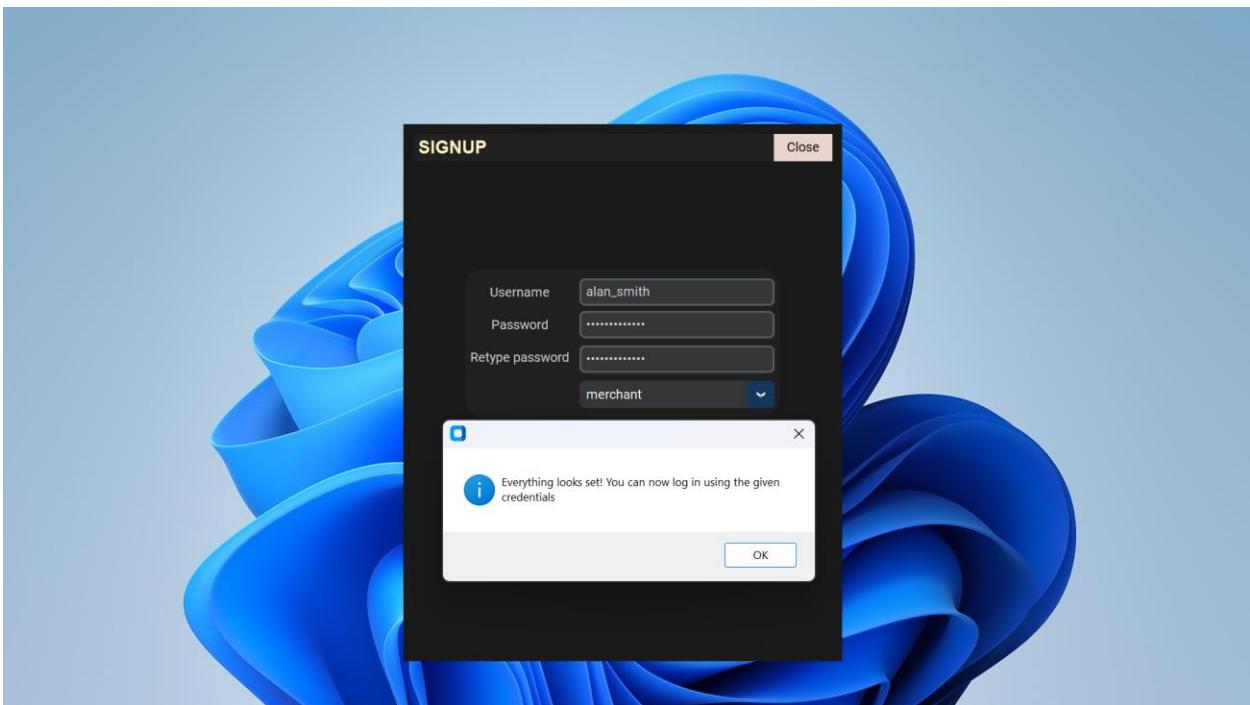


Figure 15 Success message for a successfully created account (merchant)

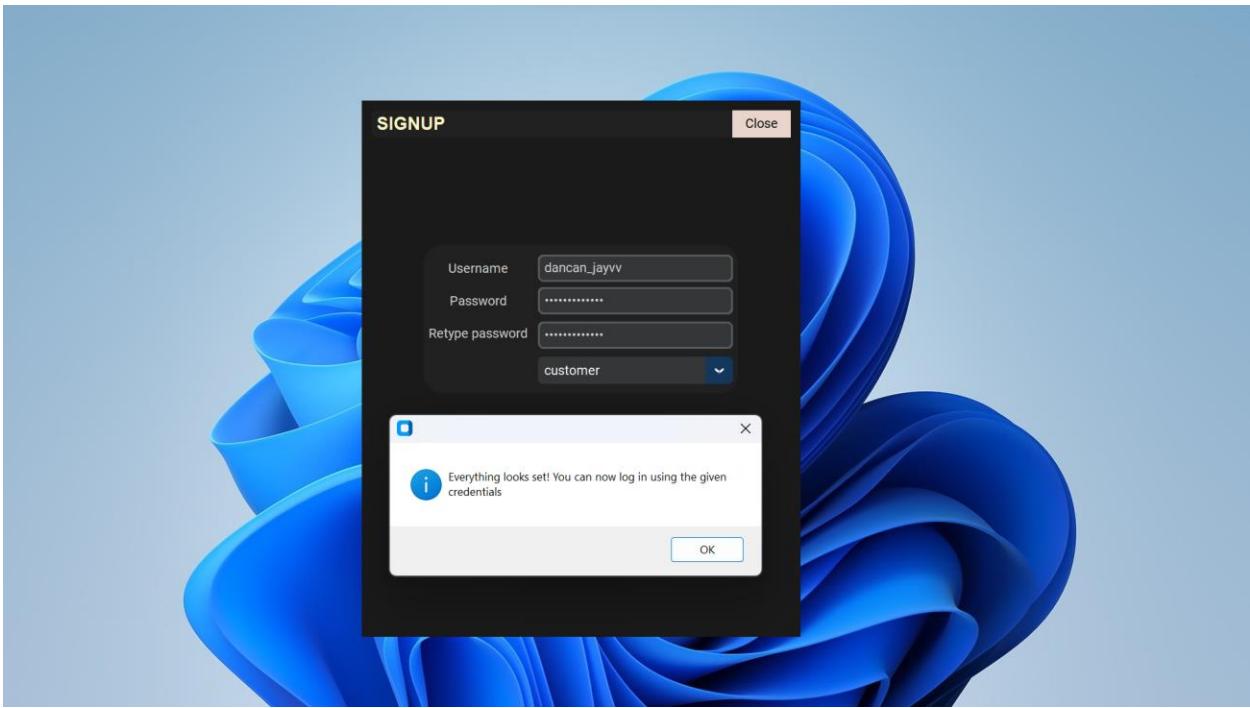


Figure 16 Success message for a successfully created account (customer)

Customer Login

Customers log in by entering their username and password in the login window.

After successful login, customers can place orders by selecting a merchant, choosing a bottle, and validating their bank account balance.

