

Baggage Handling System

GROUP 35

LUIS RODRIGUEZ IGNAS ŠABLINSKAS STEPHAN SALVI

NORBERT ROMANSKI ERNESTAS RAUDONIS VICENTE VEGAR

Fully Functional System



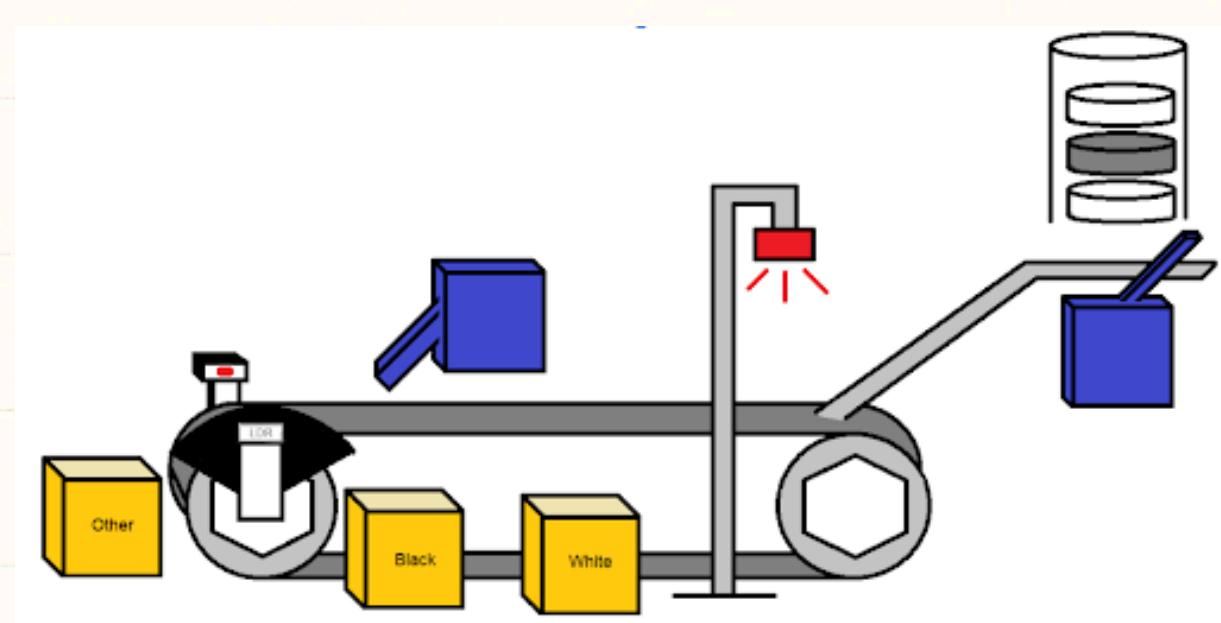
THE BACKLOG

No.	Robot Task	Purpose	Priority	Sprint N°	Status
1	Have proper infrastructure	Ramp is used to place disks on the conveyor belt, sensors are mounted to the conveyor belt	Must	2	Done
2	Mechanically push items off the conveyor belt	Place disks in boxes	Must	2	Done
3	Have a rotary encoder	Detect if the conveyor belt is stuck	Must	3	Done
4	Distinguish between black, white and coloured disks	Sort disks by color, placing them in correct boxes	Must	4	Done
5	Have buttons to activate and turn off the robot	Button is used to start feeding the disks and running the sorting system	Should	5	Done
6	Have a switch to start and stop the conveyor belt	Be able to start conveyor belt when robot is activated and stop it when an error occurs	Should	5	Done
7	Have a manual override mode	Allow user to manually sort a disk in the case that something went wrong with sensor or an unexpected disk is put on the conveyor belt	Should	5	Done
8	Detect and handle errors	Conveyor belt and sorting of disks stops and robot waits for an input to restart	Must	6	Done
9	Have LEDs that signal current status of the robot	Green LED to signal that robot is running, yellow LED to signal that robot is in manual override mode, red LED to signal that an error has occurred	Must	6	Done
10	Report its processes on a website	Display the current status and actions of the robot, the number of disks sorted (per colour) and any errors	Should	7	Done
11	Outputs audios	Alert users of current status and errors	Could	7	Done

