

**SOFTWARE REQUIREMENT SPECIFICATION**

FORESTALLING OF FISHES

|  |  |
| --- | --- |
| VERSION: 0.1 | REVISION DATE: 24.03.2020 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Approver Name** | **Title** | **Signature** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

[Forestalling of Fishes] ……………………………………………………………………………………………………………………

1. Product Description……………………………………………………………………………………………………………………..

1.1 Purpose………………………………………………………………………………………………………………………..

1.2 Scope……………………………………………………………………………………………………………………………

1.3 Stakeholders and Users………………………………………………………………………………………………..

1.4 Assumptions………………………………………………………………………………………………………………..

1.5 Constraints…………………………………………………………………………………………………………………..

2. Functional Requirements…………………………………………………………………………………………………………….

2.1 Requirement X…………………………………………………………………………………………………………….

2.1.1 Requirement X1…………………………………………………………………………………………..

2.1.1.1 Requirement X1x………………………………………………………………………….

3. Hardware requirements……………………………………………………………………………………………………………..

3.1 Circuit diagram……………………………………………………………………………………………………………..

4. Interface Requirements……………………………………………………………………………………………………………….

4.1 User Interfaces……………………………………………………………………………………………………………..

4.2 Hardware Interfaces………………………………………………………………………………………………………

4.3 Software Interfaces……………………………………………………………………………………………………….

5. Use Case Model…………………………………………………………………………………………………………………………...

5.1 Use Case Diagram………………………………………………………………………………………………………….

5.2 Use Case Descriptions……………………………………………………………………………………………………

6. Glossary………………………………………………………………………………………………………………………………………..

7. References…………………………………………………………………………………………………………………………………….

8. Revision History…………………………………………………………………………………………………………………………….

**1. Product Description**

**1.1 Purpose**

The main purpose of this Forestalling of fishes’ is that it provides provision to fishermen to prevent the escape of fishes.Fish escaping is a major issue for fishermen in rural areas during monsoons, due to the overflowing of ponds. This leads to a less profit margin for fishermen.

**1.2 Scope**

The scope of our project is to build a system whose objective and goal is to use a water level sensor to indicate the level of water (high or low). Meshed fences around the pond rise or descend accordingly, with the help of the rack-pinion to prevent fishes from escaping.

**1.3 Stakeholders and Users**

This project is a prototype for the forestalling of fishes’ system. This project is useful for Fishermen who practice fish farming in ponds. This product is used to prevent fishes from escaping the pond in rural areas during monsoons due to the overflowing of the ponds.

**1.5 Constraints**

Limited budget of the users.

**2. Functional Requirements**

**2.1 Sensor module configuration**

**2.1.1 Establishing connection between Sensor Modules and Main Control Centre using three way handshaking**

i) Main control centre will send a message to check each sensor module is up or not.

ii) Receiving the acknowledgement from each sensor modules it sends a reply that it received the acknowledgement and the connection is established.

**2.1.2 Updating the MAC address of each sensor module in the look up table**

After receiving the acknowledgment address of each sensor module is updated in the look up table.

**2.1.3 Setting the threshold value**

Threshold level value for the sensor module is set.

**2.2 Water Sensor raises alarm**

Water sensor senses the ascend or descend of water level beyond threshold level.

**2.2.1 Raises alarm indicating high water level.**

If the water level rises beyond threshold level, the water sensor will sense it and raise an alarm indicating the high water-level.

**2.2.2 Raises alarm indicating low water level**

If the water level descends below threshold level, the water sensor senses it and will raise an alarm indicating the low water- level.

**2.3 Updating the Status**

**2.3.1 Sends MAC address of the sensor module after receiving alarm**

If System raise a high alarm or low alarm, System updates the status that a high or low alarm was raised and also sends the MAC address of sensor modules (i.e. location) and code value 1 (for rising the net) or code value 0 (for lowering the net) to the main control Centre.

**2.4 Main control centre Receives updated Status**

Main control centre receives updated status .

**2.4.1 Receives the updated status.**

Main Control Centre receives the MAC address and the code value of the sensor rising the alarm as to raise the net or lower the net.

