Development of a Software System for Controlling Dosing Machines

Gergana Spasova *, Milena Karova 💻 and Ivaylo Penev

Faculty of Computer Sciences and Automation, Technical University of Varna, 9010 Varna, Bulgaria; mkarova@tu-varna.bg (M.K.); ivailo.penev@tu-varna.bg (I.P)
* Correspondence: g.spasova@tu-varna.bg
† Presented at the International Conference on Electronics, Engineering Physics and Earth Science (EEPES’24), Kavala, Greece, 19–21 June 2024.

Abstract: The liquid dosing system consists of machines and a computer (PC). The machines and the computer are connected to each other and communicate via the RS485 interface. The software installed on a computer (PC) consists of two separate modules (applications)—“server” and “operator”. The server runs continuously and communicates to and from the machines. Through the operator module, the nomenclature of customers, liquids, and cards can be created and edited. The operator module makes it possible to perform various operations such as loading credit on a card, blocking and unblocking cards, editing customer or card data, and others. Various reports can be made, such as a report on the consumption of a given customer, a report on the use of the customer card, a report on the liquid consumed, etc. The goal of the system is to perform automated dosing of liquid at the customer’s request, without the involvement of another person (operator). The choice of liquid, the amount, and the payment of the choice made is fully automatic. Payment is only by card, as in case of an attempt to make a dosage and insufficient availability for payment, the dosage is not carried out. The customer is informed of a small credit on the card. The connection between the machine and the operator display is based on a query from the display. The operator’s job involves replacing empty bottles in the machine and monitoring and following up on system irregularities.

Keywords: communication protocol; vending machines; binary protocol; liquid dispensing

1. Introduction

Dosing systems are increasingly used nowadays. Paper [1] describes the development of an FPGA-based water vending machine that efficiently manages the process of dispensing water and returning change in real-time. The vending machine accepts three inputs: the desired number of liters, the type of water (cold, hot, or normal), and the amount of money entered by the user. The results of the system include the supplied water and the correct change, which is returned to the user in real-time.

In [2], the design of an intelligent snack vending machine is described, and in [3], an FSM vending machine for dispensing products of different types and prices, verifying behavioral specifications and logic operation.

A vending machine for dosing and selling drinking water has been developed [4]. The system can check five parameters—turbidity, conductivity, total dissolved solids (TDS), pH, and water temperature. The system is designed in the automation system from a controller including GSM module 3G Shield (UC20-G) when sending data to Firebase to create a real-time database.

In [5], the efficiency and cost of vending machine development are investigated, and in [6], the design of an automatic chocolate vending machine is proposed. Radiofrequency identification (RFID) and Arduino are used. In [7], an algorithm for tracking the status of products in a vending machine is described.
Dosing machines are also used in medicine. An article [8] describes the dosing of an expensive drug to prevent blood clotting. Maintenance doses are monitored by machine learning algorithms.

In [9], a model is described that shows the availability and total amount of selected products from the vending machine. Here, FPGA [10] is used to design the proposed vending machine (VM), due to its flexibility. The proposed vending machine is mainly used to sell logic doors.

In [11], the characteristics of the use of recommendation algorithms for vending machines are described.

Article [12] presents a solution for an automatic dosing system for ophthalmic fluids. The developed eye drop device delivers drops of less than 10 μL, meanwhile, it can electronically monitor, record, and store the patient’s daily treatment at home.

In the present development, the liquid vending and dispensing machine can automatically dispense in different quantities. The main function of the designed machine is the storage of liquids. For example, an opened bottle of wine can be stored for up to 28 days after opening without affecting the taste of the wine. This is achieved by introducing gas under pressure in place of the consumed amount of wine, which prevents the wine from oxidizing and, accordingly, changing its taste characteristics. Another function of the machine is the cooling of liquids. The machine can support one, two, or more temperature zones.

### 2. Application Layer Communication Protocol Design

The overall system consists of a computer and a dosing machine (Figure 1).

![Figure 1. Conceptual model.](image)

Application software is installed on the computer to use and communicate with the wine dispensing machine. Two separate modules are installed—“operator” and “server”. The server runs continuously and communicates to and from the machines. The operator module serves to create and edit the data of users, customers, cards, and liquids; monitor the condition of the bottles in the machines; dosages performed; perform operations such as adding credit to a customer’s card, and generating reports.