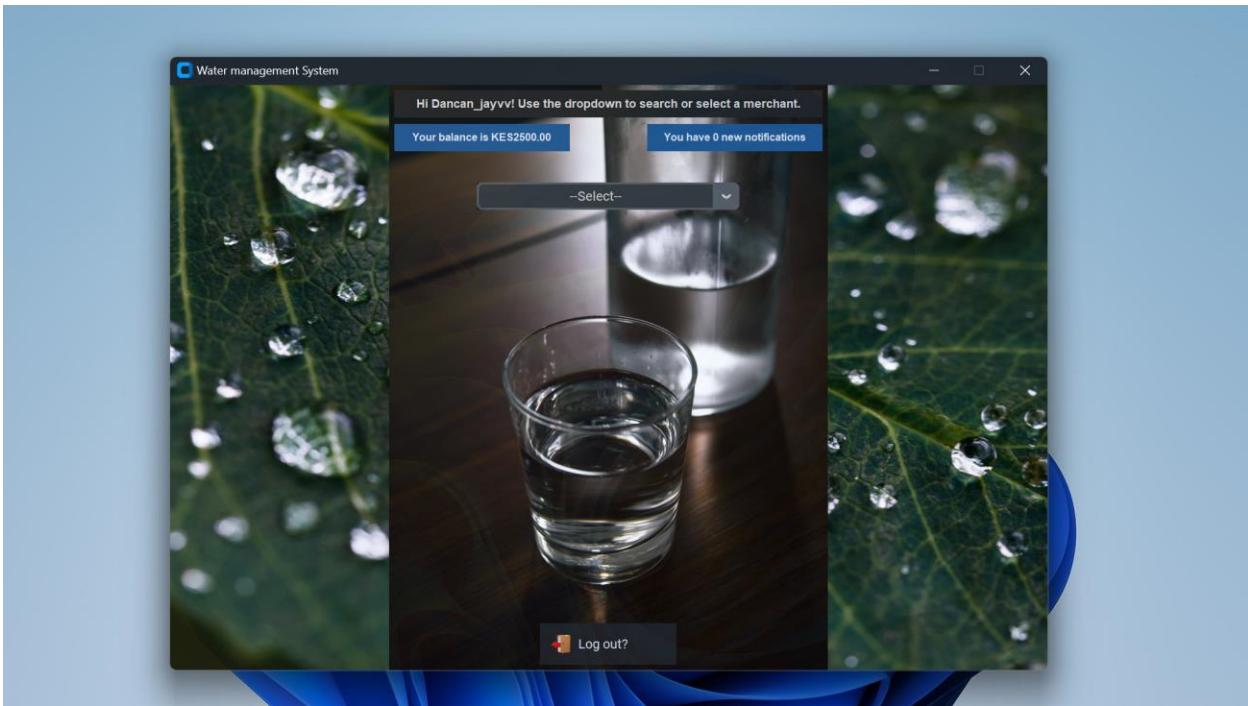


Figure 17 Customer dashboard upon login

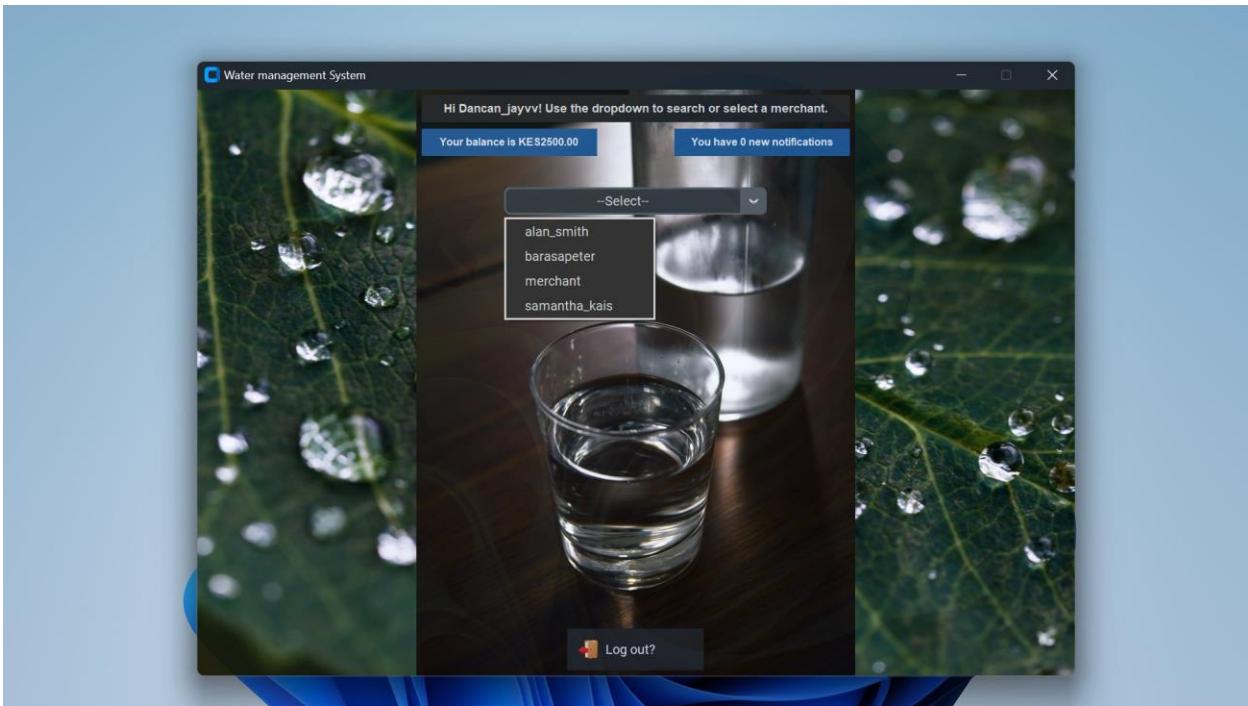


Figure 18 Customer prompt to select a merchant username

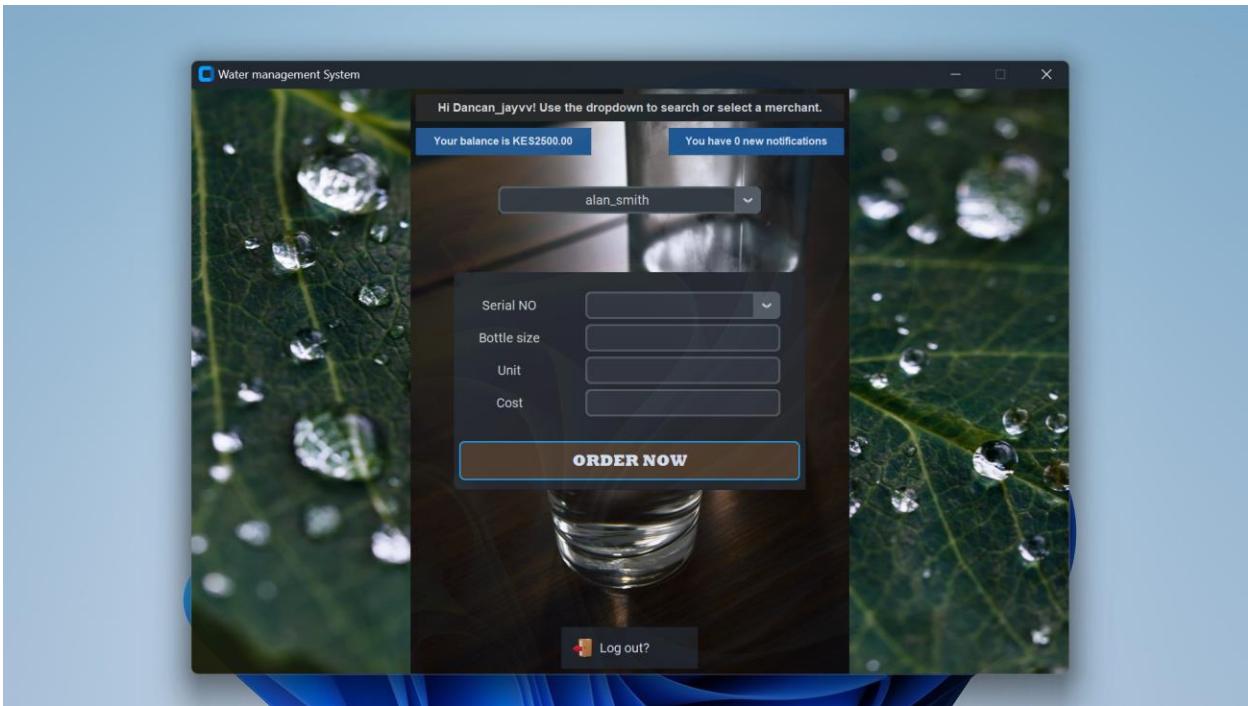


Figure 19 Merchant selected. This merchant does not have any bottles yet

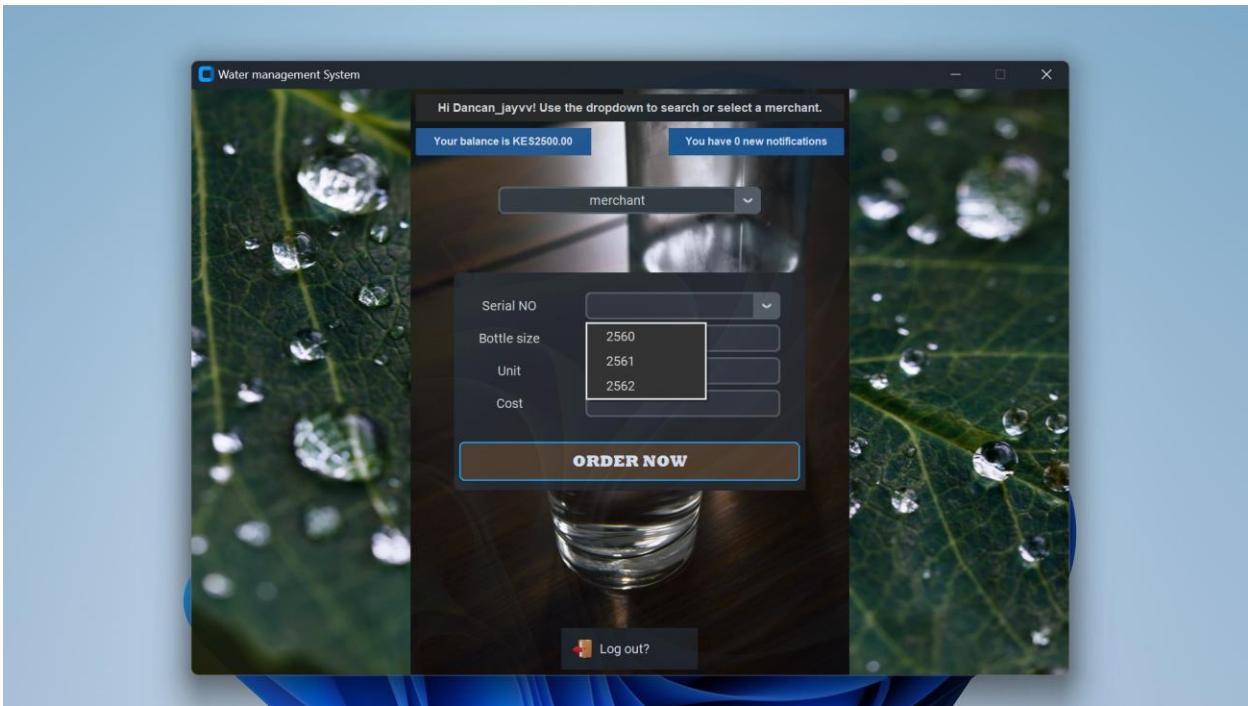


Figure 20 Merchant selected. This merchant has bottles. All the bottles he owns are cascaded on a dropdown list

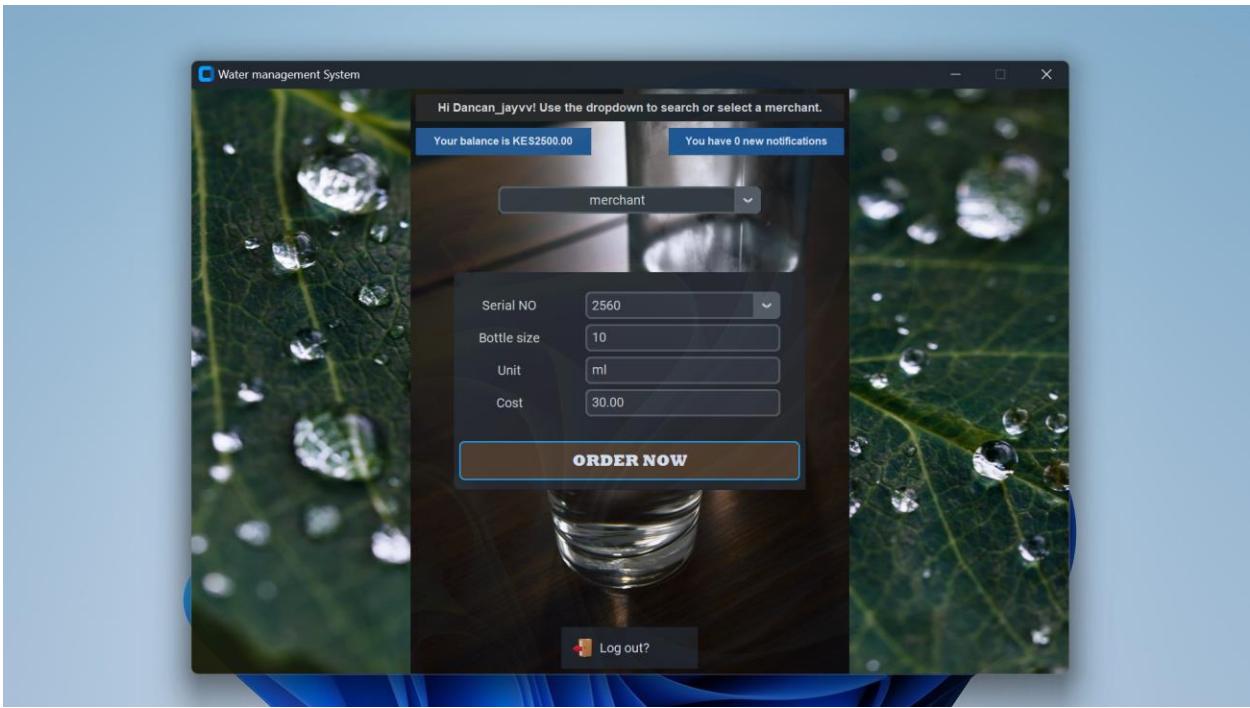


Figure 21 Bottle selected, and all its details are displayed

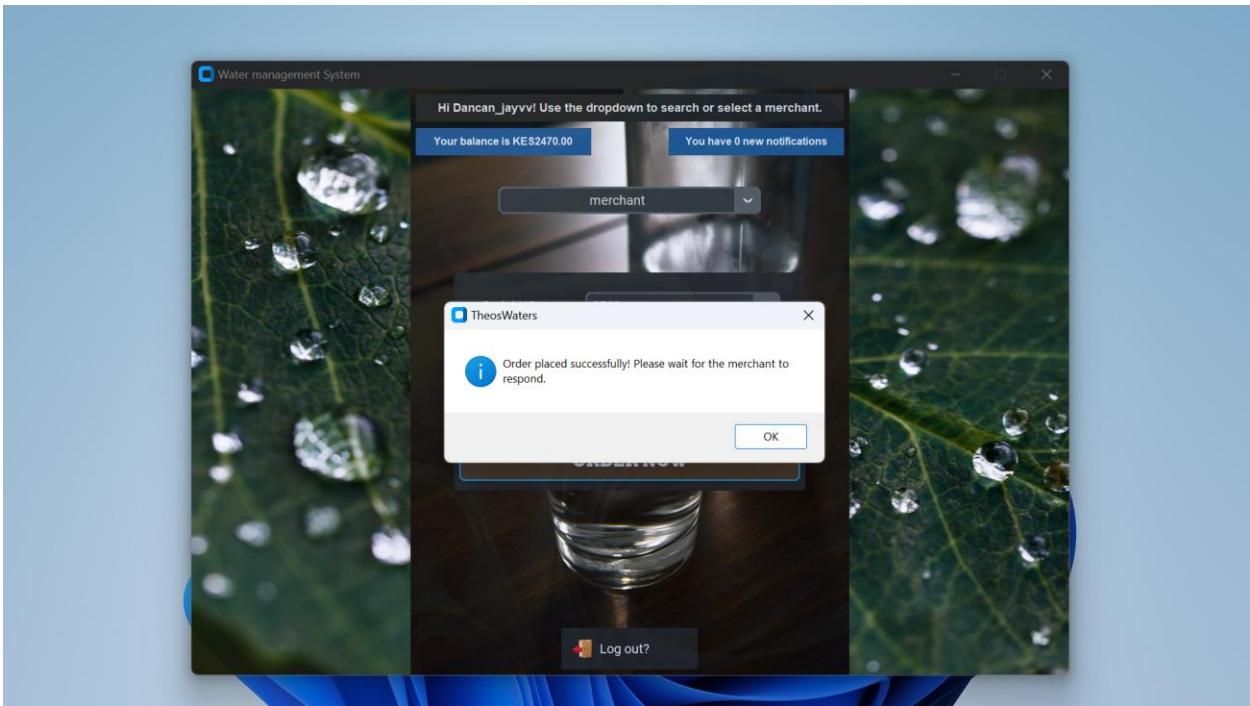


Figure 22 Success message for a successfully placed order

If the validation is successful, an order is sent to the selected merchant for processing.