**2.4.2 Makes the decision to rise or low the net**

If Main control centre receives code value is 1 from any one of the sensor module it will prepare instructions for raising the net all around the pond and for 0 it will prepare instructions for lowering the net.

**2.5 Sending Instructions to Sensor modules**

The net is controlled by the sensors considering the level of the water.

**2.5.1 Starts Sending the instructions**

After making the decision main control centre sends instruction to the sensor modules. Sensor module will then receive the instruction.

**2.5.2 Confirming that sensor modules received the instructions**

Main control Sensor follows a handshaking protocol. Sensor will then send the acknowledgement to the Main Control centre.

**2.6 Controlling the net**

The net is controlled by the sensors considering the level of the water.

**2.6.1 High water level**

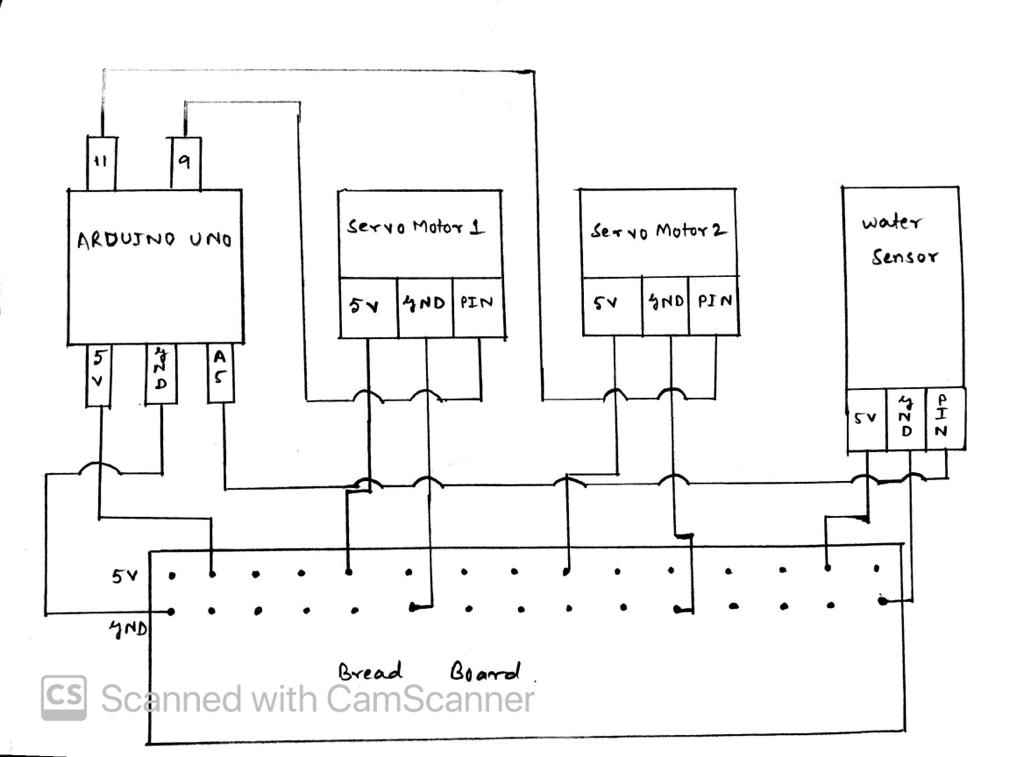
After receiving the instructions from main control centre , motor starts rotating in the clockwise direction and the net thus starts rising.

**2.6.2 Low water level**

After receiving the instructions from main control centre, motor starts rotating in the anti- clockwise direction and the net thus starts going down.

