# 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.

7. Lowest amount of space (as little floor space used as possible)

## 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.
6. -
7. Correlated with 5 and 6 when it comes to checking for useless parts, also seeking if parts can be replace with smaller ones without compromising the other higher priorities ( correctness , robustness etc).

## 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.
5. May ease implementation or can be the perfect solution when the customer puts emphasis one space.

# 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.

Here you can see our first idea to deposit disks.