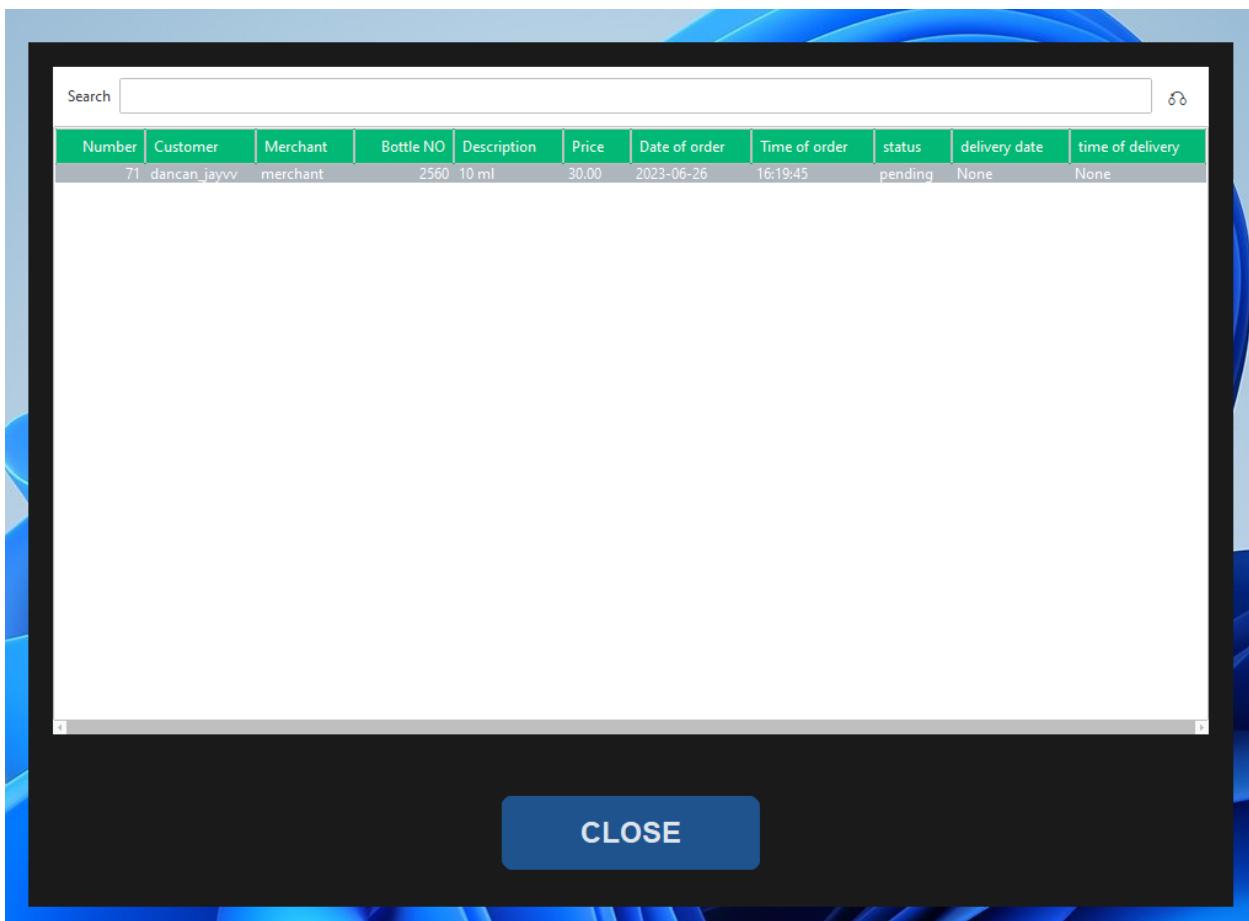


Figure 23 Placed orders display board

Merchant Login

Merchants log in by entering their username and password in the login window.

Upon login, merchants are presented with a dashboard containing various actions and features.

The dashboard includes a sidebar with buttons for Sales History, Orders (with a notification count), Receipts, Dispenser controls, and managing merchant bottles.

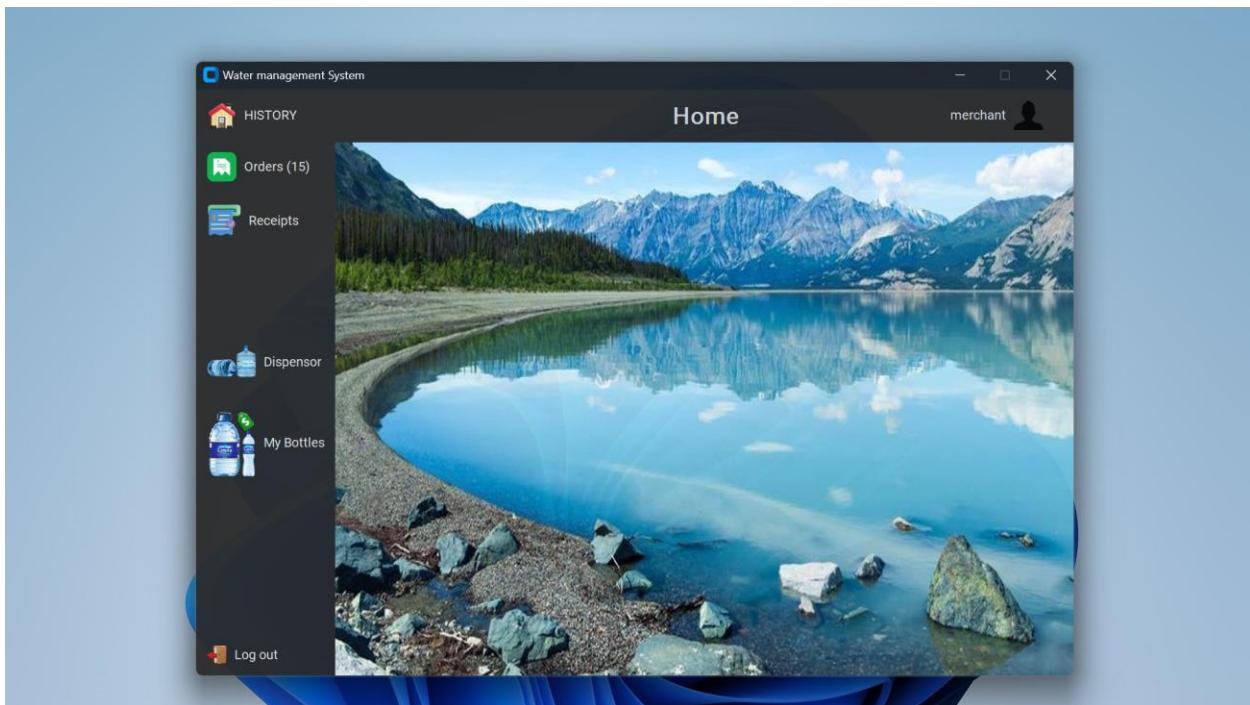


Figure 24 Display screen upon login – merchant

Data Management

Database Initialization

Upon launching the system, a MySQL database is created if it does not already exist.

The system automatically creates two tables: merchants and customers, if they do not exist.

The merchants table consists of two columns: username and password.

The customers table consists of four columns: username, password, account balance, and notifications.

Additionally, an orders table is created with columns storing order details such as order number, customer username, merchant username, bottle ID, description, price, order date, order time, delivery status, delivery date, and delivery time.

Merchant Bottles Table

When a merchant creates an account, a corresponding merchant bottles table is initialized.

This table stores bottle information, including the bottle ID, size, measurement unit, and price/cost.

Sales Table

As merchants make sales, the system records the sales data in a sales table.

This table stores information related to each transaction made by merchants.

Orders Table

The orders table stores information about customer orders, including the order number, customer username, merchant username, bottle ID, description, price, order date, order time, delivery status, delivery date, and delivery time.

Functionality

Customer Orders

Customers can log in, browse available merchants, and select a merchant to place an order.

The system displays the available bottles from the selected merchant, along with their respective prices.

Customers can choose a bottle, and the system validates their bank account balance to ensure they have sufficient funds for the purchase.

If the validation is successful, the customer can proceed to place the order, which is then sent to the selected merchant for further processing.

Merchant Actions

Tabs

Merchants can log in, view their sales history, and access various functionalities from the merchant dashboard.

The dispenser control allows merchants to initiate water dispensing and record relevant data.

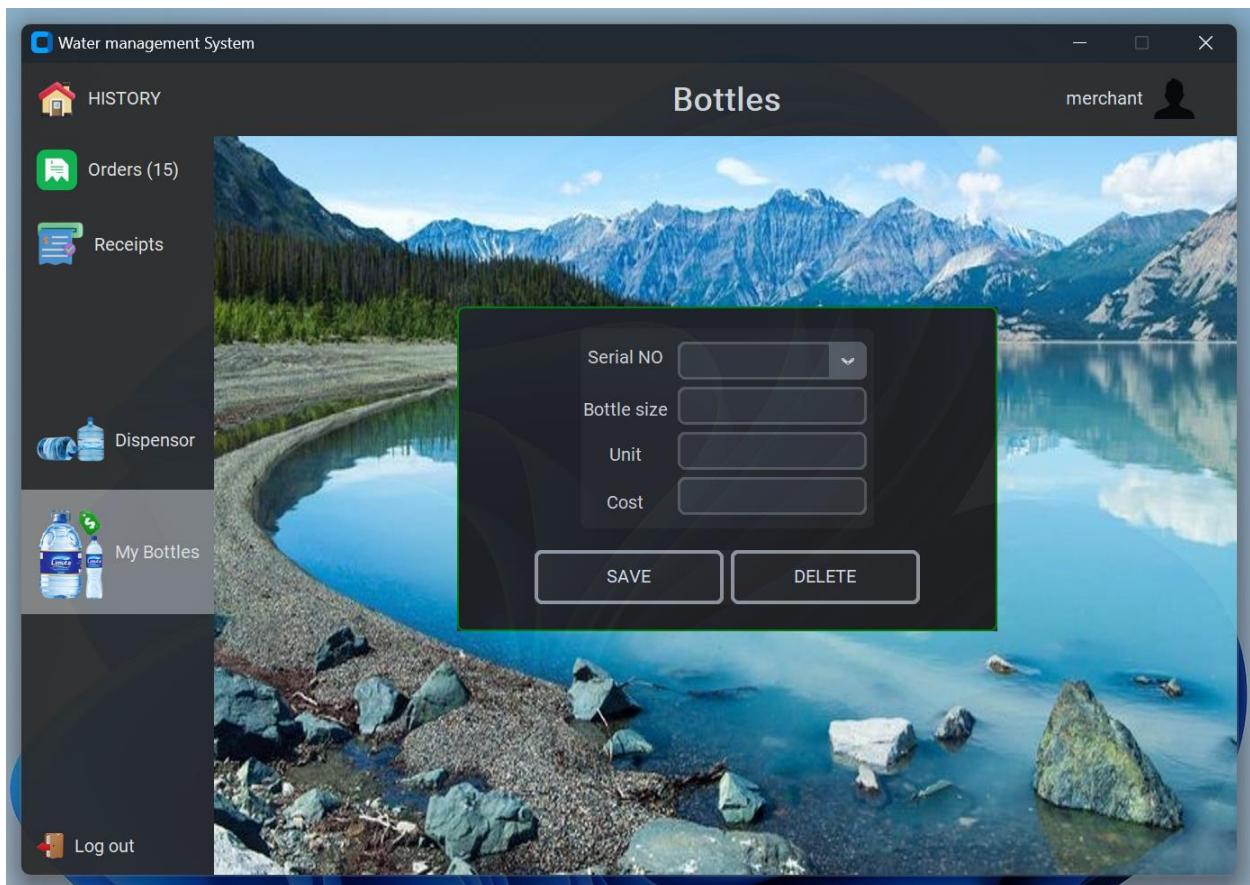


Figure 25 Bottles tab (Enables the user to add bottles, delete or edit existing ones)