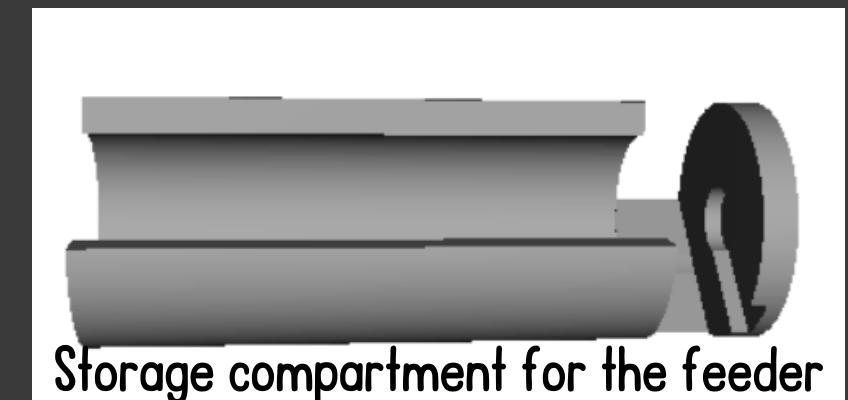
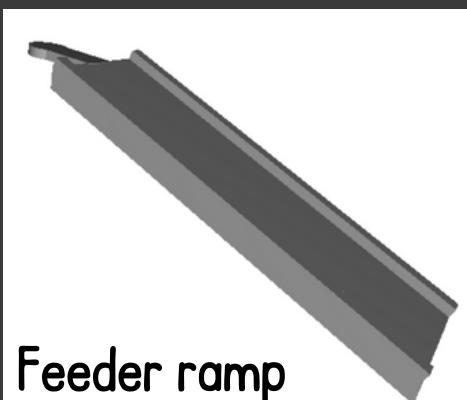
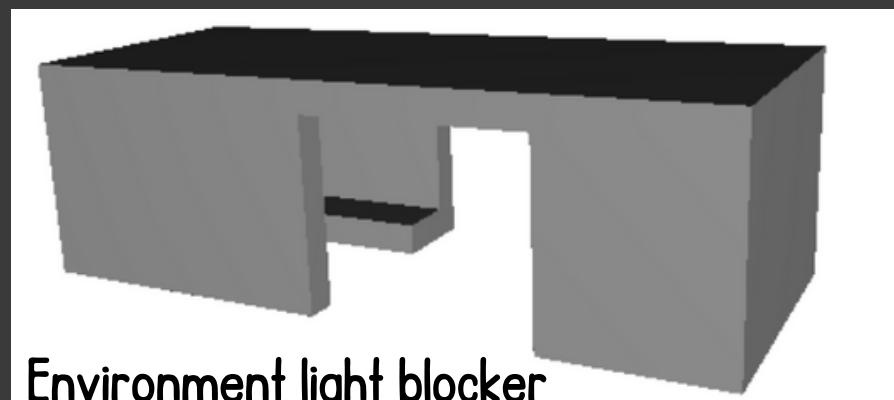
The Sorting System

- 2 ARDUINOS
- RASPBERRY PI
- LUGGAGE FEEDER
- FEEDING CYLINDER
- CONVEYOR BELT
- ROTARY ENCODER
- LIGHT RESISTOR
- RGB SENSOR
- INFRASTRUCTURE PIECES
- SORTED BAGGAGE STORAGE
- START, STOP, MANUAL OVERRIDE BUTTONS