Added to the dispenser are:

- Contactless card reader.

  The reader acts as an input keyboard for entering a customer’s card number. It reads the card from a distance of 3–5 cm and automatically records the card number in the developed software.

- Ten-touch display for the ten bottles. They display information about the corresponding bottle in the machine.

The communication between a machine and a computer consists of the following: when a card is inserted into a machine, the machine reads its number, and when the PC addresses it, the machine sends the card number. As a result, the computer returns the available card credit to the machine.

If the available credit is sufficient, the machine dispenses the selected amount of liquid, then at the next session, it sends information about the bottle number and dispensed amount to the computer. On this basis, the program reflects the database: updates card credit;
updates the amount of liquid for the given machine/position; and saves the transactions to
the database.

The server module fully takes care of communication between the computer and the ma-
chines, processing the information received from the machines and storing it in the database.

3. Server Module

3.1. UML Application Structure of Module “Server” (Figure 2)

Dbcomm Class—performs database connection, login, unsubscribe, and user valida-
tion. DbcommWine class—performs various database operations—adding and editing
customer, card, bottle, user, and machine information. SettingsDialog class—defines the
COM port settings. Worker Class—makes a connection to a COM port and communicates
with the machines on that port. The MainWindow class is related to application control,
control of visual controls, connection with other classes, and communication between this
class and others is completed by calling methods or generating signals.

![Figure 2. UML diagram of the server application.](image)

3.2. Algorithm of Operation of the Server Module

The algorithm of operation of the server module is shown in Figure 3.

![Figure 3. Algorithm of operation of the “server” application.](image)

When the program starts, a check is made for a selected machine to be monitored. If
no machine is selected, the server cannot be started. There is an option to exit the system
or select a machine. If there is a selected machine, the communication settings with the
machine are checked. With the correct settings, the server can be started, and machine status can be monitored and communicated with.

4. Operator Module

4.1. UML Application Structure of Module “Operator” (Figure 4)

BottleQTY class—related to performing an operation on the quantity of liquid in the warehouse. Class Brand—searches for liquid by brand. ChangeDoses class—related to changing the size of a small and a large dose of wine. ChangeMinQty class—is related to changing the minimum remaining quantity of wine in a bottle. Class Credit—is related to changing the available credit in a card. DateTimeFilter class—sets the start and end date for searching database records. Dbcomm class—connects to the database, registers, unsubscribes, and verifies users. DbcommWine class—connects to the database, registers, unsubscribes, verifies users and performs various database operations—adding and editing the information about customers, cards, bottles, users, and machines. Delegates class—performs formatting of the view of the tables and the data in them. DoubleValidator class—performs validation of numerical data (validation of text parameter in a double number). Dragwidget class—manages the visual controls for the machines and updates their status and colors. MergeClients class—related to merging clients. SettingsDialog class—is related to setting the path to the database. TableBottles class—performs adding/editing of liquid bottle data. TableCard class—adds a regular or service card, and changes credit per card. TableClient class—related to adding, editing, or looking up a client card. TableMachine class—performs visualization and processing of the data about the machines and the bottles in them. Class TableUser—performs visualization and processing of user data. MainWindow class—is related to application management, visual control management, connection with other classes, and communication between this class and others is completed by calling methods or generating signals.

![Figure 4. UML diagram of the operator application.](image_url)
4.2. Algorithm of Operation of the Operator Module

When the program starts, the form in Figure 5 opens.

A registered operator, manager, or administrator login is provided. There is an option to exit the system.

There are three options when logging in:

- **Operator**—has the ability to change a bottle in a machine and perform the following actions with the database: add and edit liquid bottles as items; add and remove bottles to/from the warehouse; add and edit customers in the system; add a card to a customer and edit card information; deletes a card from the system; unification of customers; adding and withdrawing credit from a card; card blocking; adding a service card; adding a simple card; search for a card by entered number. Actions are performed repeatedly until the operator logs out of the system.

- **Manager**—has all operator rights and the ability to generate reports: customer consumption report, consumption report by brand, and customer card usage report. Actions are performed repeatedly until the manager is logged out of the system.

