AUT.840-2022-2023-1
Industrial Informatics
Assignment 1

Jukka Hirvonen [H218618],

Valtteri Nikkanen [282688],

Jouni Kokkonen [151091078]

Table of contents

Introduction 3
Functionality/Idea of the code 3
Use Case Diagram 4
Class Diagram 5
Flow Chart Diagram 7
Encountered Problems & Solutions 8
Result 9

Introduction

The purpose of this assignment was to design a reliable system where an orchestrator controls a workstation that includes a robot and conveyors. The design phase was implemented by creating ULM diagrams. Based on the diagrams, a program was created which was first tested with an internal mock loop, then with the Postman REST Client and finally live in the FASTORY Lab. After testing bugs and problems were located and fixed to find a reliable solution.

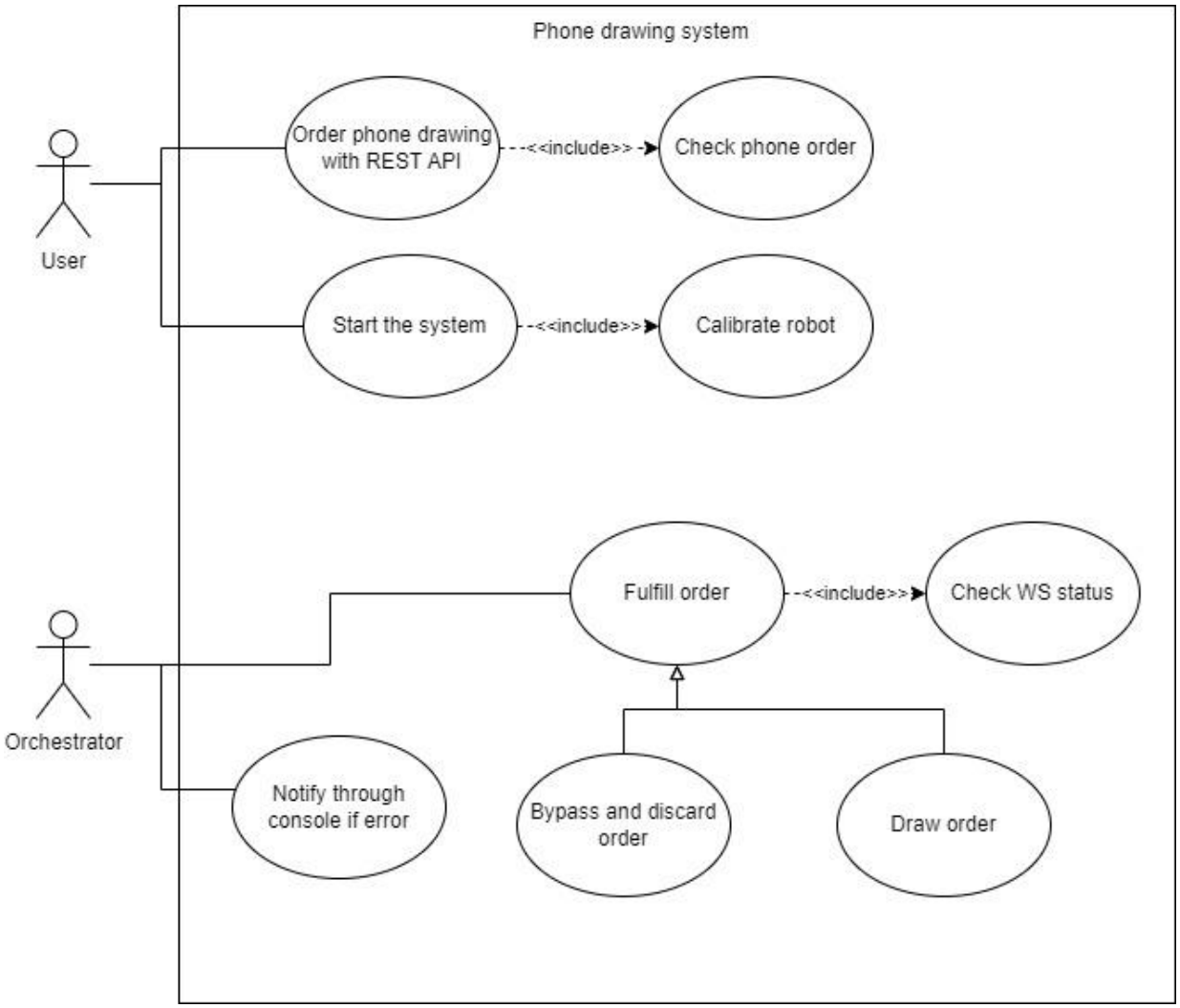
Functionality/Idea of the code

Our main idea was that the workstation used in the demo is part of a bigger line of the workstations. We gather orders to an order queue that we then pair to pallets that arrive to the zone 1 of the workstations. Our code pairs the order with the pallets RFID that is then handled as the pallet arrives to the zone 3. The order is then read and drawn to the pallet. If the workstations zones 3 and 2 are full, the order is discarded, and the pallet moved to zone 4. If there are no pending orders the pallet is also moved to zone 4 to not block the line and keep things moving. Everything is driven by the events that our code and orchestrator have subscribed to, and we get from the workstation that the code then decides what to do with.

Use Case Diagram

Use case diagram that is shown in figure 1 was created in the design phase of the project and still holds true after the project was completed. We found to possible actors and the possible use cases for them

Figure 1: Use case diagram for the assignment



Class Diagram

Class diagram was first made to help us figure out how the problem could and should be split into classes that could be programmed with OOP approach. In figure 2 first draft of our proposed classes is shown.