OUR ROUGH DEV IDEA

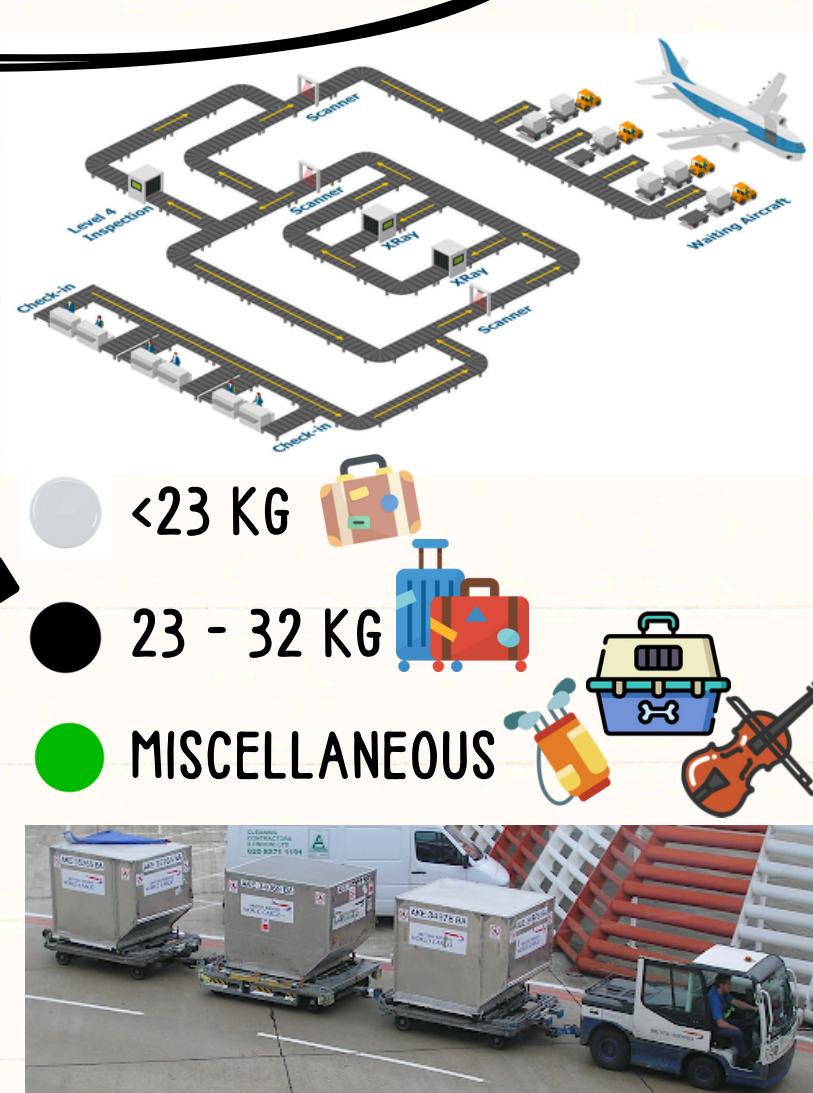


Our 3D-Printed Parts



Airport Scenario

- LAST SECTION
- LUGGAGE FROM ONE FLIGHT
- SORTING BY WEIGHT
- YELLOW BOXES REPRESENT TOW TRAILERS
- EASE LOADING OF CARGO
- MANUAL OVERRIDE
- WEBSITE



Systems Development Architecture

OUR REPOSITORY

This repository contains the code and website assets for our RPi and 2 Arduinos for the sorting and additional functionality detailed in our design documents to be possible.

How to run:

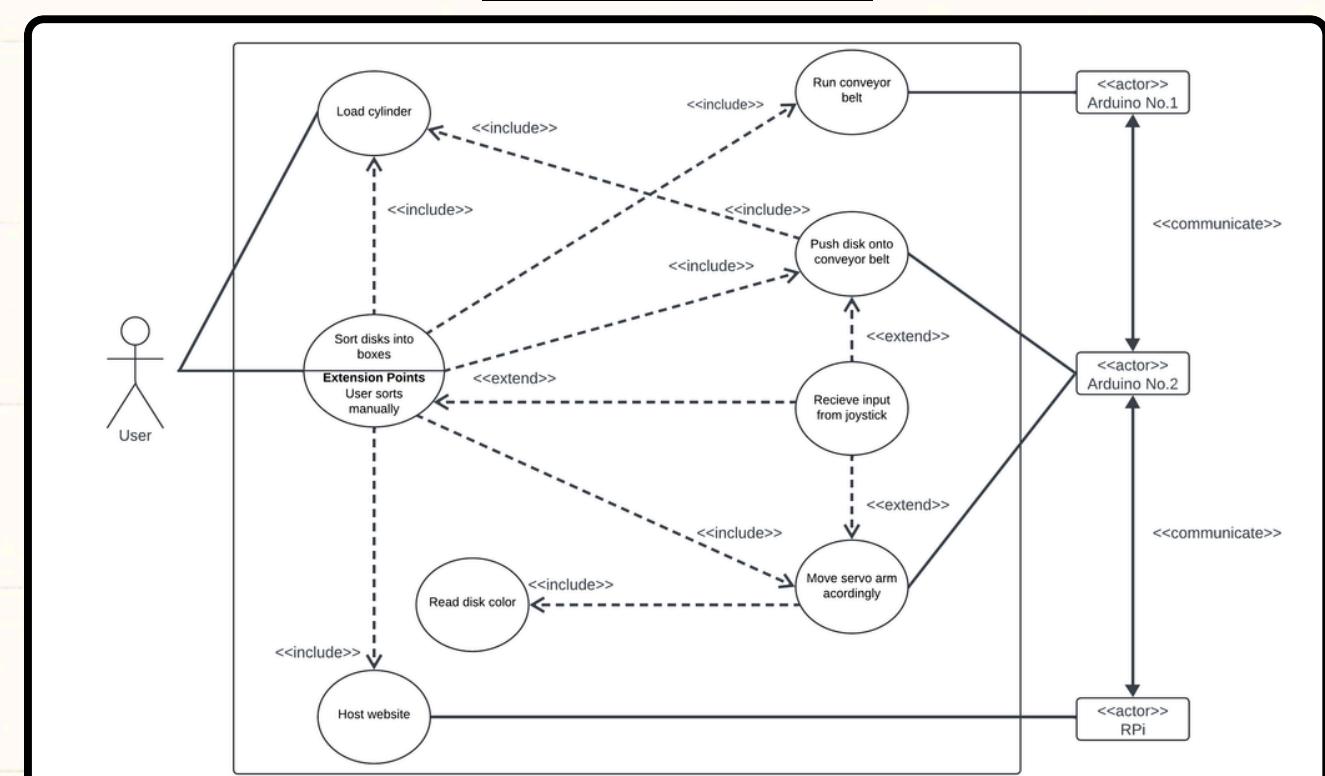
Run the Python script "Sorting_WebServer" on the Pi and Arduino to view the website being tested. Run the file "Sorting_Arduino" in the directory "ArduinoCode" on the sorting Arduino and "Conveyer_Motor" in the same directory on the conveyor belt driving Arduino. Ensure that both Arduinos are connected to the USB ports on the Pi.

See the user manual for more information on operating the robot.

Directories:

- Arduino: This directory contains code from online to develop our robots' functionality (e.g., Test and Tutorials, along with the code we are currently using or have used in the past in the Code, sub-directory. All the operational code was written by Ignas and Stephan).
- Website: This directory contains the website assets that make up the website.
- Webserver: This directory contains the website assets that make up the website. Written predominantly by Luis and Vicente, with additions written collaboratively.
- Screenshot: This directory contains screenshots of the website and the Arduino and Raspberry Pi, and website. Python is used to communicate through serial to and from the Arduino and PHP is used to communicate between the Python script and the JavaScript in the frontend. Written predominantly by Norbert, with additions written by Stephan and Ignas.