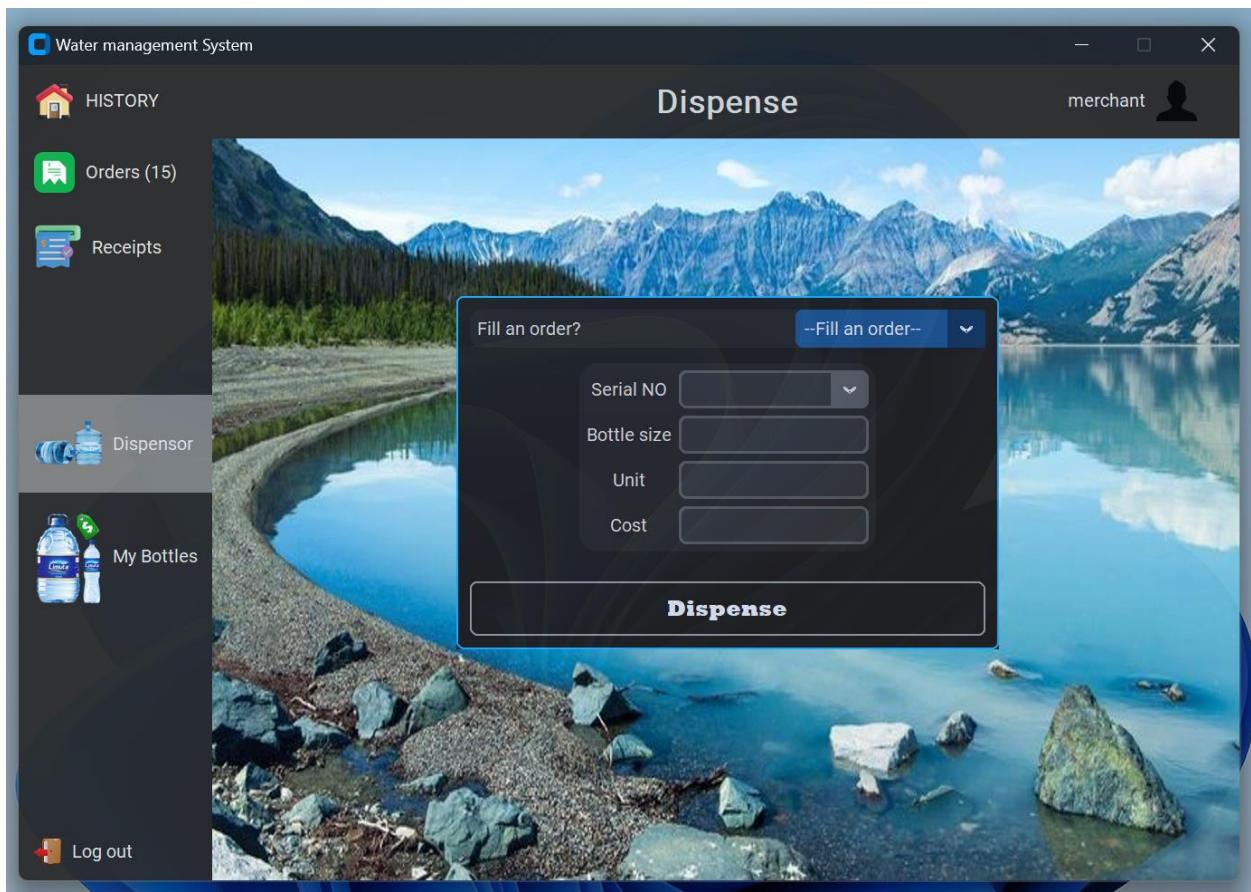


Figure 26 Dispenser interphase

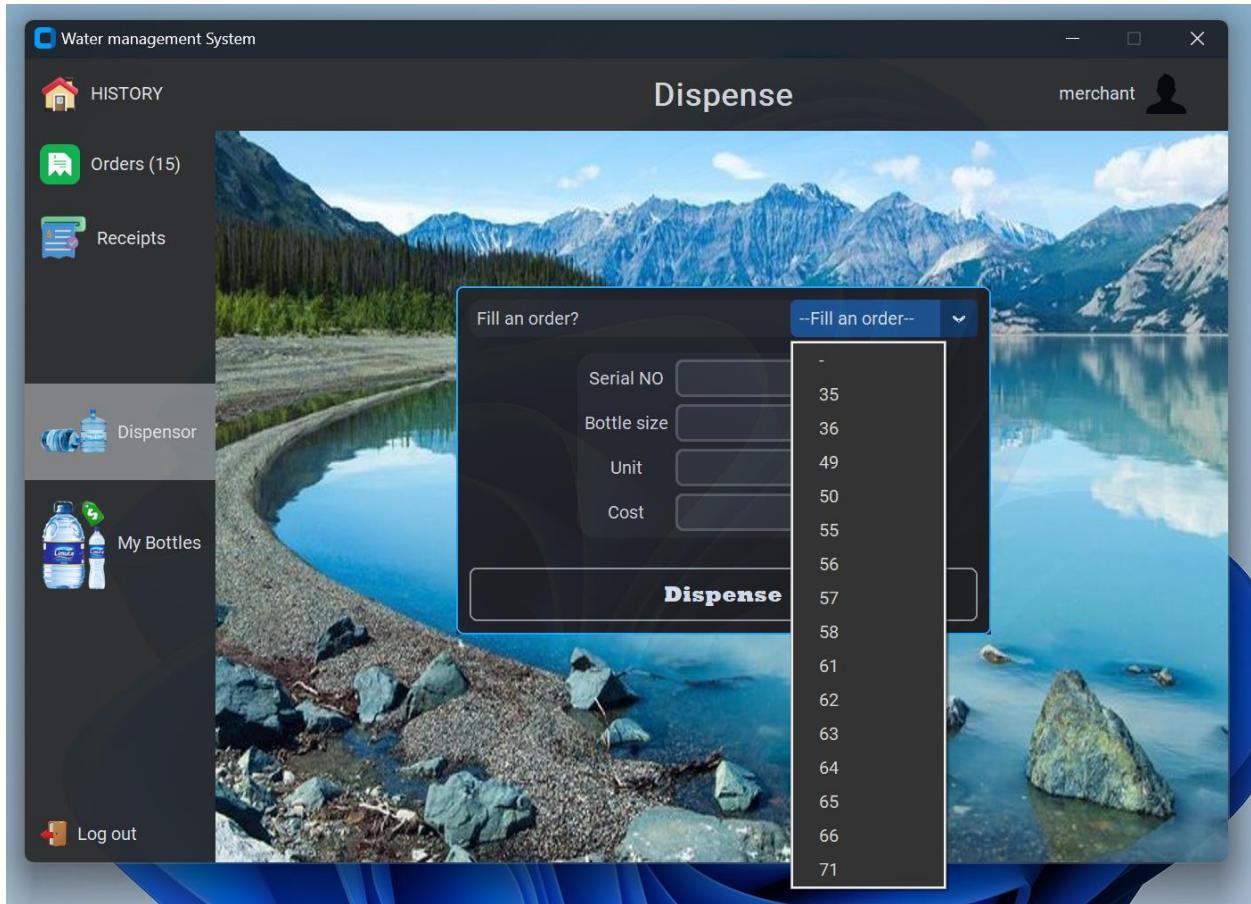


Figure 27 List of all orders from customers

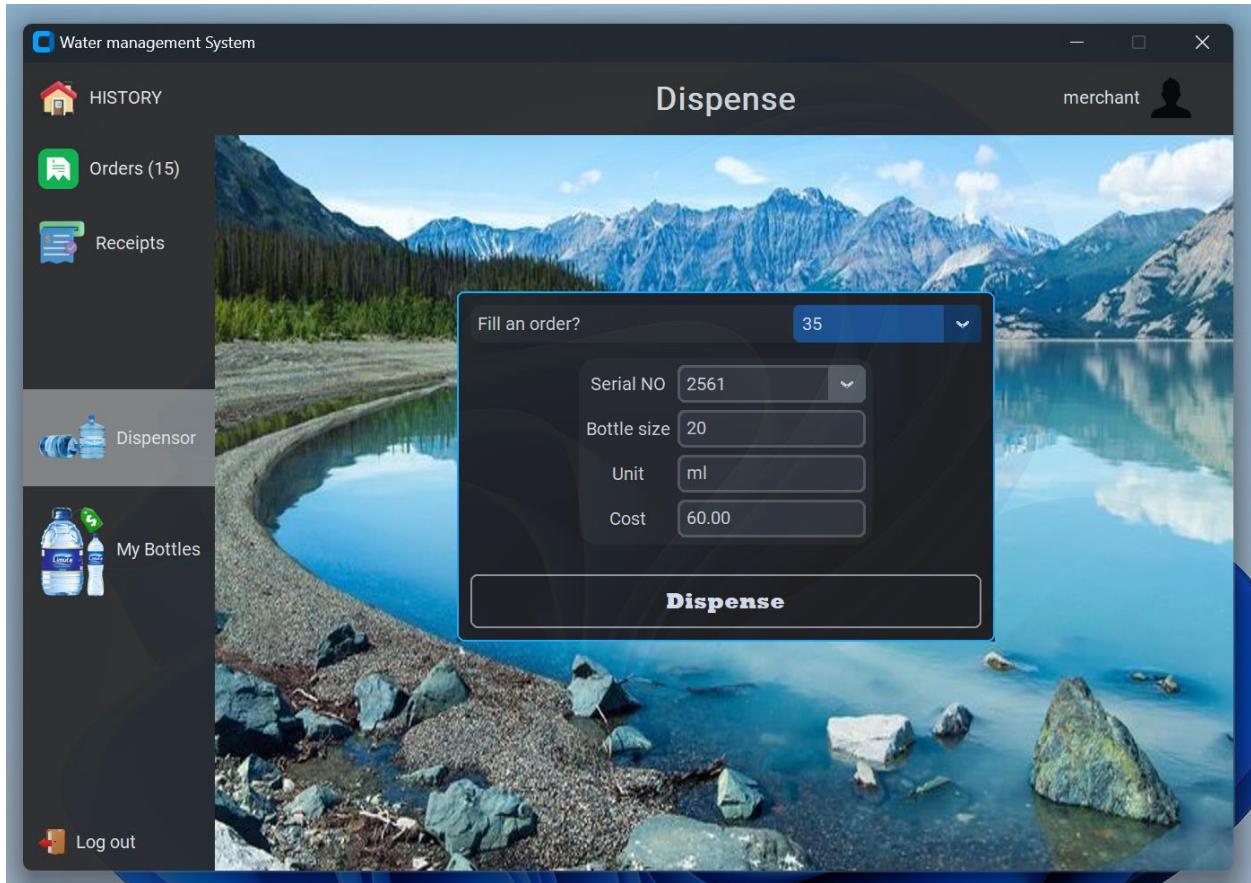


Figure 28 Sample order ID selected. The details about the order are automatically displayed! Isn't this awesome?