**3. Hardware Requirements**

**3.1 Circuit diagram**



**4. Interface Requirements**

**4.1 User Interfaces**

* Front-end software: Arduino 1.8.5 version

**4.2 Hardware Interfaces**

* Water Level Sensor: Robodo SEN18
* Servo Motor: Tower Pro S490 9g Mini Servo
* Rack-pinion
* Breadboard
* Display: Standard Output Display

**4.3 Software Interfaces**

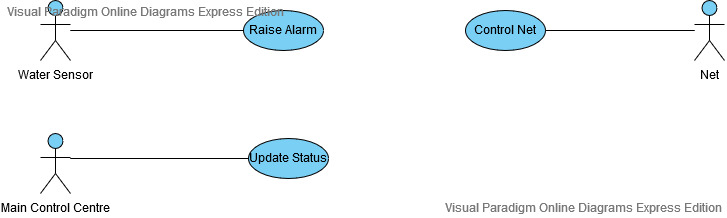
Following are the software used for the Forestalling of fishes’

|  |  |
| --- | --- |
| **Software used** | **Description** |
| Operating system | We have chosen Windows operating system for its best support and user-friendliness. |
| Arduino 1.8.5 version | To implement the Forestalling of fishes’ project we have used Arduino Uno for its more interactive support |

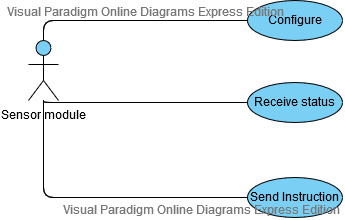
**5. Use Case Model**

**5.1 Use Case Diagram**

**1.SENSOR MODULE**

****

**2.MAIN CONTROL CENTRE**

****

**5.2 Use Case Descriptions**

**5.2.1 Description for Use Case CONFIGURE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID: | Configure | | | |
| Use Case Name: | Configure | | | |
| Created By: | Suraj Ray | | Last Updated By: |  |
| Date Created: | 11.04.2020 | | Date Last Updated: | 11.04.2020 |
| Actor: | | Sensor Module | | |
| Description: | | Configuration | | |
| Preconditions: | |  | | |
| Postconditions: | |  | | |
| Priority: | |  | | |
| Frequency of Use: | |  | | |
| Trigger: | |  | | |
| Flow of Events: | | STEP 1: In the beginning of the process, connection between the sensor modules and main control centre is established using three way handshaking.  STEP 2: After receiving the acknowledgement, MAC address of each sensor module is updated in the look up table according to their respective locations with respect to the pond.  STEP 3: Threshold level value for the sensor module is set. | | |
| Alternative Flows: | |  | | |
| Exceptions: | |  | | |
| Includes: | |  | | |
| Special Requirements: | |  | | |
| Assumptions: | |  | | |
| Notes and Issues: | |  | | |

**5.2.2 Description for Use Case RAISE ALARM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID: | Raise alarm | | | |
| Use Case Name: | Raise alarm | | | |
| Created By: | Yajushi Dey | | Last Updated By: |  |
| Date Created: | 11.04.2020 | | Date Last Updated: | 11.04.2020 |
| Actor: | | Water Sensor | | |
| Description: | | Raise Alarm | | |
| Preconditions: | |  | | |
| Postconditions: | |  | | |
| Priority: | |  | | |
| Frequency of Use: | |  | | |
| Trigger: | |  | | |
| Flow of Events: | | STEP 1: If water level rises beyond the threshold value, water sensor senses high water-level.  STEP 2: System will then raise an alarm indicating high water level.  STEP 3: If water level descends below the threshold value, water sensor senses low water-level.  STEP 4: System will then raise an alarm indicating low water level. | | |
| Alternative Flows: | |  | | |
| Exceptions: | |  | | |
| Includes: | |  | | |
| Special Requirements: | |  | | |
| Assumptions: | |  | | |
| Notes and Issues: | |  | | |

