# Specification

We are using Fisher Technik to build the machine and a PP2 processor to control it.

We have to make a so called sorting machine. This machine should be able to separate, by colour, small black and white plastic discs. The only real requirement in achieving this is that we need to use at least one conveyor belt. We think the conveyor belt is our biggest disadvantage. It’s rather slow compared to the detectors.

# Design decisions(prepared/improvised)

## Priorities

In order to make decisions we will have to specify our priorities. Our priorities are:

1. The correctness of the sorting (as correct as possible)

2. The speed of the sorting (as fast as possible)

3. The robustness of the machine (it should not break very easily)

4. User accessibly (It should not be hard for the user to do the actions required)

5. Difficulty of building the machine (as low as possible)

6. The amount of parts of the machine (as few as possible)

respectively.

## Validation

1. To check correctness of the sorting machine we will prove that the code controlling the sorting machine is correct. We also will perform long-term tests.
2. For the speed we will measure how many discs are processed in a certain amount of time and then we could try to improve it.
3. We assume that if the machine didn’t break during the building and testing of it, it’s robust enough.
4. To check user accessibility we check if the machine is compatible with the user constraints.
5. And 6: Check if there are useless parts.

## Reasons

1. Money
2. Because a user has to control it
3. So clients could be interested in having the sorting machine, because they can implement it in their company on a short notice.
4. Lowering the building costs, ease error detection (because you have less parts to worry about) and simplifying the overview of the machine.

# Machine design

## Use-cases

Use-cases are use scenarios with the machine. We are using use-cases to make the high level specification more specific and to validate if we met the requirements.

## Example

One of our use-cases is the starting of the machine. The brief description is that the operator starts the machine, machine parts go to their initial state and the machine starts sorting process. If this use-case is successful the machine starts the sorting process. As a precondition the machine has to be in its initial state. To trigger this use-case the user has to perform an action on the machine. The basic flow of this use-case is that the machine puts the devices in their initial state and the machine starts the sorting process.

This sounds all very silly and repetitive, but that’s how use-cases have to be specified.

## User constraints

User constraints limits the user’s freedom in controlling the machine. This is necessary so the machine achieves its purpose according to the use-cases, without influencing the condition of the machine. An example of a user constraint is that when the abort button is pressed, the operator should remove all discs still present on the belt.

## Safety properties

Now we know how the machine is going to be used it might be nice to cover safety. We do this with safety properties. A safety property is a description of what needs to be ensured to guarantee safety. To give an example: one of our safety properties is that after pressing an emergency button, within 50 ms there should be no moving part in the machine.

## Sketch

Now we were talking about the use-cases, user constraints and safety properties, you might have forgotten that we were talking about machine design. So let’s take a design decision we made.

For the use-case “Sort unsorted discs” we need something that disposes discs one by one on the conveyer belt. We came up with two designs and build both of them.

\*Show designs\*

Both designs make use of the hollow tubes stacked on top of each other. This is convenient because they are reliable in containing the discs and fast because if the bottom disc is removed the next disc, if any, will drop down to the position of the first disc. Robustness is covered too, because there are few parts that may come loose. For user accessibility it’s easy to put new discs to be processed at the top of the tubes.

In the first design there’s a turning wheel with a cam that is able to push the discs one by one onto the belt. A wall to the left of the container makes sure the disk is pushed up and not to the left.

In the second design a moving block would push the discs onto the belt. The block would be driven by a lever attached to a rotating wheel.

Both designs are consistent with the use-case. Tudor will tell you more about our priorities, but both designs were reliable and there was no difference in speed. Both designs didn’t hinder the user, so user accessibility is all right. At the end we chose for the first design, because it was easier to build, had less parts and was a lot more compact.

Then now I will give the word to Tudor, so he can tell you about design decisions.