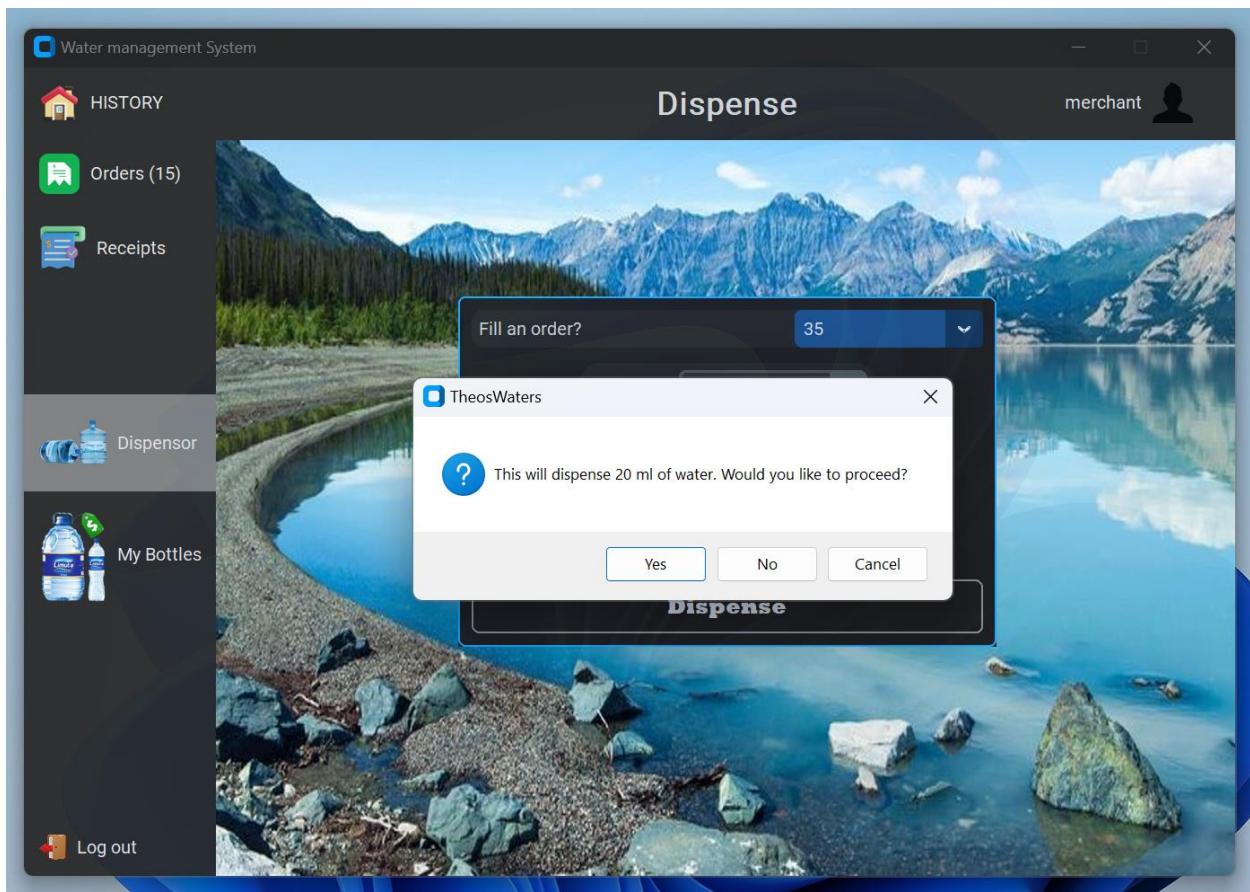


Figure 29 Confirm the action to proceed

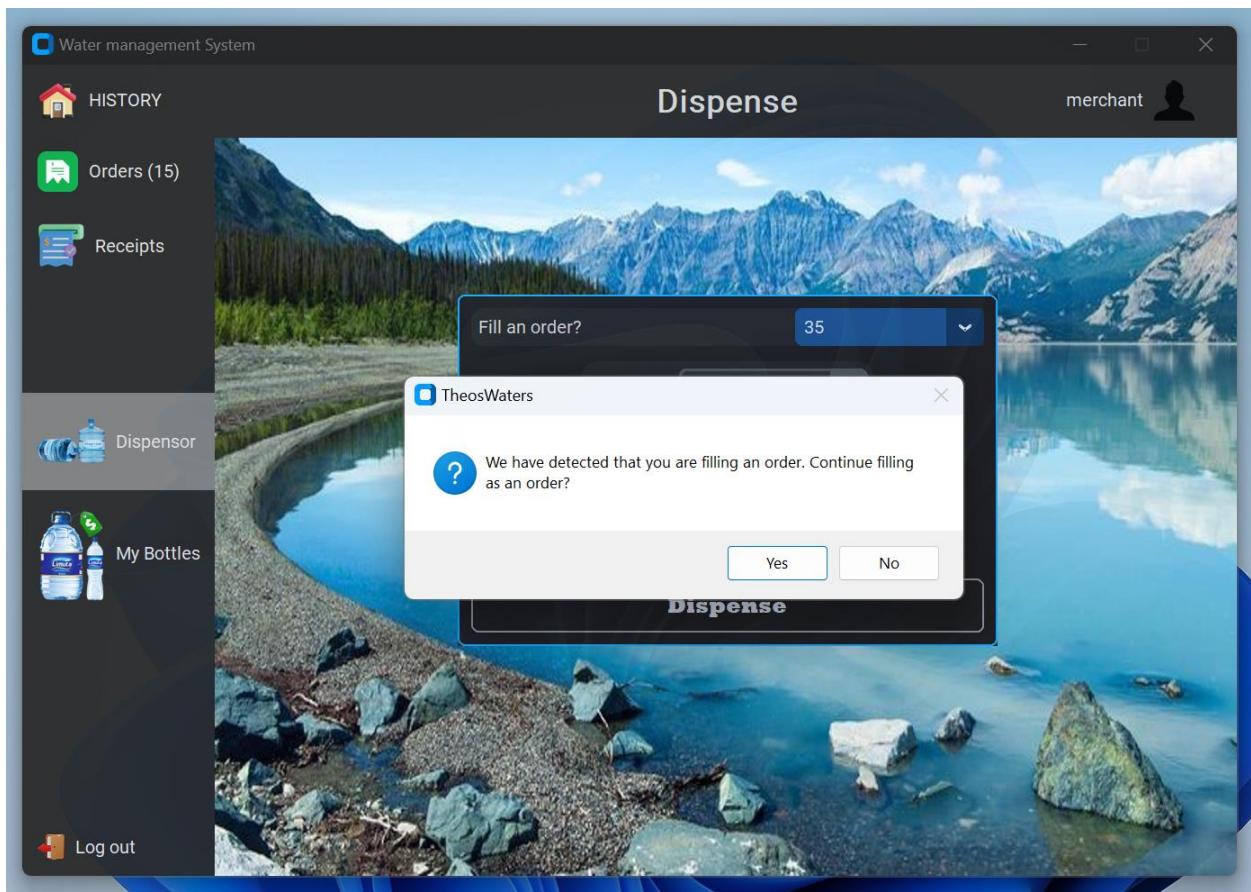


Figure 30 System detects an order fill. Prompt before dispense as "Order"

