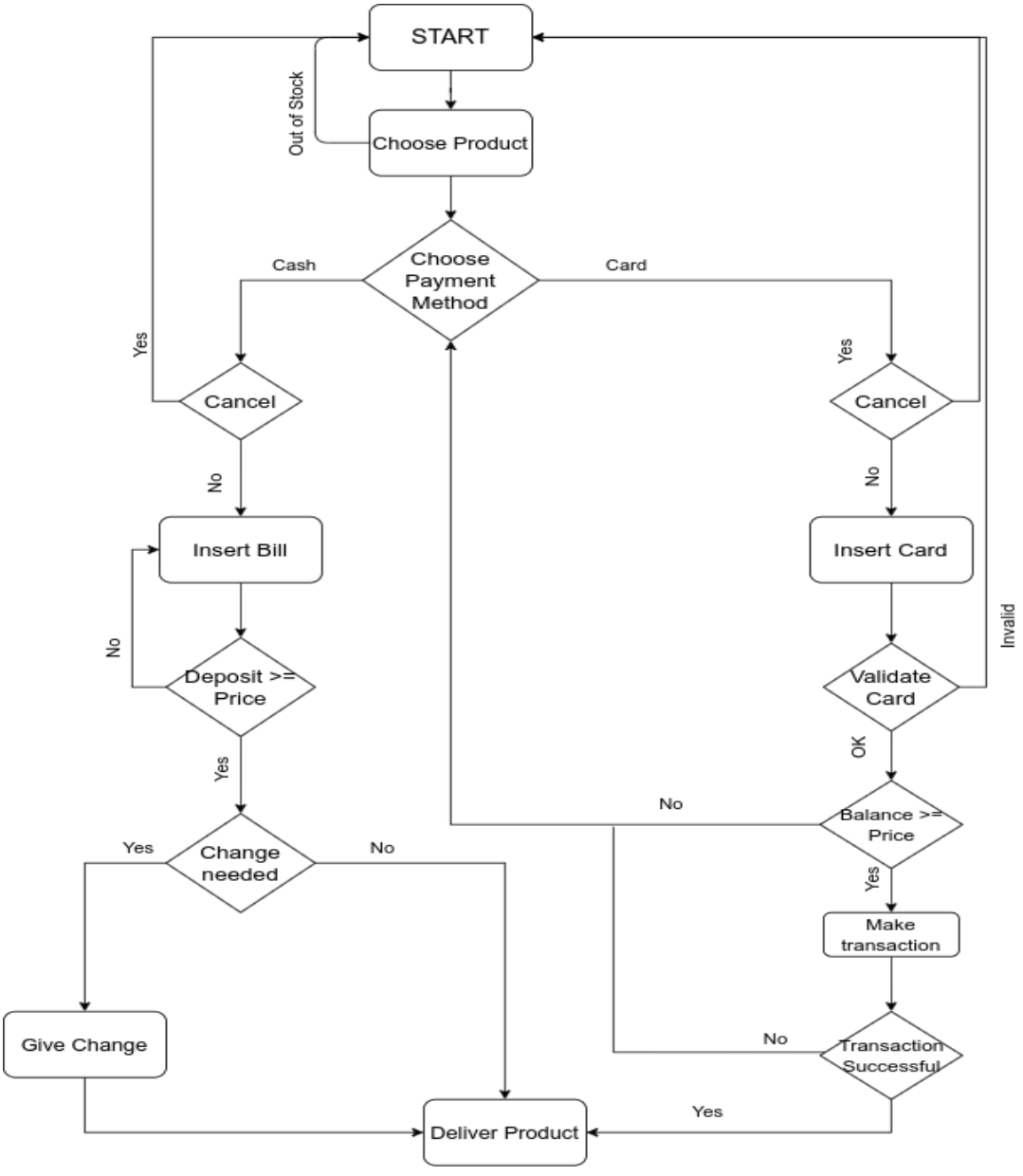
AVIRA Vending Machine

1. Introduction

For this task we chose to implement an application that simulates as close as possible the functionalities of a vending machine used for distributing Avira Products in the hall of our university. Our goal is to keep the flow of the application as simple as possible and give the user enough options to make his interaction with the system easy and reliable.

