HackItAll Documentation

PANDA

Girla Ionut-Alexandru

Boncu Adriana Iuliana

Anghel Diana

1) Design idea

The vending machine stores physical copies of the products. Every time someone buys a product (no matter the payment method), a physical copy of the product he paid is dropped from the machine's shelf to a small opening placed on the bottom part of the machine which can be accessed by that person to retrieve his product.

If there is any need for change there will be a **20 characters change code** sent on that person's email. This code can be used on new purchases from any machine connected to the same database.

Our base idea behind the implementation of a vending machine was that we have a java application that generates various windows with pop-up dialogs. This java application should get the information about available products from a MySQL database (in this example created on a same host, but it can be created on a remote host as well)

Any transaction realized will be store in the MySQL database.

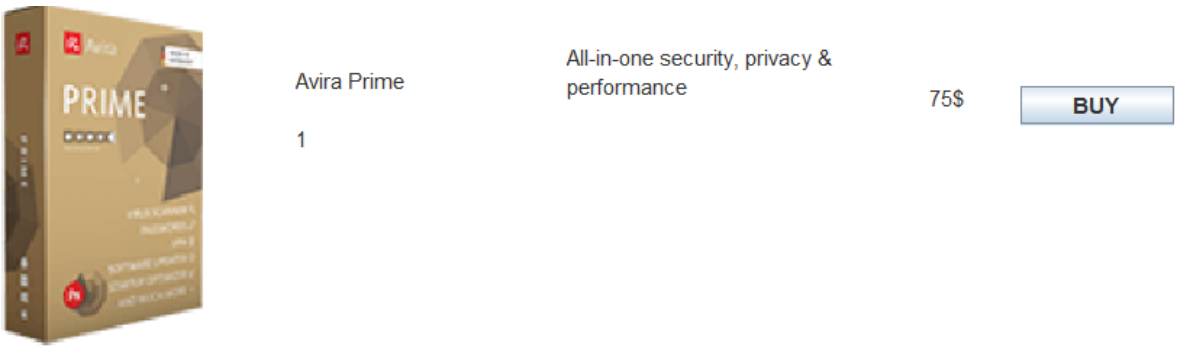
The validity of the credit card is checked by the Luhn algorithm.

2) Purchase flow:

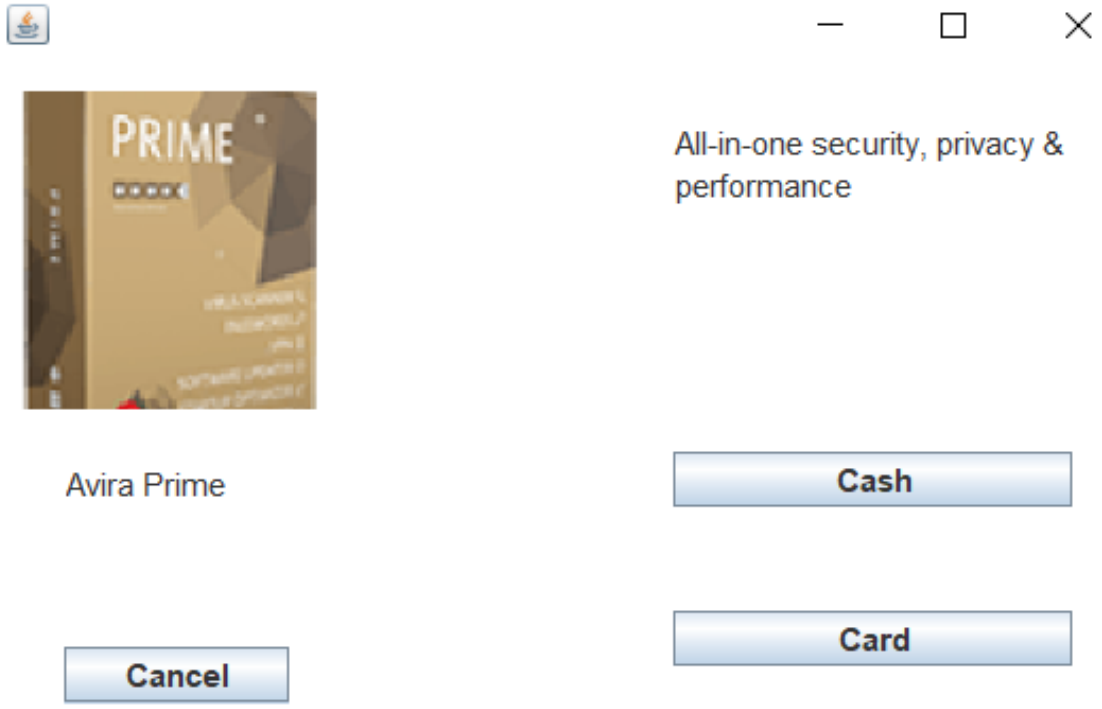
A) Payment without change code

The user is prompted with the following screen that will present all the available options. There is a picture of the product box, the product name, the quantity available at this machine, a short description of the product and it's price.

The buyer selects one product by pressing the "BUY" button :



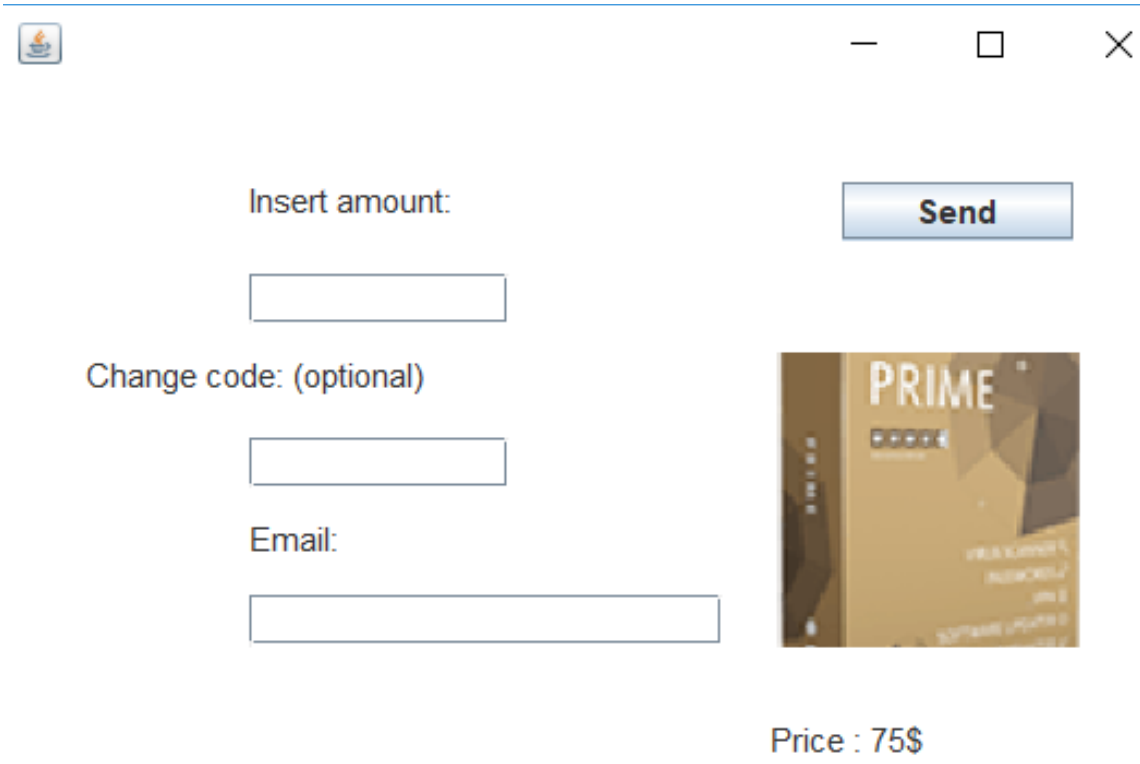
A small pop-up window presenting the product description, name and picture appears. From this window the buyer can select his preferred payment method.



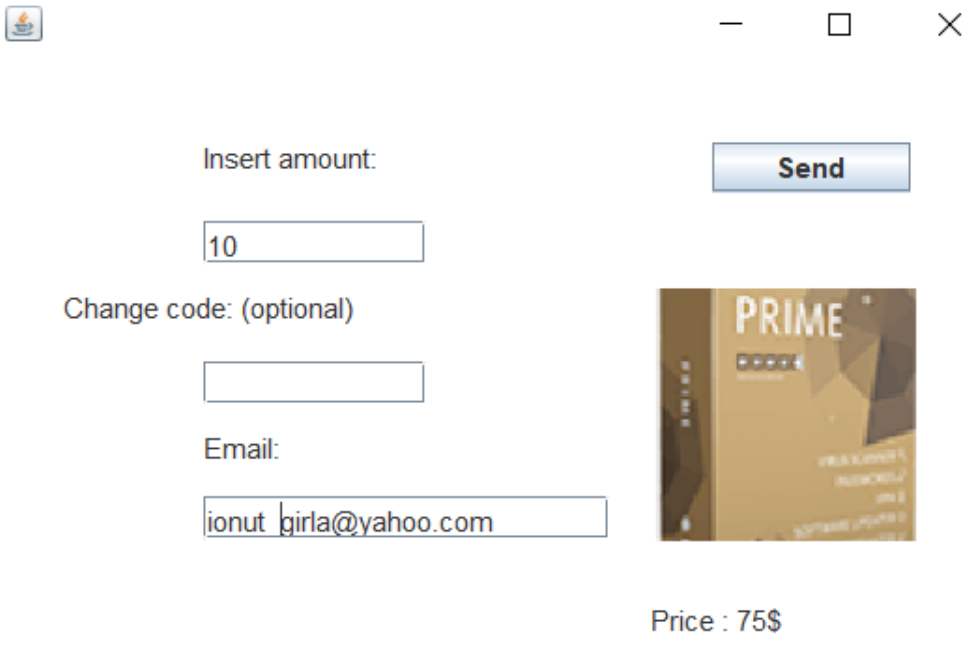
If the payment method is cash, the following screen will ask for the inserted amount of money, a change code and an email address.

The email address is used to send the transaction details and the change code to the user.

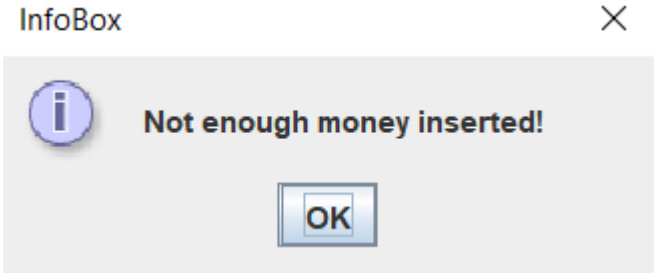
We used “Insert amount” field to simulate the actual use of a vending machine cash sensor that will detect the payed amount automatically.



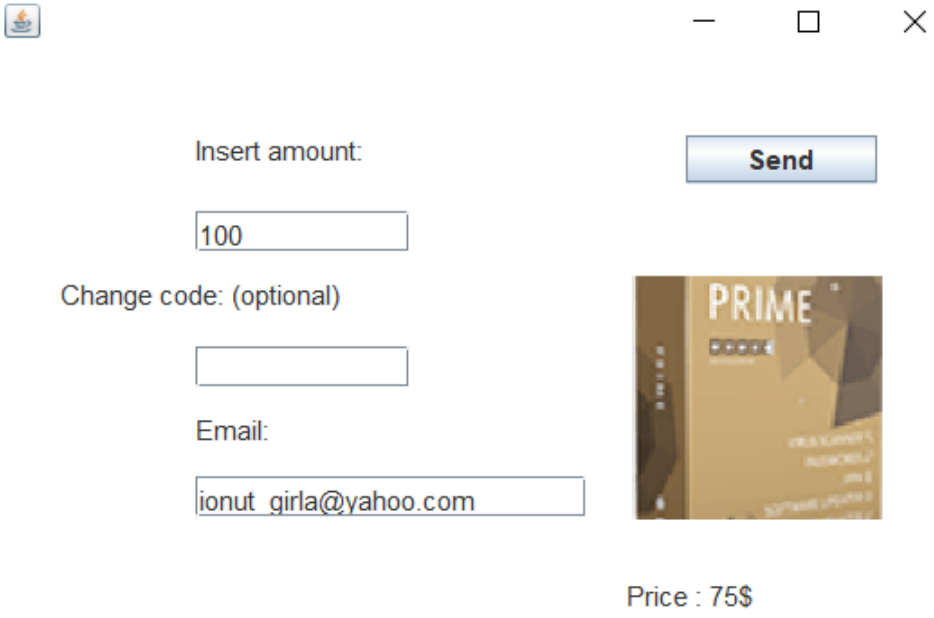
In this example the inserted amount is smaller then the price of the product and there was no change code used.



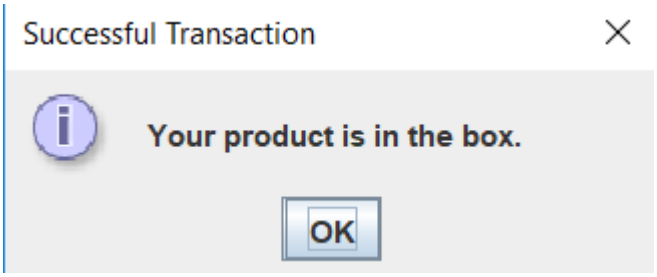
The following pop-up will appear informing the user he needs to insert more money.



If the amount of money inserted + any change generated by the change code is bigger then the price of the product the transaction is registered in the database and a confirmation email is sent to the user.



And a small popup informing the user of the success of the transaction appears on the screen.



The machine reloads the first page after the transaction is completed:

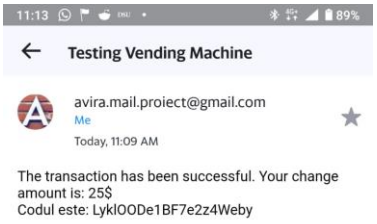
Avira Prime

0

All-in-one security, privacy & performance

75$

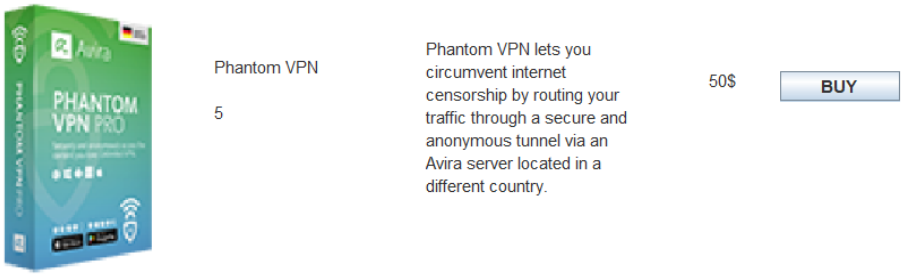
Confirmation email example:



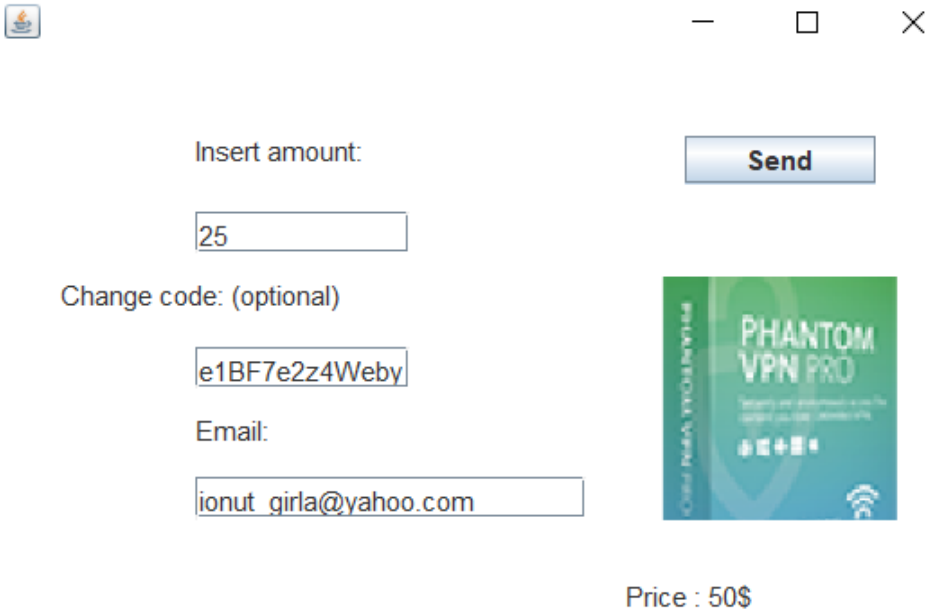
Database entry for the last transaction:

036	2020-03-09 11:09:58	001	Cash; Change ammount:25$ Change code: LyklOODe1BF...	ionut_girla@yahoo.com	25.00

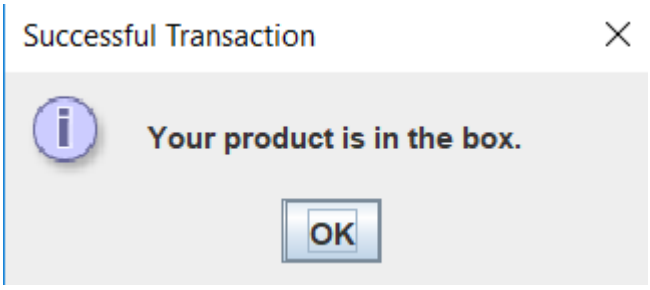
B) Cash buy using change code : LyklOODe1BF7e2z4Weby for 25$



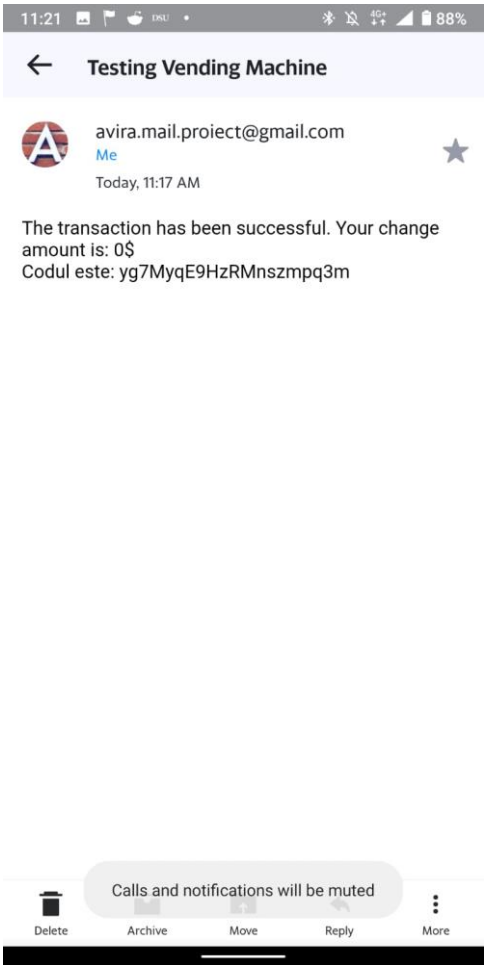
After using the 25$ code we generated earlier we will be able to finalize the transaction even we just inserted 25$.