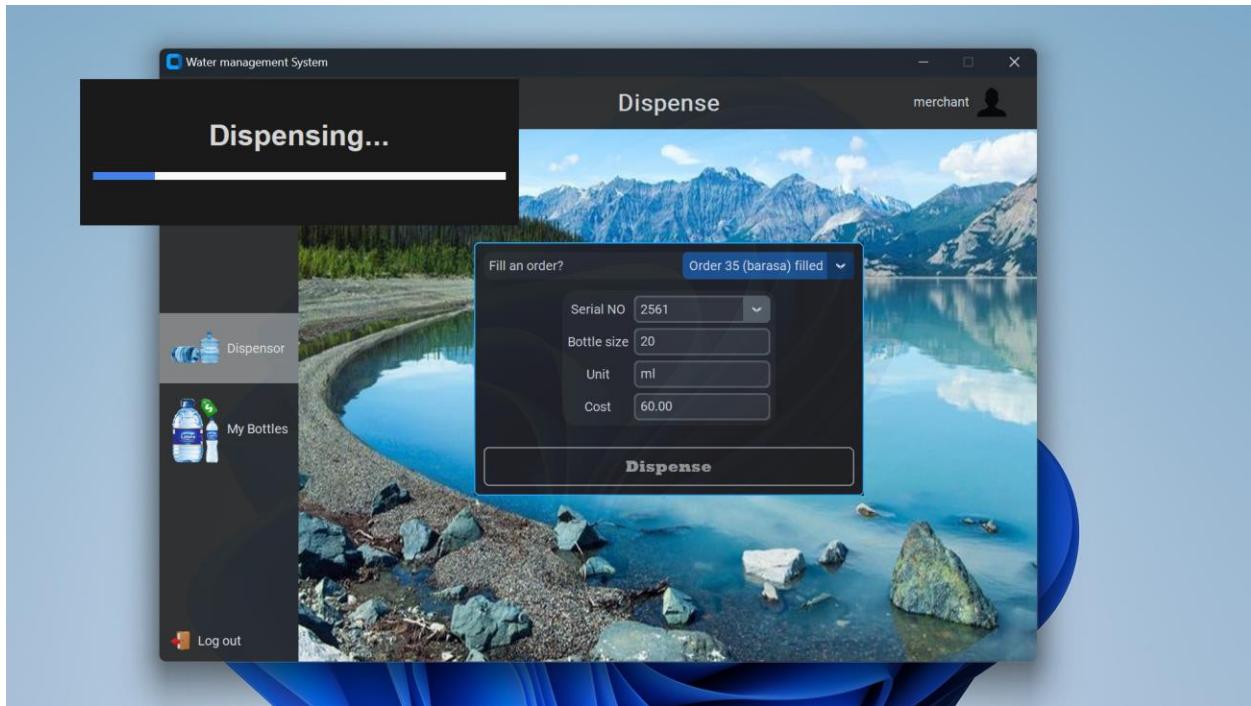


Figure 31 Dispense progress shown

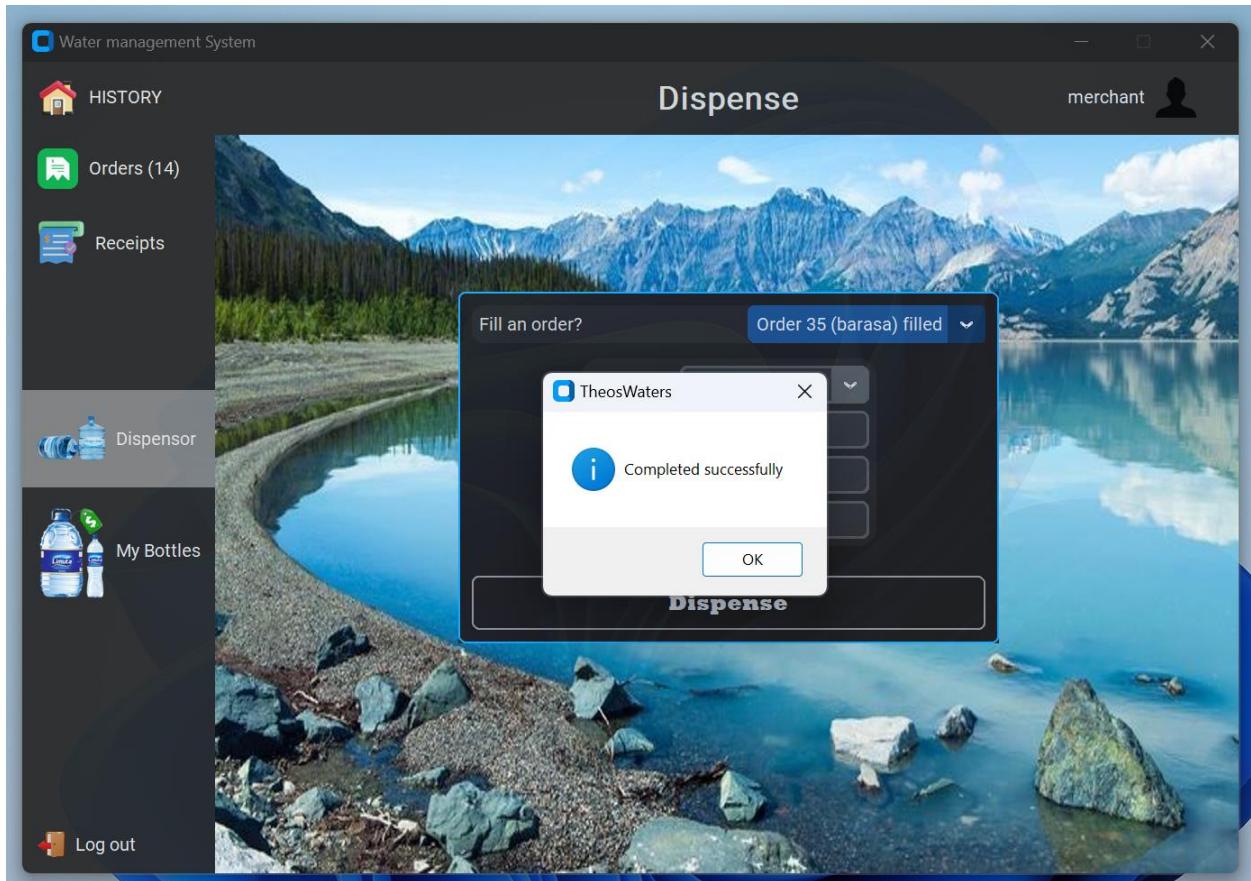


Figure 32 Confirmation message

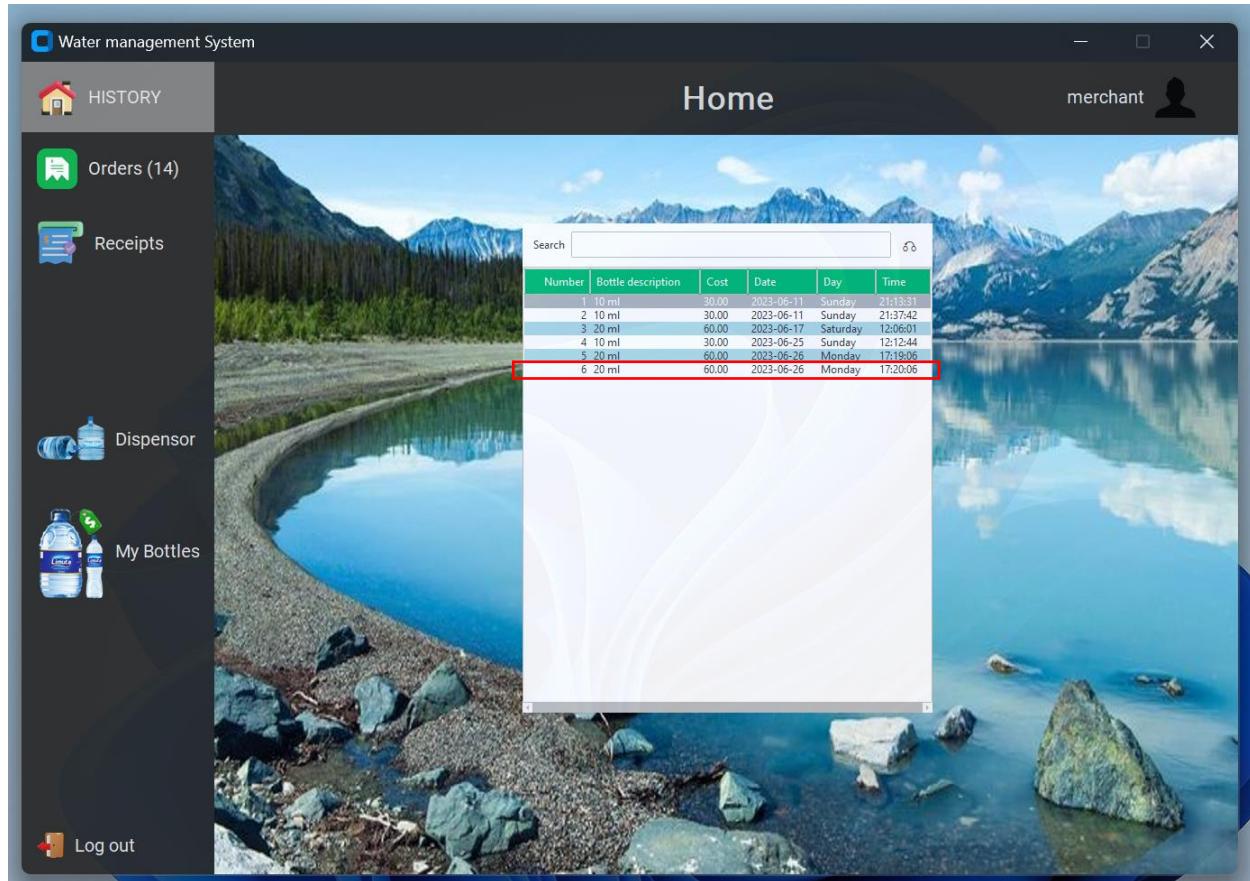


Figure 33 Action recorded in general transaction history

Orders

Number	Customer	Merchant	Bottle NO	Description	Price	Date of order	Time of order	status	delivery date	time of delivery
2	barasa	merchant	2560	10 ml	30.00	2023-06-11	21:37:17	delivered	2023-06-11	21:37:44
34	barasa	merchant	2561	20 ml	60.00	2023-06-12	23:50:32	delivered	2023-06-17	12:06:10
35	barasa	merchant	2561	20 ml	60.00	2023-06-12	23:50:37	delivered	2023-06-26	17:22:30
36	barasa	merchant	2561	20 ml	60.00	2023-06-12	23:50:40	pending	None	None
44	barasa	merchant	2560	10 ml	30.00	2023-06-17	12:05:14	delivered	2023-06-25	12:12:47
49	ben	merchant	2561	20 ml	60.00	2023-06-17	12:10:19	pending	None	None
50	ben	merchant	2562	50 ml	10.00	2023-06-17	12:10:24	pending	None	None
55	barasa	merchant	2560	10 ml	30.00	2023-06-22	16:56:45	pending	None	None
56	barasa	merchant	2561	20 ml	60.00	2023-06-22	16:56:57	pending	None	None
57	barasa	merchant	2562	50 ml	10.00	2023-06-22	16:57:15	pending	None	None
58	barasa	merchant	2562	50 ml	10.00	2023-06-22	16:57:24	pending	None	None
61	barasa	merchant	2560	10 ml	30.00	2023-06-23	10:46:15	pending	None	None
62	barasa	merchant	2561	20 ml	60.00	2023-06-23	10:46:22	pending	None	None
63	barasa	merchant	2562	50 ml	10.00	2023-06-23	10:46:39	pending	None	None
64	barasa	merchant	2561	20 ml	60.00	2023-06-23	10:50:24	pending	None	None
65	barasa	merchant	2561	20 ml	60.00	2023-06-23	10:50:36	pending	None	None
66	barasa	merchant	2561	20 ml	60.00	2023-06-23	10:50:42	pending	None	None
71	dancan_jayvv	merchant	2560	10 ml	30.00	2023-06-26	16:19:45	pending	None	None

Figure 34 Action recorded in orders history

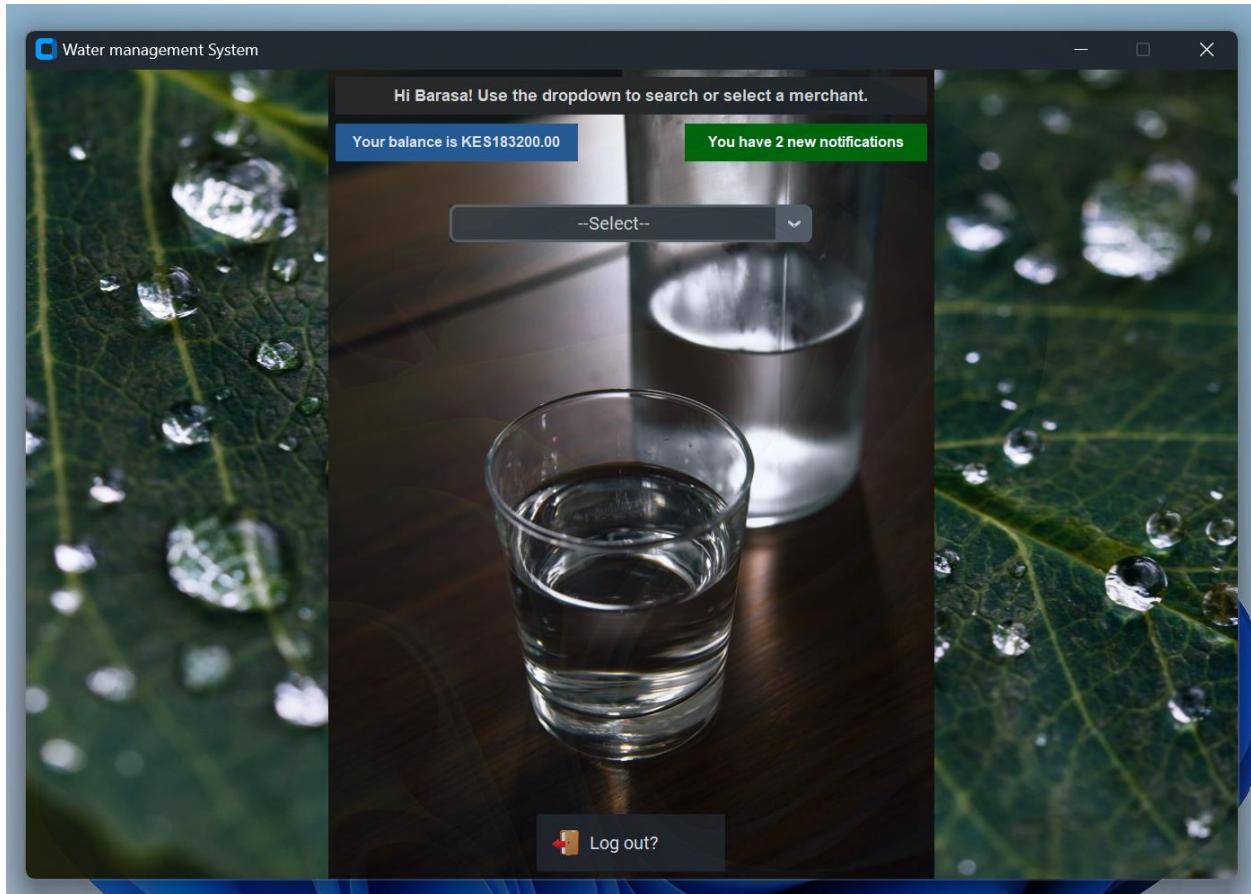


Figure 35 The customer receives a notification that his previous order has been filled!