USE CASE DIAGRAM



FULL DECISION DOCUMENT

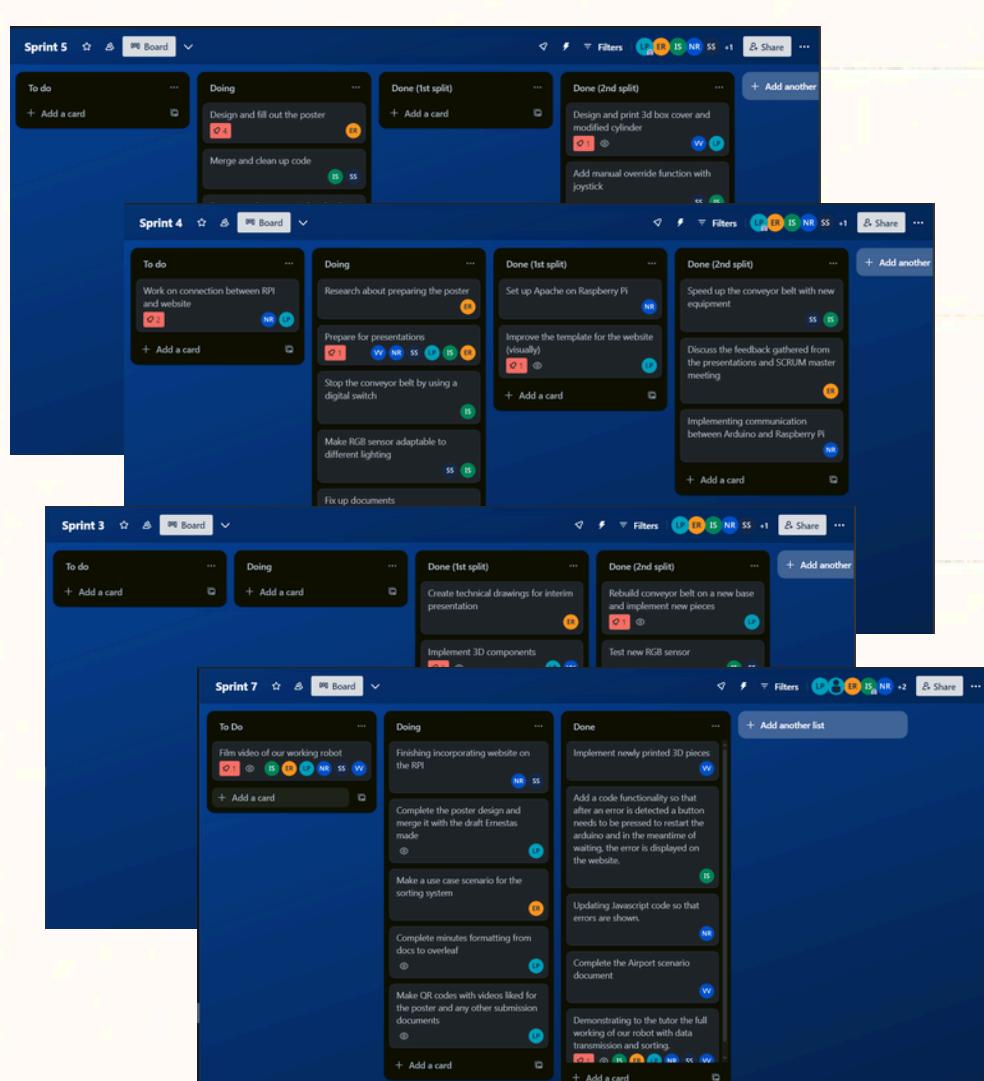


OUR MAIN DECISIONS

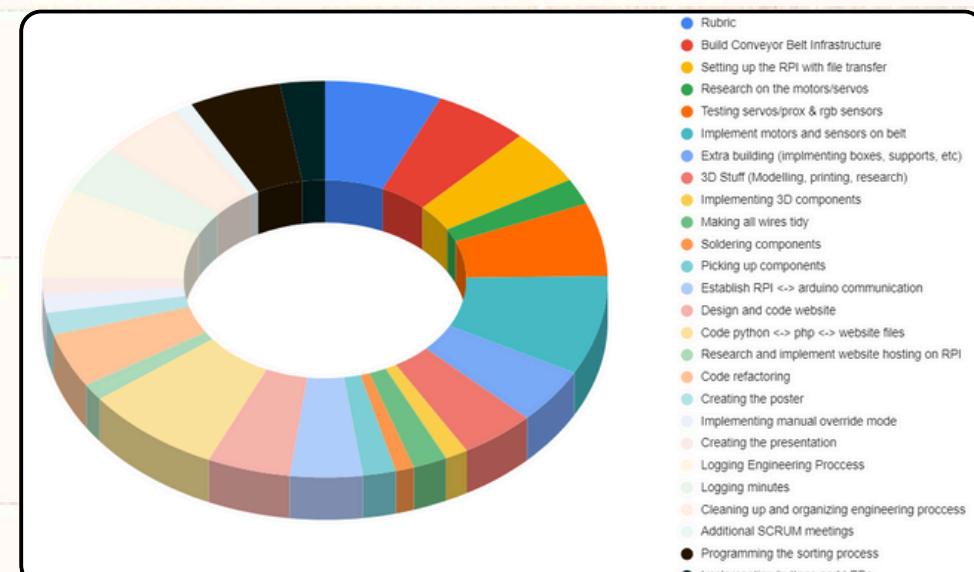
Decision taken	Purpose	Sprint N°
Use RGB sensor (purchased from Timytronics)	Light sensor provided does not allow our robot to have the desired functionality	1
Use a rotary encoder to detect if conveyor belt is stuck/stopped running	Proximity sensor was inaccurate and inconsistent	1
Only use one arm to place disks in boxes	More efficient and less error prone	2
Use a second arduino (purchased from Timytronics)	One arduino will be used to drive the conveyor belt and the other one will be used to control servos and serve motors	2
Design and print 3D cylinder and ramp	We want to have a fully automated system which feeds disks onto our conveyor belt	3
Design and print a box to enclose RGB sensor	Cylinder box was not very effective and disks often got stuck while going through it. 3D box will provide greater enclosure and will exactly fit the conveyor belt's dimensions	5