Transaction confirmation appears and the confirmation email is sent



Email for the cash + change code transaction:



The code we used (LyklOODe1BF7e2z4Weby) will have it's **Change_amount** field updated to 0, the date of this update is stored in the second field of the entry and a new transaction is inserted in the database with a new change code.

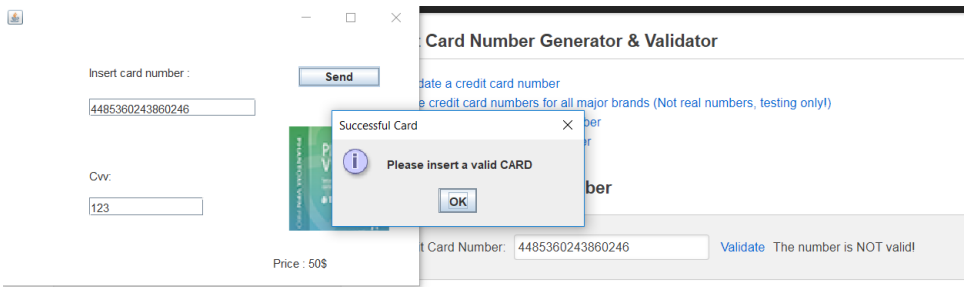
036	2020-03-09 11:17:37	001	Cash; Change ammount:25$ Change code: LyklOODe1BF...	ionut_girla@yahoo.com	0.00
037	2020-03-09 11:17:37	003	Cash; Change ammount:-25$ Change code: yg7MyqE9Hz...	ionut_girla@yahoo.com	0.00

C) Credit Card Payment:

If the chosen payment method is credit card we will need to get the card number and Cvv from the user (since we don't have a better method of checking the validity of the card number and Cvv we used the Luhn algorithm)

Credit payment window:

If the inserted Card number is not valid we get a pop-up message that asks for a valid card number.



Using a valid card number will store the card data in the database and will finalise the transaction. No email is sent here since there is no need for any change code.

Insert card number :

Send

4485360243860248

Cvv:

222

Price : 50$

Successful Transaction

Your product is in the box.

OK

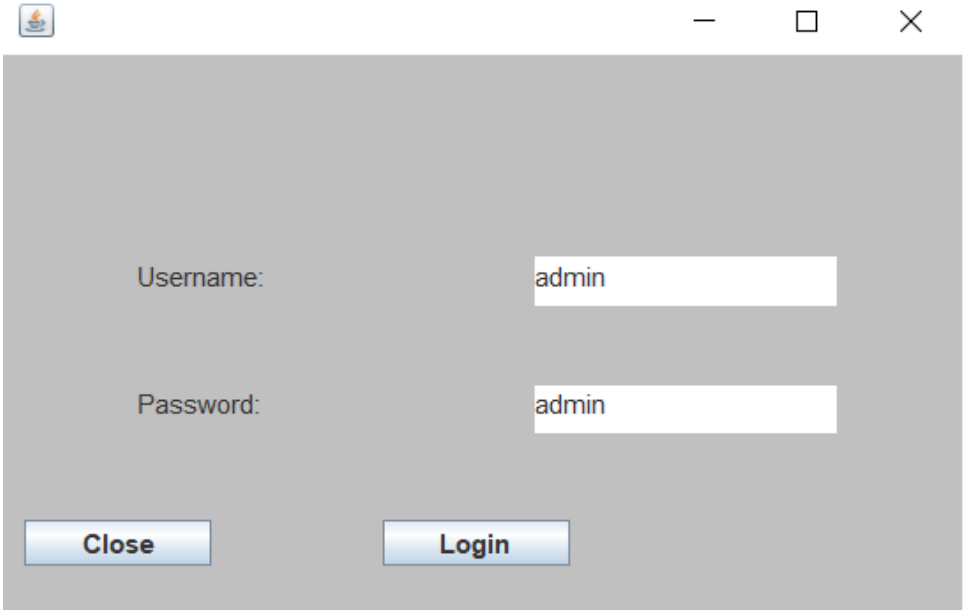
card number

4485360243860248 Validate The number is valid!

D) Administration and setting up the stock of the m Administration and setting up the stock of the machine