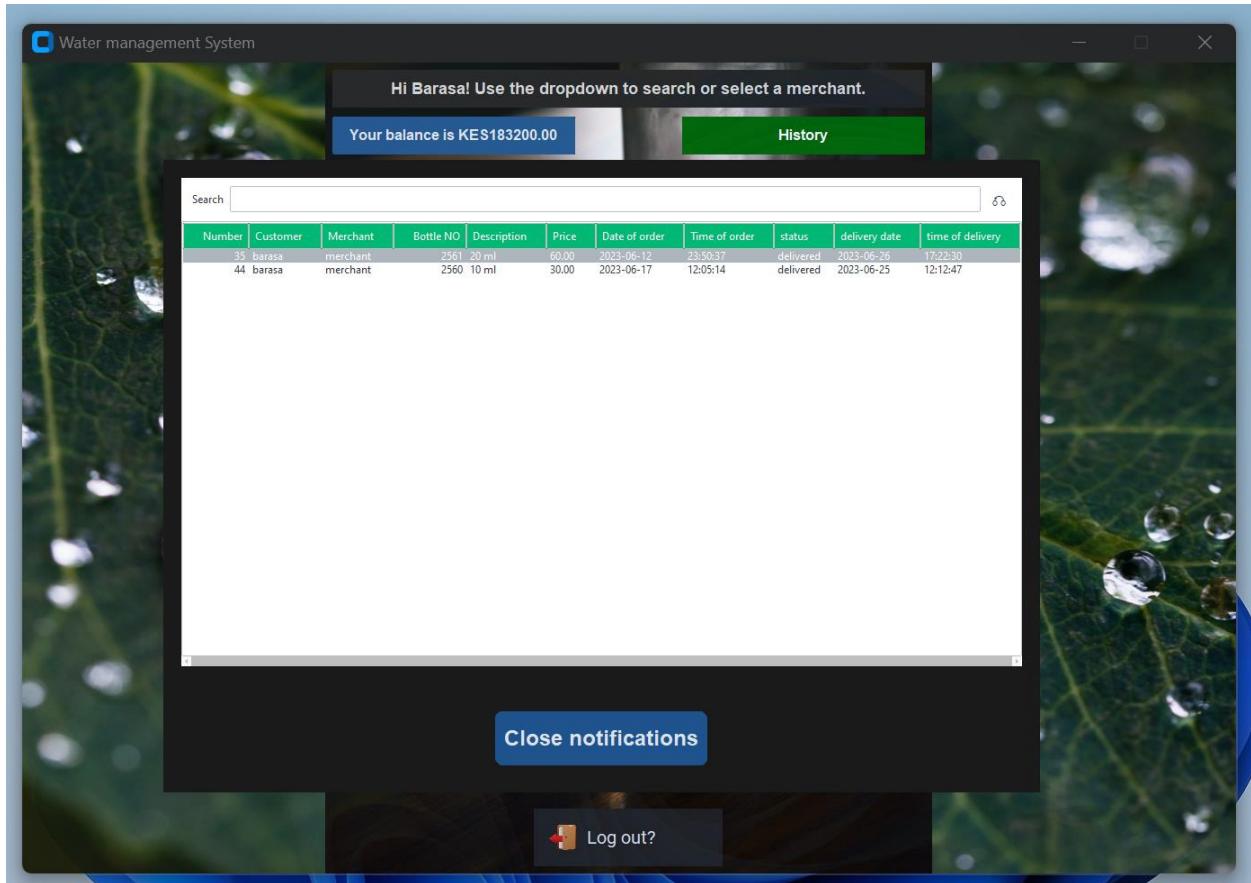


Figure 36 Notification showing delivered orders

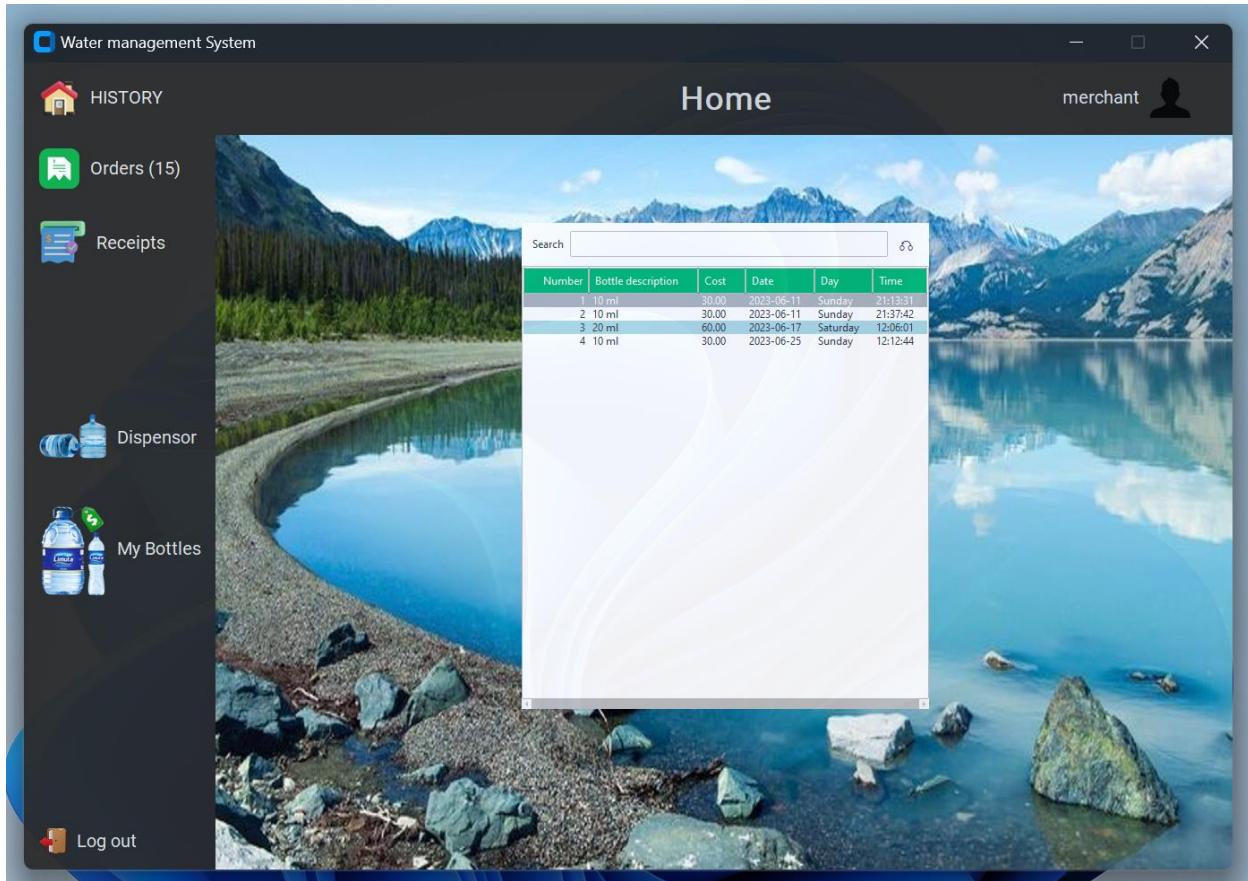


Figure 37 Cumulative sales history tab

The sales history provides an overview of past transactions, including order details, dates, and amounts.

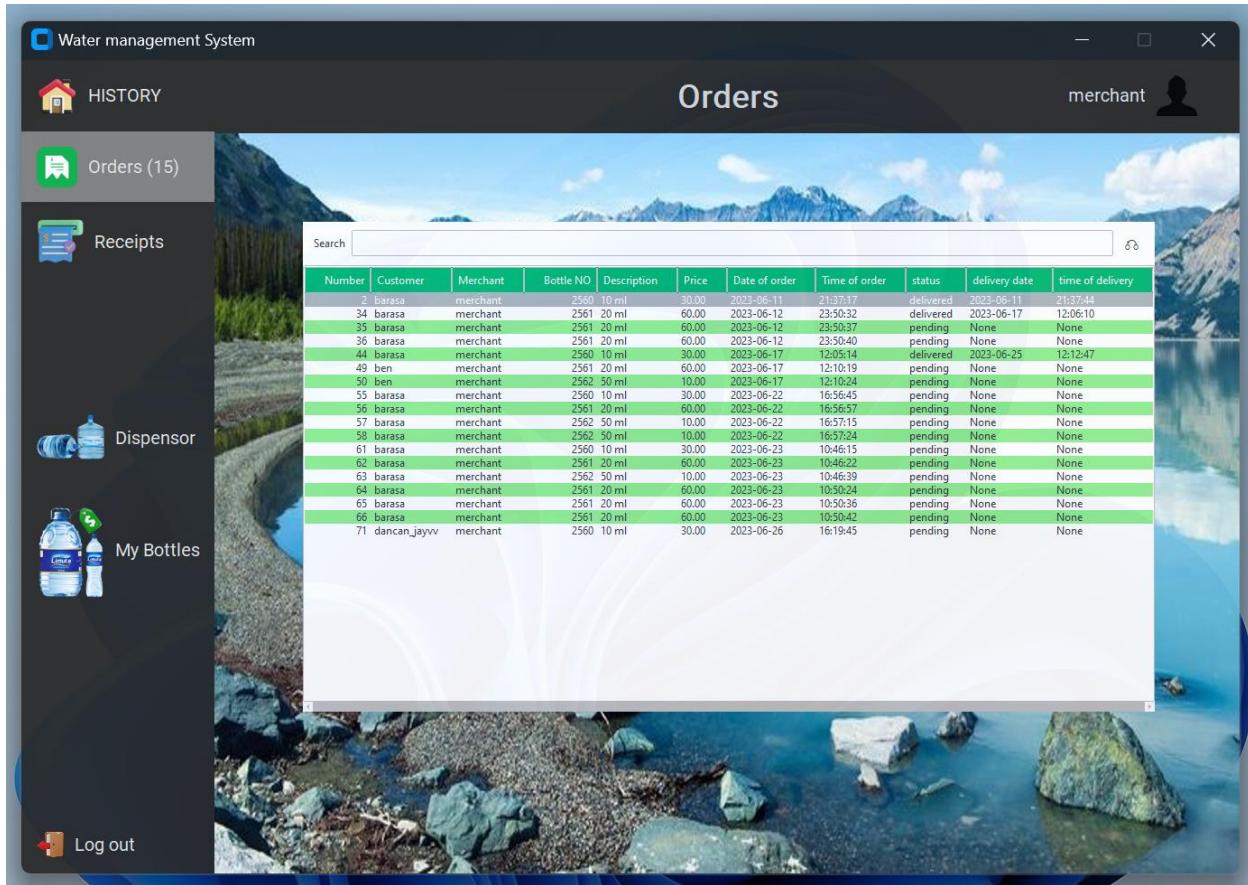


Figure 38 Orders transaction history

The dashboard also displays the number of new orders with a notification count.