**5.2.3 Description for Use Case UPDATE STATUS**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | Update Status | | |
| Use Case Name: | Update Status | | |
| Created By: | Shyanka Basak | Last Updated By: |  |
| Date Created: | 20.04.2020 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | Main Control Centre |
| Description: | Updates Status |
| Preconditions: | Alarm has been raised |
| Postconditions: | Main control centre checks the look up table |
| Priority: |  |
| Frequency of Use: |  |
| Trigger: |  |
| Flow of Events: | STEP 1: Sensor module senses the high alarm.  STEP 2: System updates the status that a high alarm was raised and also updates the status in the look up table.  STEP 3: Sensor module senses the low alarm.  STEP 4: System updates the status that a low alarm was raised and also updates the status in the look up table. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**5.2.4 Description for Use Case RECEIVE STATUS**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | Receive Status | | |
| Use Case Name: | Receive Status | | |
| Created By: | Somshubhra Roy | Last Updated By: |  |
| Date Created: | 24.04.2020 | Date Last Updated: | 24.04.2020 |

|  |  |
| --- | --- |
| Actor: | Sensor Module |
| Description: | Receive Status |
| Preconditions: |  |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: |  |
| Trigger: |  |
| Flow of Events: | STEP 1:Sensor module updates the look up table for a high alarm.  STEP 2: System will then receive the update status.  STEP 3: Sensor module updates the look up table for a low alarm.  STEP 4: System will receive the updated status and prepare the instructions accordingly. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**5.2.5 Description for Use Case SEND INSTRUCTIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | Send Instruction | | |
| Use Case Name: | Send Instruction | | |
| Created By: | Smita Bandyopadhyay | Last Updated By: |  |
| Date Created: | 24.04.2020 | Date Last Updated: | 24.04.2020 |
|  |  |  |  |

|  |  |
| --- | --- |
| Actor: | Sensor Module |
| Description: | Send Instrstruction |
| Preconditions: | Main Control Centre has received alarm and decision is already made whether sensor module  has to raise or lower the net. |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: |  |
| Trigger: |  |
| Flow of Events: | STEP 1:System sends instruction to the sensor module.  STEP 2: Sensor module will then receive the instruction.  STEP 4: Sensor module will then send the acknowledgement after receiving the instruction.. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**5.2.6 Description for Use Case CONTROL NET**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | Control net | | |
| Use Case Name: | Control net | | |
| Created By: | Siddharth Chand Ramola | Last Updated By: |  |
| Date Created: | 20.04.2020 | Date Last Updated: | 20.04.2020 |

|  |  |
| --- | --- |
| Actor: | Net |
| Description: | Controlling raising or lowering of the meshed fence |
| Preconditions: | System receives instructions from the main control centre. |
| Priority: |  |
| Frequency of Use: |  |
| Trigger: |  |
| Flow of Events: | STEP 1 : Main control centre gives instructions to raise net.  STEP 2 : System raises net.  STEP 3: Main control centre gives instructions to lower net.  STEP 4: System lowers net. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |
|  |  |

**6. Glossary**

|  |  |
| --- | --- |
| **Data** | **Description** |
| Servo motor | A servo motor is an electrical device which can push or rotate an object with great precision. If motor is used in DC power then it is called a DC servo motor. The position of a servo motor is decided by electrical pulse and its circuitry is placed beside the motor. The Tower Pro S490 9g Mini Servo is 180-degree rotation servo |
| Rack-pinion | Pinion is attached with servo motor and rack is attached with meshed fence. A Rack-pinion is a type of linear actuator that comprises a circular gear engaging a linear gear |
| Arduino Uno | Arduino Uno is a microcontroller board with USB interface, 6 analog input pins, 14 I/O digital ports that are used to connect with external Electronic Circuits. Out of the 14 I/O ports, 6 pins can be used for PWM output. |
| Water Level Sensor | Robodo SEN18 Water Level Sensor is a Depth of detection Water Sensor for Arduino". It is used and interfaced with Arduino Uno. Operating Voltage: DC 3-5V, Operating Current: Less than 20 mA, Sensor Type: Analog, Sensor Dimensions: 60mm x 20mm |
| Breadboard | A breadboard is a rectangular plastic board with a bunch of tiny holes in it. These holes let you easily insert electronic components to prototype (meaning to build and test an early version of) an electronic circuit, like this one with a battery, switch, resistor, and an LED (light-emitting diode) |
| Meshed Fence | It is just like fishing nets used by fishermen for fishing. It is a device made from fibers woven in a grid-like structure. The meshes are formed by knotting a relatively thin thread. |

**7.References**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document No.** | **Document Title** | **Date** | **Author** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**8. Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Description** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |