

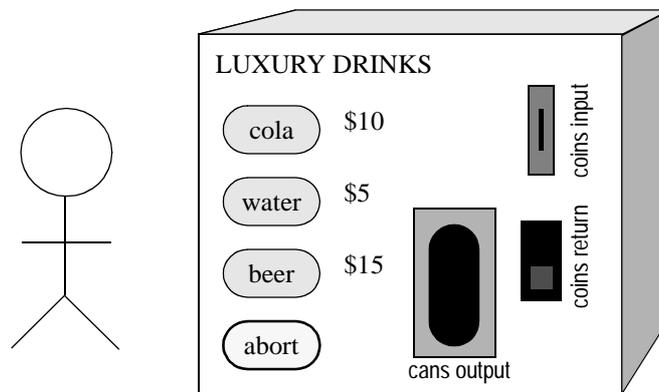
# Can Machine

Version 1.0

## 1. Requirements

### 1.1 Introduction

The system described is a simple soda-machine, where a customer can buy cans with three different types of expensive luxury drinks: cola, water, and beer. Water costs \$5, cola costs \$10, and beer costs \$15. A picture of the machine is given below.



We only concentrate on the software of the machine, and to make the system simple we make the following assumptions:

- The machine will never run out of cans of any kind.
- The machine will always have enough coins to give back as change.
- The hardware takes care of coin recognition.
- We assume that the user always starts by pressing one of the drinks buttons.

The system consists of two different components, `CanHandler` and `MoneyHandler`.

### 1.2 Functional Requirements

#### 1.2.1 CanHandler

1. `CanHandler` is responsible for monitoring the drinks buttons and releasing the cans to the customer.
2. When the user presses a button for a specific type of drink, the `CanHandler` shall determine the price.
3. If the payment procedure is interrupted, `CanHandler` shall not release any cans.

#### 1.2.2 MoneyHandler

1. `MoneyHandler` shall keep track of the coins inserted.

2. MoneyHandler shall detect whether the amount is correct for the desired drink.
3. MoneyHandler shall check that the user does not wait too long before or between coin insertion. Maximum time is 10 time units.
4. If the paid amount is greater than the price, MoneyHandler shall return the amount of exchange back to the user.
5. If the payment procedure is interrupted, MoneyHandler shall give back the money that might have been inserted into the input slot.

### **1.2.3 Special Cases**

6. No special cases are considered, like for example where the user starts a transaction with coin insertion or pressing the abort button.