MERCHANTS can download receipts for each sale, enabling them to maintain records and generate reports.

The user has two download options: They can download all receipts all at once or download the latest receipt. This is made possible by two buttons in the tab. (They are beautiful, aren't they?)

The “Download all receipts” button allows the user to select a directory to download all receipts in his database. The other button does the exact same thing – except it only downloads one receipt (the latest receipt).

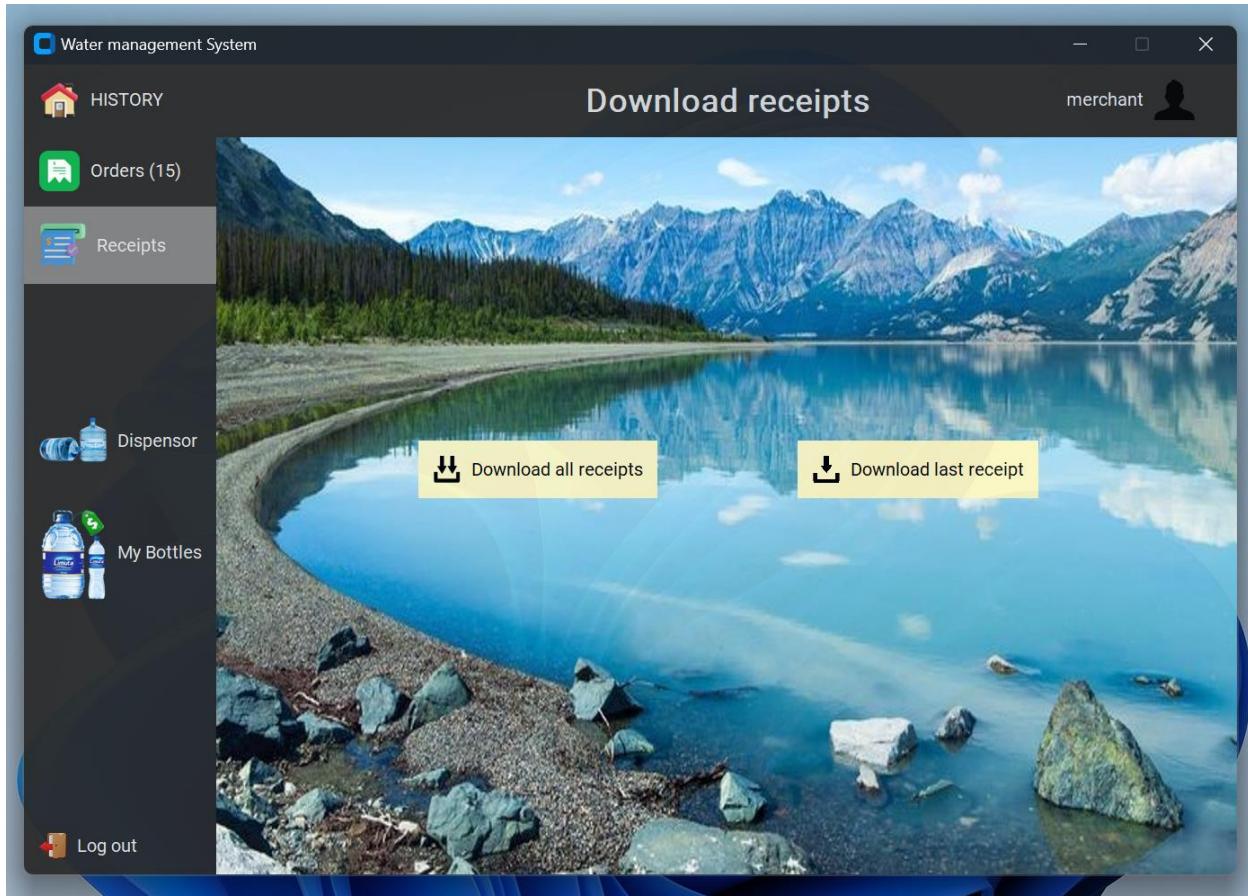


Figure 39 Download receipts tab

Merchants can manage their merchant bottles, adding or deleting bottles as necessary.

Troubleshooting

Common Issues

If users encounter any issues while using the system, such as login failures, incorrect balances, or missing data, they can refer to the following suggestions for troubleshooting:

Verify that the entered username and password are correct.

Ensure the bank account balance is accurate and up-to-date.

Check the network connection and ensure the MySQL server is running.

If the issue persists, contact the system administrator for further assistance.

Limitations

Scalability and concurrency

The current version of the system is designed for small to medium-scale usage. Scaling the system for larger deployments may require additional considerations, such as optimizing database performance and handling increased user loads.

The database has also been designed to handle one request at a time. It cannot, therefore handle more than one user request at a time.

Hardware Requirements:

The water management system has no specific hardware requirements and can run on standard computer systems with Python, Tkinter, and MySQL installed.

Future Improvements:

In future versions of the system, potential improvements can include enhancing user interfaces, optimizing database queries, incorporating automated notifications, and introducing more advanced features to streamline the water management process.

References

[Python – The Python software foundation](#)

[Tkinter - The Python software foundation](#)

[MySQL](#)

[Customtkinter](#)

[TTKbootstrap](#)

APPENDICES

Appendix A: Sample Reports

This appendix provides sample reports generated by the water management system for reference.

Customer Order Report

The customer order report provides details of all orders placed by a customer within a specified date range.

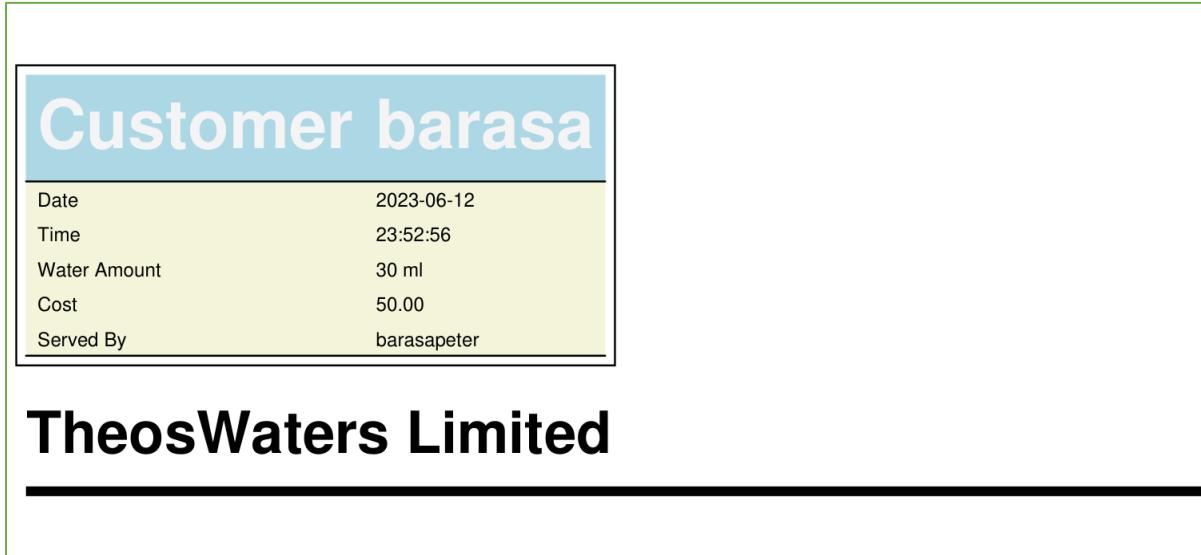


Figure 40 Customer's completed order receipt

Key fields:

- Order ID
- Order Date
- Merchant Name
- Bottle Details (Size, Quantity, Price)
- Total Amount
- Order Status

Merchant Sales Report

Copy of customer's sales receipt is also retained for the merchant.

The merchant sales report provides aggregated sales data for a merchant within a specified period.

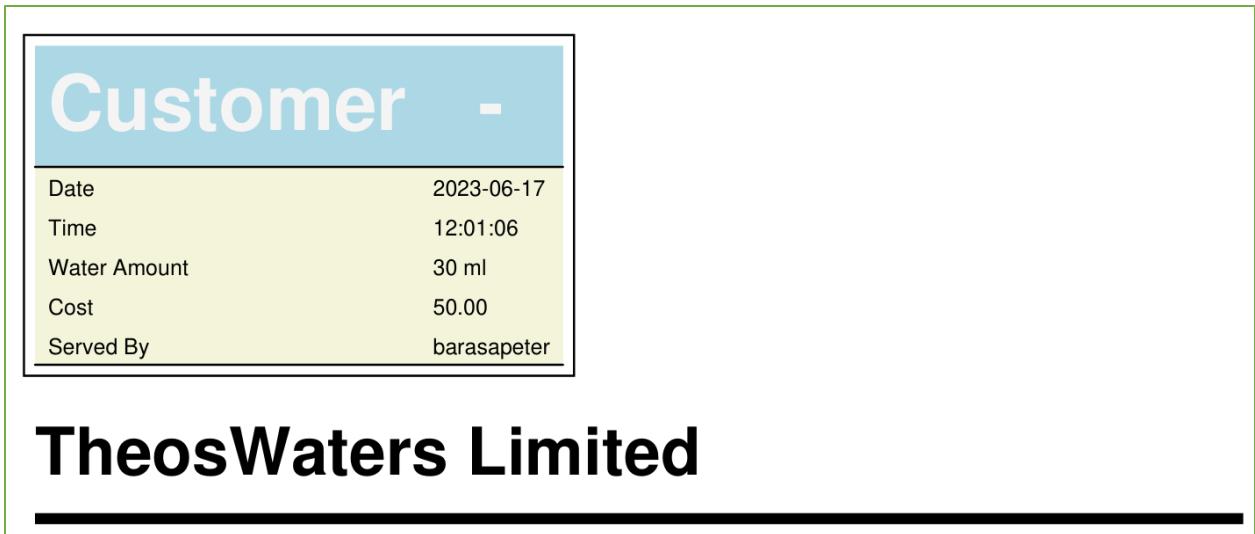


Figure 41 Receipt exclusive to merchant. The customer is not defined.

Key fields:

- Date
- Total Orders
- Quantity Sold
- Gross Sales
- Top Selling Products

Inventory Audit Report

The inventory audit report provides a snapshot of a merchant's bottle inventory levels.

Key fields:

- Bottle Type
- Opening Inventory
- Units Sold
- Units Added
- Closing Inventory

These sample reports demonstrate the business insights and data analytics available to customers, merchants and administrators via custom reports in the system. Additional reports can be designed to address specific reporting requirements.