Functional Specifications

Project	Usine Robots		
Creation Date	*03/03/2025*	**Version**	*0.0.1*
Domain and Objective of the System	Embedded Systems - Industrial Robots		
	The system is supposed to automate triage, transport and distribution processes of industrial objects. It is composed of multiple subsystems: 1. *Robot arm - Triage: Responsible for sorting and distributing objects to the transport layer* 2. *Turtle Bot: Transport layer. Moves the objects from one end (Robot arm - Triage) to the other (Robot arm - Deposit).* 3. *Robot arm - Deposit: Responsible for distributing the objects to their designated deposit area.*		
Description of the Actors	[Robot Arm]: Embedded system connected via ROS X to the Coordinator. [Turtle bot]: Turtle bot 2 controlled by a raspberry Pi 3 connected via ROS 2 Iron to the Coordinator. The robot has access to a Kinect camera as well as two motors. [Coordinator]: Central system responsible for directing and handling communications between both Robot Arms and the Turtle bots. [Operator]: Employee that oversees the operation. [Object Source]: Entity that holds the unsorted triable objects. [Intruder]: Any unknown entity inside the system's designated space		

Scenario ID	*SC0*	Title	Setup
Actors	**User:** Operator		
	Preconditions: The operator has loaded the object source; The operator turns on all of the systems		
	Postconditions:		
	Triggers:		
Steps	**Iteration Sequence**		
	[MAIN FLOW] 1. The *operator* executes the setup program on the *coordinator* 2. The *coordinator* attempts to connect to the *Turtle bot* via ssh [A1] 3. The *coordinator* executes the setup program in the *Turtle bot* 4. The *coordinator* attempts to connect to the *robot arms* via ssh [A1] 5. The *coordinator* executes the setup program in the Robot Arm - Triage. 6. The *coordinator* executes the setup program in the Robot Arm - Deposit containing the deposits information (whether deposit A should take objects of color 'red' or 'blue', for example).		
	[ALTERNATIVE FLOW] *[A1]* The *Coordinator* failed to connect to a *remote system* 1. The *coordinator* attempts to connect to the *remote system* via ssh again [B1][E1] 2. If succeeded, return to step 3 of *[MAIN FLOW]*		
	[EXCEPTION FLOW] *[E1]* The *Coordinator* cannot to connect to a *remote system* 1. The *coordinator* terminates		
	[BUSINESS RULE] *[B1]* The Coordinator re-attempts a connection at most 3 times		

Scenario ID	*SC1*		Title	Give permission to take object from source
Actors	**Systems:** Coordinator			
	Target Subsystem: Robot arm - Triage			
	Preconditions:			
	Postconditions:			
	Triggers: Message of <u>inform of arrival</u> from Turtle Bot to the coordinator			
Steps	**Iteration Sequence**			

	[MAIN FLOW]
	   1. The *coordinator* sends a message via ROS to the *target subsystem* containing the position of the turtle bot, the color of the object it will hold and the amount of objects it can hold.
	   2. The *target subsystem* sends an acknowledge message via ROS to the *coordinator*.			

	[ALTERNATIVE FLOW] *[A1]* 3.			
	[EXCEPTION FLOW] *[E1]* 2.			
	[BUSINESS RULE] *[B1]*			

Scenario ID	*SC2*		**Title**	Inform objects were put onto Turtle Bot
Actors	**System:** Robot arm - Triage, Turtle Bot			
	Target Subsystem: Coordinator			
	Preconditions: <u>Give permission to take object from source</u> has happened			
	Postconditions:			
	Triggers:			
Steps	**Sequence des Iterations**			

	[MAIN FLOW]
	   1. The *robot arm* moves towards the position informed by the permission message.
	   2. For each object in the object source area:
	      a. The *robot arm* identifies the object's silhouette from the object source area.
	      b. The *robot arm* takes the object from the object source area.
	      c. The *robot arm* identifies the color of the object to be sorted. [A1]
	      d. The *robot arm* identifies the area on top of the object where there is available space. [B1]
	      e. The *robot arm* puts the object in the available space on top of the turtle bot corresponding to the identified color.
	      f. The *robot arm* sends a message via ROS to the *coordinator* informing an object of the specific color was put on top of the Turtle Bot.
	   3. The *robot arm* sends a message via ROS to the *coordinator* informing all objects were put on top of the Turtle Bot.			