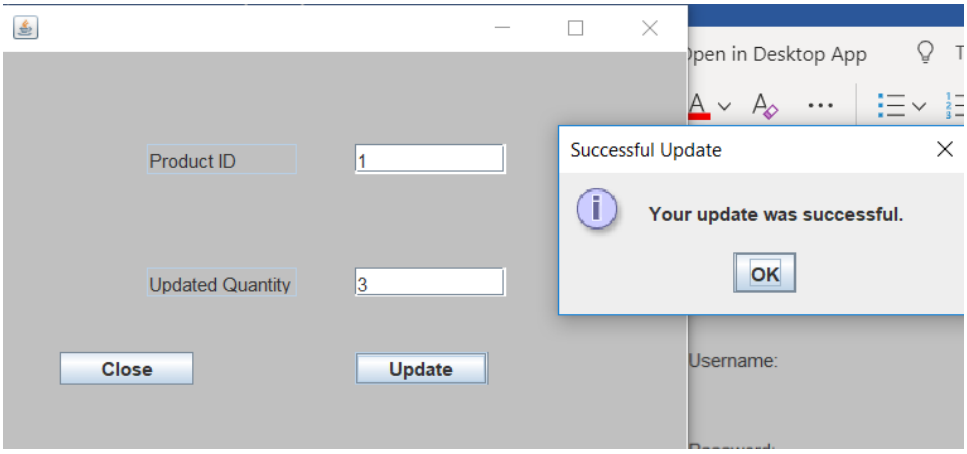
In the bottom right part of the window there is an admin button. This allows the machine to be restocked.

Pressing the button will prompt the user with a login page:

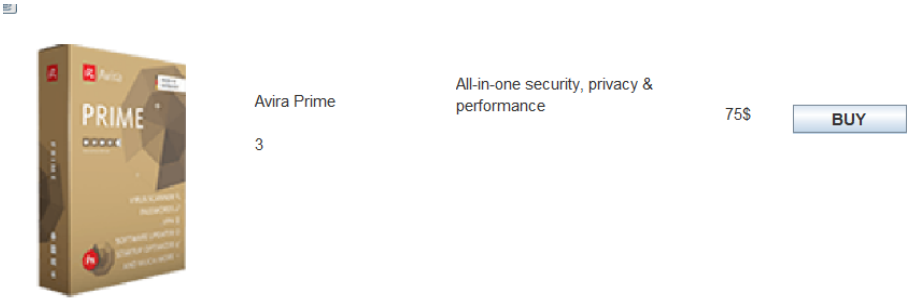


After inserting the credentials and pressing the **Login** button the administrator is prompted with a new window that asks for the id of the product: 1=> Avira Prime … 6 => System Speedup

After chosing the added amount, the database will be update with **existing_stock+new_stock**



Result of the last action:



The database entries for the three posible operations (Restock, Card Purchase and Cash Purchase)

037	2020-03-09 11:17:37	003	Cash; Change ammount:-25$ Change code: yg7MyqE9Hz...	ionut_girla@yahoo.com	0.00
038	2020-03-09 11:31:37	003	Card; Card Number: 4485360243860248 CVV: 123	null	0.00
040	2020-03-09 11:39:35	001	ALIMENTARE Avira Prime	Admin	*NULL*

3) Error protection

To make sure there is no double charging of the same product we only allow one transaction at a time. This transaction will be stored in the database at it’s own timestamp. To make sure there are no duplicate transactions, the data base is checked for duplicate entries occasionally.