Figure 2: First draft of the class diagram

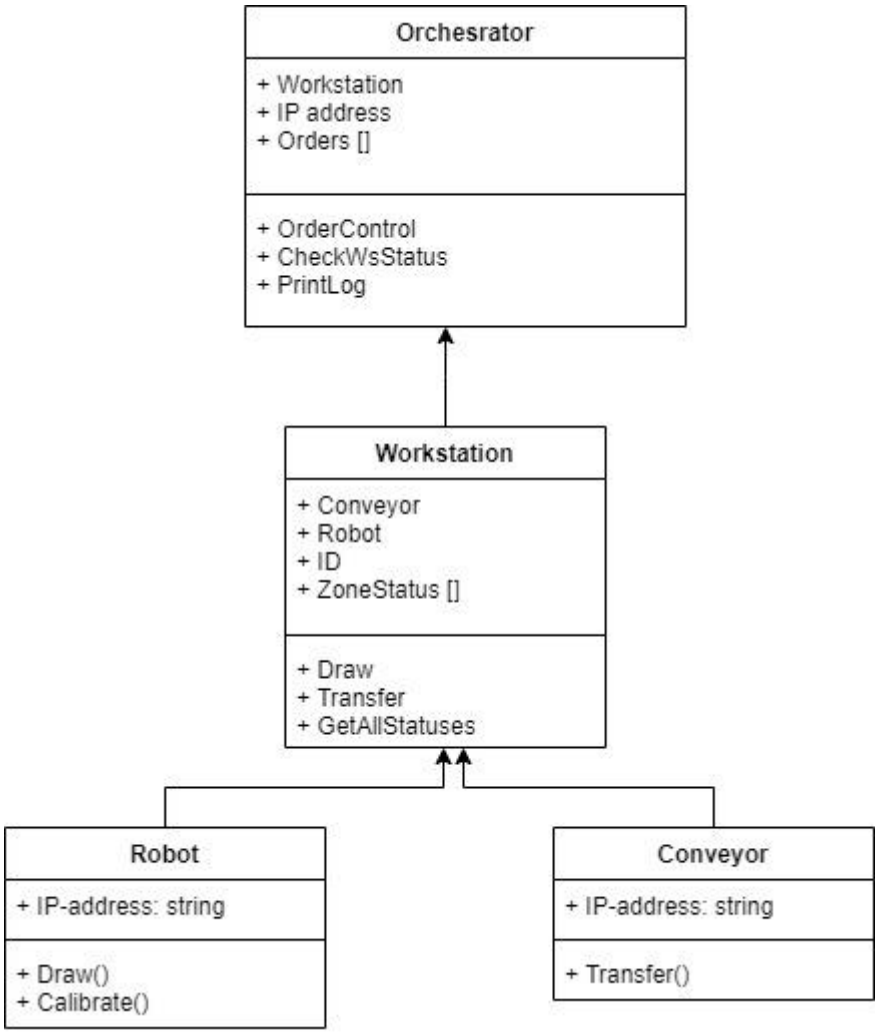
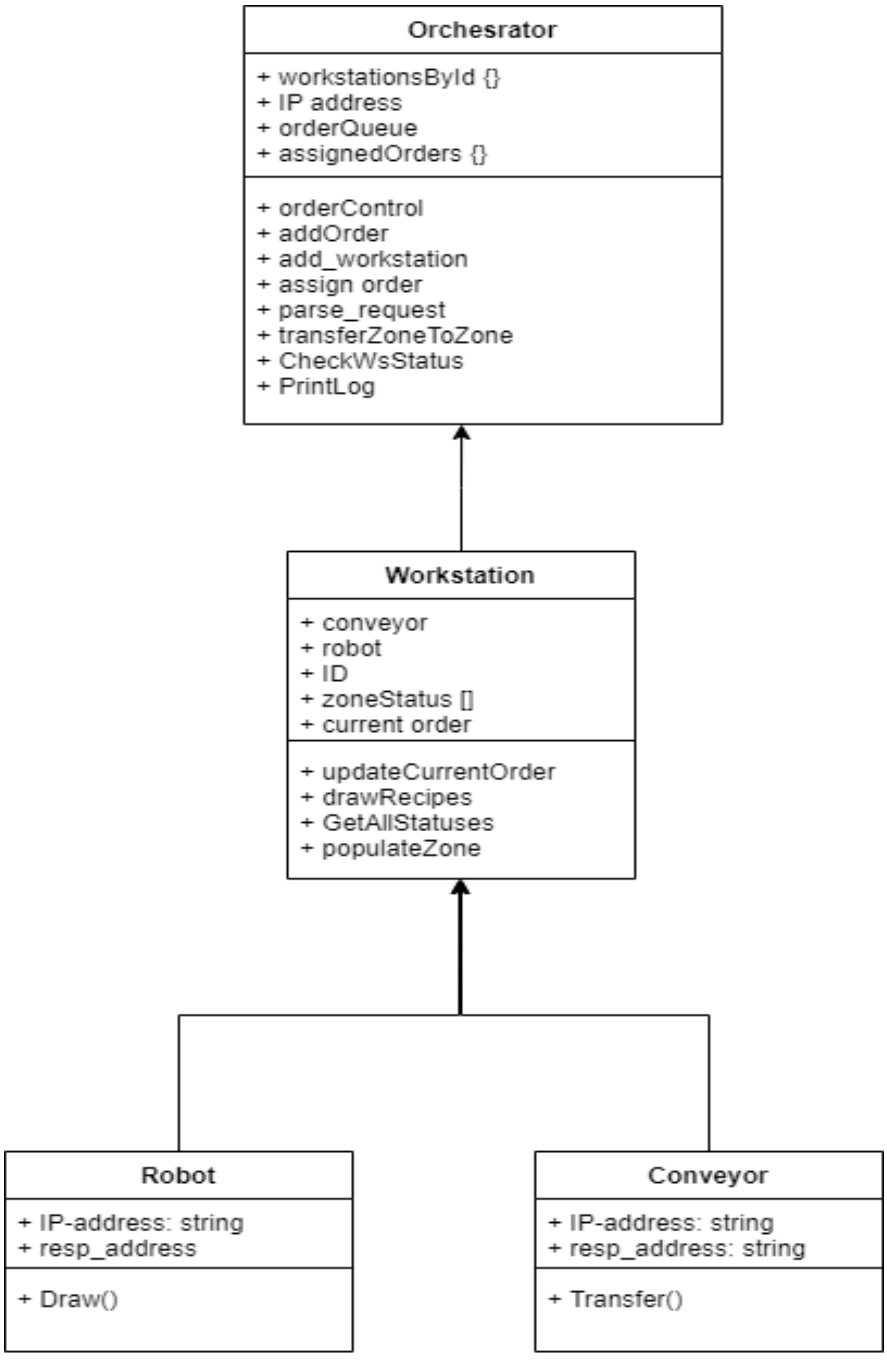


