**“Use-case Description”**

**Diagram

Description automatically generated**

**“The user actor”**

**1- Setup use-case:**

1. **The use case ID & Name:** 1 to setup position.
2. **The initiator**: the user.
3. **Goal:** to determine the field in which the robot will operate.
4. **Pre-condition:** none.
5. **Post-condition:** the field dimension saved in robot database.
6. **Main scenario:** The user setup the position of his\her field.
7. **Non-Functional Req.**

* Response Time
* Memory-Capacity

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**2- Turn on use-case:**

1. **The use case ID & Name:** 2 to turn on.
2. **The initiator:** the user.
3. **Goal:** to make the robot do his tasks.
4. **Pre-condition:** the robot must be setup.
5. **Post-condition:** the robot is allowed to do his tasks.
6. **Main scenario:**
7. The user must setup the robot at first.
8. Then allow it to do his tasks by using turn on.
9. **Non-Functional Req.**

* Response time

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**3- Turn off use-case:**

1. **The use case ID & Name:** 3 to turn off.
2. **The initiator:** the user.
3. **Goal:** to stop running the robot.
4. **Pre-condition:** the robot must still be running.
5. **Post-condition:** the robot is shut down.
6. **Main scenario:**
7. The robot is operating.
8. Then the user will turn it off.
9. **Non-Functional Req.**

* Response time

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Diagram

Description automatically generated**

**4- intra-row weeding use-case:**

1. **The use case ID & Name:** 4 to intra-row weeding.
2. **The initiator:** the Blades.
3. **Goal:** to remove weeds from the field in parallel
4. **Pre-condition:** the robot must still be running.
5. **Post-condition:** All weeds in the field have been removed.
6. **Main scenario:**
7. The robot must still be running.
8. It must make sure it is in the field.
9. then it detects the weeds.
10. Calculate the distance between it and the weeds.
11. Determine the intra-row weeding.
12. Extend blades and walk forward.
13. Intra-row weeding
14. **Failed scenario:**
15. The robot must still be running.
16. It must make sure it is in the field.
17. If it was outside the field, it must notify the user and stop working.
18. **Dependency:**

* Extend blades.
* Detect the intra-row weeding
* Walk forward

1. **Non-Functional Req.:**

* High Performance
* Accuracy
* Safety
* Concurrency

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Diagram

Description automatically generated**

**5- Display data use-case:**

1. **The use case ID & Name:** 5 to Display data.
2. **The initiator:** the Display screen.
3. **Goal:** to display analysis data about the soil to the user.
4. **Pre-condition:** the robot must move.
5. **Post-condition:** the user know data about his soil such as soil-type, etc.
6. **Main scenario:**
7. The robot must still be running.
8. the robot capture image for the field by its cameras.
9. The robot processes the images.
10. and gather data about the soil.
11. **Dependency:**

* Data analysis

1. **Non-Functional Req:**

* Accuracy
* Concurrency
* Quality
* Memory-capacity

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Diagram

Description automatically generated**

**6- Navigate on the headland use-case:**

1. **The use case ID & Name:** 6 to navigate on the headland.
2. **The initiator:** the robot.
3. **Goal:** to stop his task “intra-row weeding” until find the next weed row.
4. **Pre-condition:** the robot must still be running.
5. **Post-condition:** find the next weed row to start weeding again.
6. **Main scenario:**
7. The robot must still be running.
8. Get his position in the field using GPS and some calculations.
9. Detects the headland.
10. Calculate the distance between it and the headland.
11. If the robot is close to the headland, it stop moving and retract his blades.
12. Then detect the next intra-row weeding to rotate in its direction.
13. **Failed scenario:**
14. The robot must still be running.
15. Get his position in the field using GPS and some calculations.
16. Detects the headland.
17. Calculate the distance between it and the headland.
18. If the robot is far away from the headland.
19. Then it completes his task “intra-row weeding”.
20. **Dependency:**

* Detect the headland
* Rotation

1. **Non-Functional Req.**

* Accuracy
* Flexibility
* Performance

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Diagram

Description automatically generated**

**7- Detect battery low status use-case:**

1. **The use case ID & Name:** 7 to Detect Battery Low.
2. **The initiator:** The Battery.
3. **Goal:** to make sure that the system in his better state and complete his operation.
4. **Pre-condition:** the robot must still be running.
5. **Post-condition:** knowing the status of battery.
6. **Main scenario:**
7. The robot must still be running.
8. The robot gets info. about his battery.
9. If the battery is at good state.
10. Then it completes its operations.
11. **Failed scenario:**
12. The robot must still be running.
13. The robot gets info. about his battery.
14. If the battery is low.
15. Then the robot notifies the user about the status of the battery.
16. **Dependency:**

* Get status

1. **Non-Functional Req.**

* Lifetime
* Accuracy

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**7- Notify use-case:**

1. **The use case ID & Name:** 8 to Notify the user.
2. **The initiator:** the robot.
3. **Goal:** is to send notification to the user.
4. **Pre-condition:** the robot must still be running.
5. **Post-condition:** the user get the notification.
6. **Main scenario:**
7. The robot must still be running.
8. If the battery is low, the robot go outside the field or the robot covers the whole field and finish his task.
9. Then it notifies the user.
10. **Dependency:**

* Detect Battery low

1. **Non-Functional Req.**

* Response time
* A start and end time of task
* Throughput