	[ALTERNATIVE FLOW]
	[A1] The robot arm identifies a color that does not correspond to any color present in the permission message
	   1. The robot arm restarts the identification method. [E1]
	   2. The robot arm resumes Main flow 2.d			

	[EXCEPTION FLOW]
	[E1]
	   1. If the identification method still fails to match the object's color to the color present in the message from the coordinator:
	      a. Place the object back on the deposit area.
	      b. Exit scenario.			

	[BUSINESS RULE] *[B1] The turtle bot* has enough space to hold all objects informed by the permission message with a error margin of 2cm for each object.			

Scenario ID	*SC3*	Title	Inform objects were put on the deposit.
Actors	**System:** Robot arm - Deposit, Turtle Bot		
	Target Subsystem: Coordinator		
	Preconditions: *Inform arrival of Turtle bot* has happened		
	Postconditions:		
	Triggers:		
	Sequence des Iterations		

Steps	*[MAIN FLOW]*
	1. The *robot arm* moves towards the position informed by the permission message.
	2. For each object on top of the turtle bot:
	    a. The *robot arm* takes the object from the turtle bot
	    b. The robot arm puts the object on the specific deposit.
	    c. The robot arm sends a message via ROS to the coordinator informing the object was put on the specific deposit.
	3. The robot arm sends a message to the coordinator informing all objects were sorted.		

	[ALTERNATIVE FLOW] *[A1]* 3.		
	[EXCEPTION FLOW] *[E1]* 2.		
	[BUSINESS RULE] *[B1]*		

Scenario ID	SC4		Title	Inform of arrival
Actors	**System:** Turtle bot			

	Target Subsystem: Coordinator
	Preconditions: The turtle bot has finished its route
	Postconditions:
	Triggers:

Steps	**Sequence des Iterations**
	[MAIN FLOW] 1. The *Turtle bot* checks it's within acceptable bounds of the final position once it reaches the cushion area.[A1][B1] 2. The *Turtle bot* sends a message to the coordinator with its position
	[ALTERNATIVE FLOW] *[A1]* The *Turtle bot* is not within acceptable bounds 1. move until error in x-axis direction is below the error [B1] 2. move until error in y-axis direction is below the error [B1]
	[EXCEPTION FLOW]
	[BUSINESS RULE] *[B1]* The Turtle bot should reduce its speed until full stop the closer it is from the cushion area, starting from a distance of 30cm to the cushion area.

Scenario ID	*SC5*		Title	Transport Batch

Actors	**System:** Coordinator
	Intruder

	Target Subsystem: Turtle bot
	Preconditions:
	Postconditions:
	Triggers:

	Sequence des Iterations
Steps	*[MAIN FLOW]* 1. The *coordinator* sends the route to the *turtle bot* 2. The *turtle bot* inverts its direction 3. The *turtle bot* starts the route [B1][A1]
	[ALTERNATIVE FLOW] *[A1]* The *turtle bot* detects an intruder 1. The *turtle bot* stops while it's still detecting an intruder 2. else return to step 3
	[EXCEPTION FLOW]
	[BUSINESS RULE] *[B1]* The turtle bot should follow its reference route with a position error margin of 10cm

Non Functional Requirements			
ID	Description	Priority	ID of Corresponding Functional Requirement
NFR1	The arm must be able to distinguish between colors with a difference of 1 ΔE *	**High**	SC2
NFR2	The arm must be able to pick up objects with dimensions of at most X SI	**High**	SC2
NFR3	The arm must be able to handle a maximum load of Y SI	**Medium**	SC2
NFR4	The arm must take at most T second(s) for each action	**Medium**	SC2
NFR5	The turtle must be at most X SI from the arm robot in the deposit zone	**High**	SC5
NFR6	The turtle robot must be able to carry X objects	**Medium**	SC5
NFR7	The turtle must have a battery life of X hours	**Low**	—
NFR8	The system must have an object flow of at least X objects / hour	**Medium**	SC2, SC5, SC3
NFR9	The 2 robot arms must be at least X meters apart	**Low**	SC5