Figure 3 on the other hand presents the final versions of the classes that were implemented in the python code. The main structure between classes is the same but there are more methods and attributes in the classes as they were added when felt necessary for the implementation.

Figure 3: Final version of the class diagram



Flow Chart Diagram

Figure 4 shows our first draft of a flow chart that tries to explain how our system works. It does a mediocre job of explaining what is happening but does not fully accurately show what was truly going on in the final version of the project.

Figure 4: First draft of a flow chart that presents a pallets journey through the workstation.

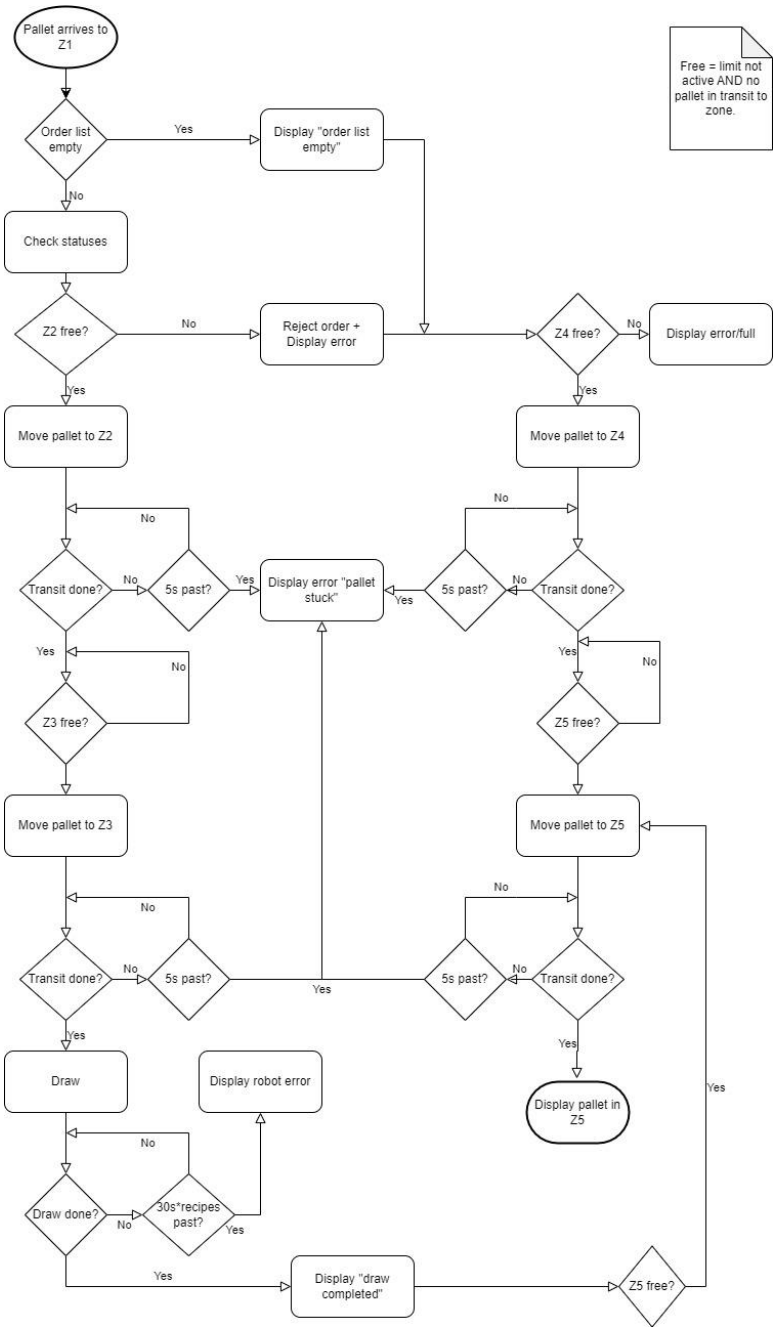
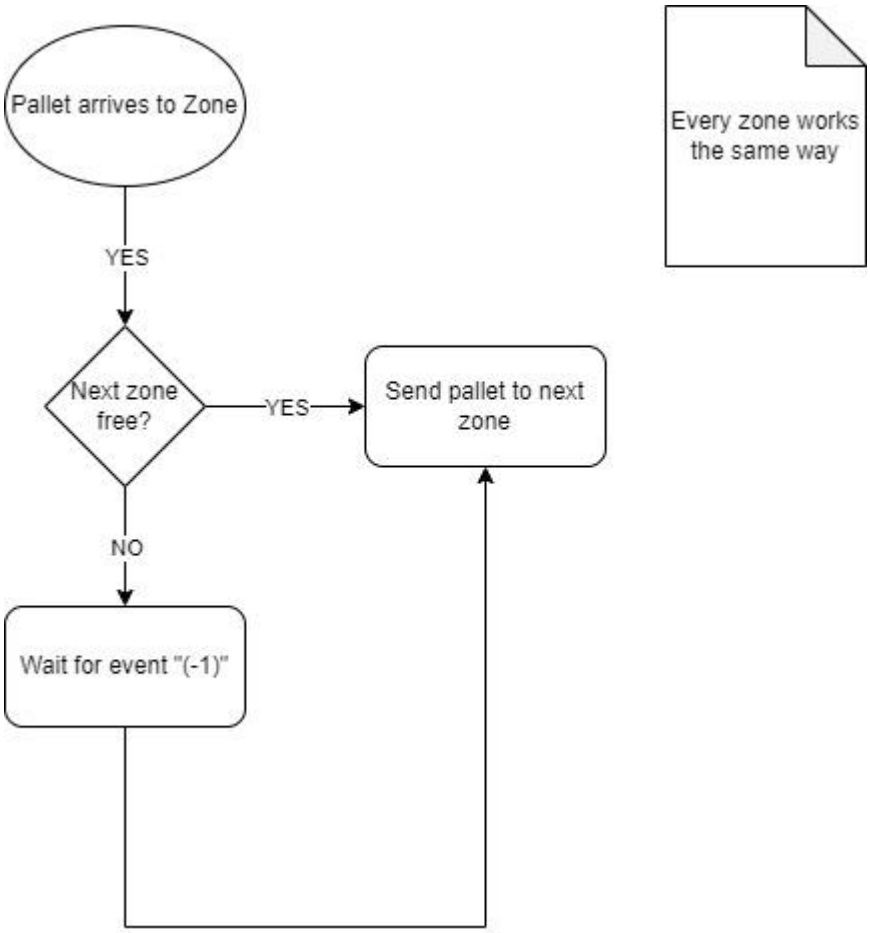


Figure 5: Flowchart diagram show how each zone works.



Encountered Problems & Solutions

We encountered problems with the functionality of some workstations, such as the ID reader wasn't working, or JSON response to zone status request was empty. After troubleshooting we switched to a working workstation. In terms of time usage, our biggest problem was robot's power switch that we located after 3 weeks of troubleshooting.

On the software side, the problems were mostly related to finding the correct data types and request formats. Determining the format of the events sent by the RTU's required some live testing as we couldn't decipher the fully correct format from the Fastory pdf.

Result

The solution was not flawless but was functional. Passing line worked and the pallets did not collide with each other. Some minor bugs were still present at the demo but have been ironed out since.

We used the request, flask, time and json libraries in the final version of the python code. Final project was also based on event driven architecture. Everything worked based on the events that were received from the workstation. The project weas also fully based on OOP which means that it should be easily scalable by creating more instances of workstations, conveyors, and robots.