Documentation and Time Management

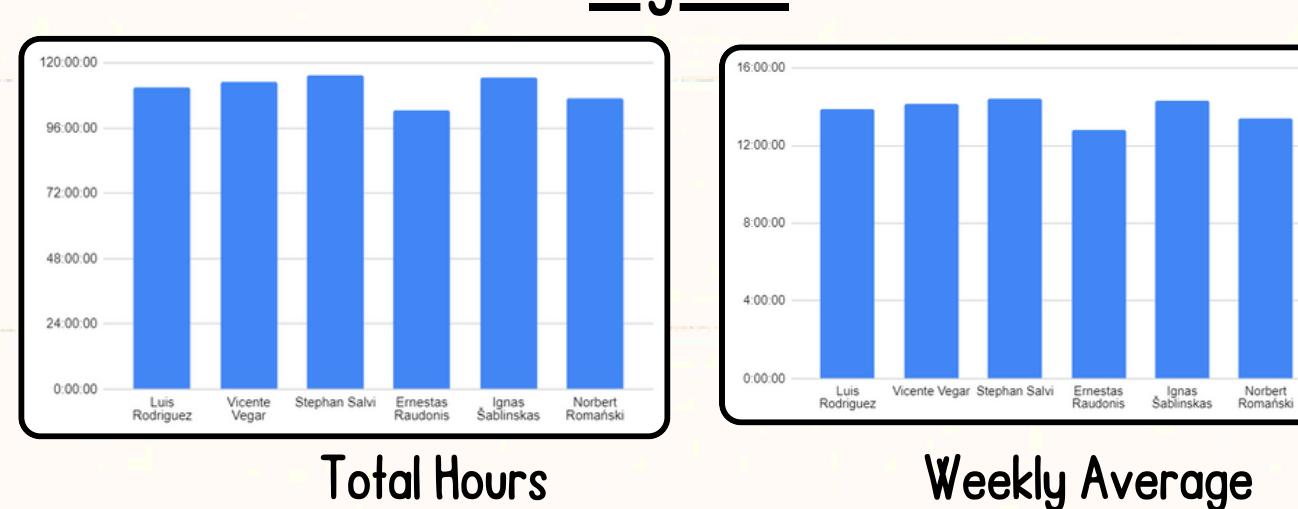
Trello Boards



Our task distribution



Logbook



ERROR HANDLING

CONVEYOR BELT IS STUCK!

- ROTARY ENCODER DETECTS THAT THE CONVEYOR BELT WAS NOT MOVING FOR 3 SECONDS AND SENDS THE INFORMATION TO RASPBERRY PI



LUGGAGE WAS EXPECTED BUT IS MISSING

- RGB SENSOR DOES NOT DETECT ANY CHANGE ON THE CONVEYOR BELT AND REPORTS THE ERROR TO RASPBERRY PI



RGB SENSOR ISSUES

- RGB SENSOR REPORTS EXTREMELY LOW VALUES TO RASPBERRY PI INDICATING THAT THE SENSOR IS BLOCKED OFF OR LIGHTNING CONDITIONS ARE TOO DARK



HOW DO WE HANDLE ERRORS?

- ONCE THE ERROR IS DETECTED GREEN LED TURNS OFF AND RED LED TURNS ON. IN ADDITION, THE RASPBERRY PI REPORTS THE SPECIFIC ISSUE ON THE WEBSITE INTERFACE.