2. System Design

For this application we found that the most suitable is to follow the design and the flow of a Finite State Machine, or FSM, which is a computation model that comprises states, inputs and outputs and can be used to represent and control execution flow. Because of the properties of a FSM we found it crucial to use it for the design of the vending machine .



3. System Functionalities

First, we considered the basic functionalities of a vending machine: a user selects a product, pays for it and the product is delivered by the machine after the verifying that the payment was correct. So for our simulation we implemented this base functionality among some new ones that could be implemented in the software.

The flow of the application is very easy to follow:

- When nobody interacts with the machine it goes into a waiting state that should be optimized at hardware and software level to be energy efficient
- A user interacts with the machine for the first time by pressing any button and this is the point where our simulation of the program begins
- The user must choose what product he wants to buy and since this may not be an easy choice to make we should display some information about each product on the vending machine. For this we thought about having a QR code with a link to Avira page for each product available so the user can easily access it and also the information about the product can be simply updated when something changes.
- After the user chooses the product he wants to buy, he can choose to pay cash or card.
- **Card**. We thought that the card functionality is absolutely necessary not because the cards are more popular but because of the quick growth of the mobile payments applications such as Apple Pay and Google Pay that allow users to pay with their smartphones or their smartwatches. We simulated the use of the card by manually typing the card info such as card number, name, expiration date and CVV code.
- To verify the card, first our application uses Luhn algorithm that validates every card number and is used by the majority of cards. Then a transaction is made and if it is successful then the payment is considered done and the product must be delivered to the client.
- **Cash**. Regarding the cash payment method we decided to accept only 5, 10, 20, 50, 100 bills to simplify the use of our simulation, but we consider that in a real world environment it would be desirable to accept all bills and even coins.
- To verify the cash bills the vending machine should use a magnetic reader to detect the magnetic signature of a bill to ensure it's real and determine its denomination. After the user inserts all the money needed, the machine checks to see if any change is needed and if so it delivers the change to the client along with the product it selected.
- We considered that a user should be able cancel the ongoing purchase any time, even after he inserted a part of the requested money. If he chooses to cancel the money inserted should be given back as change.

3. Statistics and Logging

The application keeps a log of all transactions, either successful, failed or cancelled. The logs can be used to track the purchases and all the details about any operation made:

- timestamp
- product name
- product price
- payment method
- transaction status.

We considered that the memory of the machine would be large enough(1-4 GB) to keep a part of the logs locally. When the memory is full it would be nice that from one request all the logs are copied to a remote server and the memory is empty again. This implies having an internet connection.

Keeping these logs allows us to generate statistics very easily. Regarding that an internet connection is provided, a log management system can be connected and help parse the log files and view in real time how the machine functions.

Important statistics should be: most purchased items, most used payment methods, waiting hours, how the stock changes, etc. These info can be turned into visual statistics and graphics using a client such as Kibana, Grafana etc. to interpret the data easier and deliver hardware and software solutions more efficiently.

Also using this real time solution, allows the operator to see how the stock is modifying and when should a restock be scheduled and what quantity of each product is needed.

Following the statistics, the provider should be able to adjust the quantity from each product that is kept in the machine because one product might get out of stock quicker than others and it would be nice to have a bigger amount of that product in the machine to avoid quick restocks.

4. Further Improvements

We kept our implementation at a base level but we continued to think about some improvements that would make the Avira Vending Machine to be one of the most technologically advanced.

- **Payment Methods**
    - The cashless method covers pretty much of the payment methods used: credit/debit cards, Apple Pay, Google Pay, etc.
    - We think it would be nice if the machine would be able to accept payment through a PayPal account as there is an amount of people who uses this on a daily basis.

    - Another payment method would be through a QR code. The machine generates a QR code with the current transaction and the user only has to scan the QR code with his smartphone and is redirected to the payment application.
    - A payment method that we found interesting but more complex is one that uses biometrics such as fingerprint or face recognition. The concept is already implemented in many places, but the complexity comes from securing the data about the person that uses this type of payment because a lot of data about a person are being used and brought together.
- **Hardware**
    - It would be nice to have some sensors installed that could help at Real-Time Problem Reporting.

5. Setup and Running

Implementation can be found here: https://github.com/vladcocos/hackathon

To run the application, you will need access to a command line (Unix terminal or Windows cmd) with Python3 installed.
To start it, run the following command:

alina@alina-pc:~$ *python3 menu.py*