- **Administrator**—has all the rights of the operator and manager and the ability to add and edit operator/s; actions are performed repeatedly until the administrator is logged out of the system. After logging out, a return to the main program is made.

5. Experiments

A timer is built into the written software that checks for communication problems every second.

After successful registration, the machines and their positions are visualized (Figure 6):

Machine visualization field—10 machines are visualized. Each machine has a name and the state it is in. The color of the machine indicates the status of the machine:

- **Gray**—no connection to the machine and no information about the bottles.
- **Green**—the machine is connected, and the condition of the bottles can be monitored.
- **Yellow**—there is an empty bottle or a changed bottle awaiting operator confirmation.
- **Red**—error connecting to the machine/unable to connect to the machine.
Tests were performed using wine bottles in the machines. There are 10 bottles in each machine. When selecting a given machine, a dialog window appears with data on the bottles in the machine (Figure 7):

![Figure 6. Visualization of the machines.](image)

![Figure 7. Data on the bottles in the machine.](image)

The information displayed for each bottle is:

- Position (Pos) of the bottle in the machine;
- Status (Status) of the bottle. There are six states the bottle can be in: OK—everything is fine. The information row for this bottle is colored white; N.A—no information on the bottle. The information row for this bottle is grayed out; CHANGED—the bottle has been changed. The information row for this bottle turns orange; EMPTY—the bottle is empty. The line with information about this bottle turns yellow; WAITING—waiting to update information about the bottle after its replacement. The information row for this bottle turns blue; EXPIRED—the time for the maximum stay of the bottle in the machine has passed. The line with information about this bottle turns red.
- Brand of wine (Brand);
- Current amount of wine in the bottle (Current Quantity);
- Residual amount of wine (Residual Quantity);
- Bottle volume (Bottle Volume);
- Amount of wine in a small dose (mL)—Sample Volume;
- Price of a small dose (20 mL)—Sample Price/20 mL;
- Amount of wine in a large dose (mL)—Glass Volume;
• Price of a large dose (125 mL.)—Glass Price/125 mL;
• Date and time of placing the bottle in the machine;
• Date and part, for maximum stay of the bottle in the machine.

When selecting the “Bottles” button, the following window is displayed (Figure 8):

![Figure 8. Bottle data in the database.](image)

Main elements: list of all wines in the database; filter field;

• “Add Wine” button—adding a new record to the database.
• “Edit Wine” button—changing the data for a bottle that has already been added to the database.
• “Add” button—adding bottles of a given brand to the warehouse.
• “Remove” button—removal of wine bottles of a selected brand from the warehouse.
• “Update” button—updating the data from the database.

6. Conclusions

The realized software applications find real applications in a wine dispensing system and a water dispensing system. With the wine dispensing machines, it is possible to constantly monitor for any irregularities in the communication with the machines, the state of the machines, and the bottles in them, and perform certain operations such as—loading credit on a customer’s card, generating reports, and others. Several additional and more sophisticated features have also been added, such as a wine storage function. An opened bottle can be stored for up to 28 days after opening without affecting the taste of the wine. Another function is wine cooling. The machine can support one, two, or more temperature zones, i.e., white, rosé and red wines can be chilled in one machine. The implementation ensures automation and autonomy in the dosing of wine. With the push of a button, it is possible to perform one dosage and withdraw credit from a customer’s card for the dosage performed. No external intervention is required to perform the operations. If the machine fails to perform a certain operation, information is sent to the operator’s console about a problem that has occurred. When implementing the applications in a water dosing system, an additional function is water cooling.

The applications realized in this way can be implemented in various areas where it is necessary to dose some liquid with small changes, such as beer, juice, soft drinks, and even alcohol since the system has the possibility of limiting the amount of liquid, after which the card is blocked and no more can be purchased.
Author Contributions: Conceptualization, G.S., M.K. and I.P.; methodology, G.S.; software, G.S.; validation, G.S., M.K. and I.P.; formal analysis, I.P.; investigation, G.S. and I.P.; resources, G.S.; data curation, M.K.; writing—original draft preparation, G.S.; writing—review and editing, G.S., M.K. and I.P.; visualization, G.S.; supervision, G.S.; project administration, G.S.; funding acquisition, G.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable due to privacy.

Conflicts of Interest: The authors declare no conflict of interest.

References

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.