**Smart Water Management System**

**Software Requirements Specification(SRS)**

Version 1.4



**Group Id: F2302894E5**

**Supervisor Name:** **Dr. Israr Ullah**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date (dd/mm/yyyy) | Version | Description | Author |
| 07//12/2023 | 1.0 | This IoT-based project aims to develop a smart water management system that automates the monitoring, filling, and pump operation process of a home water tank. The system utilizes sensors to continuously monitor the water level and automatically start and stop the water pump based on predefined thresholds. A companion mobile app allows users to monitor the current water level and manually control the pump. | BC200406015 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. Introduction
   1. Purpose
   2. [Scope (of the project)](#scope)
2. System Overview
   1. Project Objectives
   2. System Components
3. Functional Requirements
   1. For Automated System
   2. For Companion Mobile App
4. Non Functional Requirements
5. [Use Case Diagram](#UCD)
6. [Usage Scenarios](#UCS)
7. [Adopted Methodology](#Adopted)
8. [Work Plan (Use MS Project to create Schedule/Work Plan)](#Gantt)

**SRS Document**

1. Introduction

# The Smart Water Management System (SWMS) is an innovative and user-friendly solution designed to optimize water usage and provide peace of mind for homeowners. This system utilizes a network of sensors and a mobile app to monitor water levels, automate pump operation, and eliminate the need for manual intervention.

* 1. Purpose

The purpose of this project is to develop a smart IoT-based solution for automating water tank monitoring, filling, and pump operation. This will help to address common household challenges such as water overflow, pump damage, and manual labour.

* 1. [Scope of Project:](#scope)

[The scope of this project is to develop an IoT-based solution for automating the water tank monitoring, filling, and pump operation process. The system will consist of two main components: a hardware component and a IoT Project component.](#scope)

[The hardware component will include an ESP32 development board, an ultrasonic sensor, a water flow sensor, a relay module, and a 5V water pump. The IoT Project component will include a Blynk mobile app and an ESP32 Micro-Controller.](#scope)

1. **System Overview**

The Smart Water Management System, comprises a network of interconnected components that work together seamlessly to achieve its intended goals. The core of the system lies in the ESP32 development board, which acts as the central processing unit. This board collects data from strategically placed sensors, namely an ultrasonic sensor at roof tank for measuring water level and a water flow sensor near ground tank for monitoring water flow. Based on the acquired data, the ESP32 board utilizes a relay module to control the operation of the water pump.

The system's functionality extends beyond hardware by incorporating a user-friendly mobile application. This app provides a real-time view of the water level in the tank, allowing users to remotely monitor the system's performance. Additionally, the app offers the ability to manually turn the pump on or off, ensuring complete control over water resource management.

* 1. **Project Objectives**

The main objectives of this project are to:

* Create a system that can continuously monitor the water level in a roof tank.
* Automate the process of filling the roof tank when the water level drops below a certain threshold.

# Prevent water overflow by automatically stopping the pump when the tank is full.

# stop the pump when there is no water flow (indicating that the ground tank is empty).

* Develop a mobile app that allows users to monitor and control the system remotely.
  1. **System Components**

The system will consist of the following components:

* **Hardware:**
* **ESP32 Development Board**: Acts as the brain of the system.
* **Ultrasonic Sensor:** Continuously measures the water level in your rooftop tank.
* **Water Flow Sensor:** Detects water flow in the pipe leading to ground tank.
* **Relay Module:** Safely controls the 220V water pump
* **5-volt Water pump:** for prototype
* **Software:**
* Firmware for the microcontroller to manage sensor data and pump control
* Mobile application for user interaction and remote monitoring

1. **Functional and non-Functional Requirements:**

**a. Functional Requirements**

* Automated System
* The automated system will have the following functional requirements:
* Continuously monitor the water level in the roof tank using an ultrasonic sensor.
* Automatically start the water pump when the water level falls below a predefined threshold in the roof tank.
* Monitor water flow in the pipe using a water flow sensor and stop the pump when there is no water flow (indicating that the ground tank is empty).
* Automatically stop the water pump when the water level rises above a predefined threshold to prevent water overflow.
* **Companion Mobile App**
* The companion mobile app will have the following functional requirements:
* Display the current water level in the roof tank.
* Allow users to manually turn on/off the water pump from the app.

**b. Non-Functional Requirements**

The system will have the following non-functional requirements:

* Reliability: The system will be reliable and operate without failures for extended periods of time.
* Accuracy: The water level measurements will be accurate.
* Responsiveness: The system will respond to changes in water level and pump status quickly.
* User-friendliness: The mobile app will be easy to use and navigate.

1. **Use Case Diagram(s):**

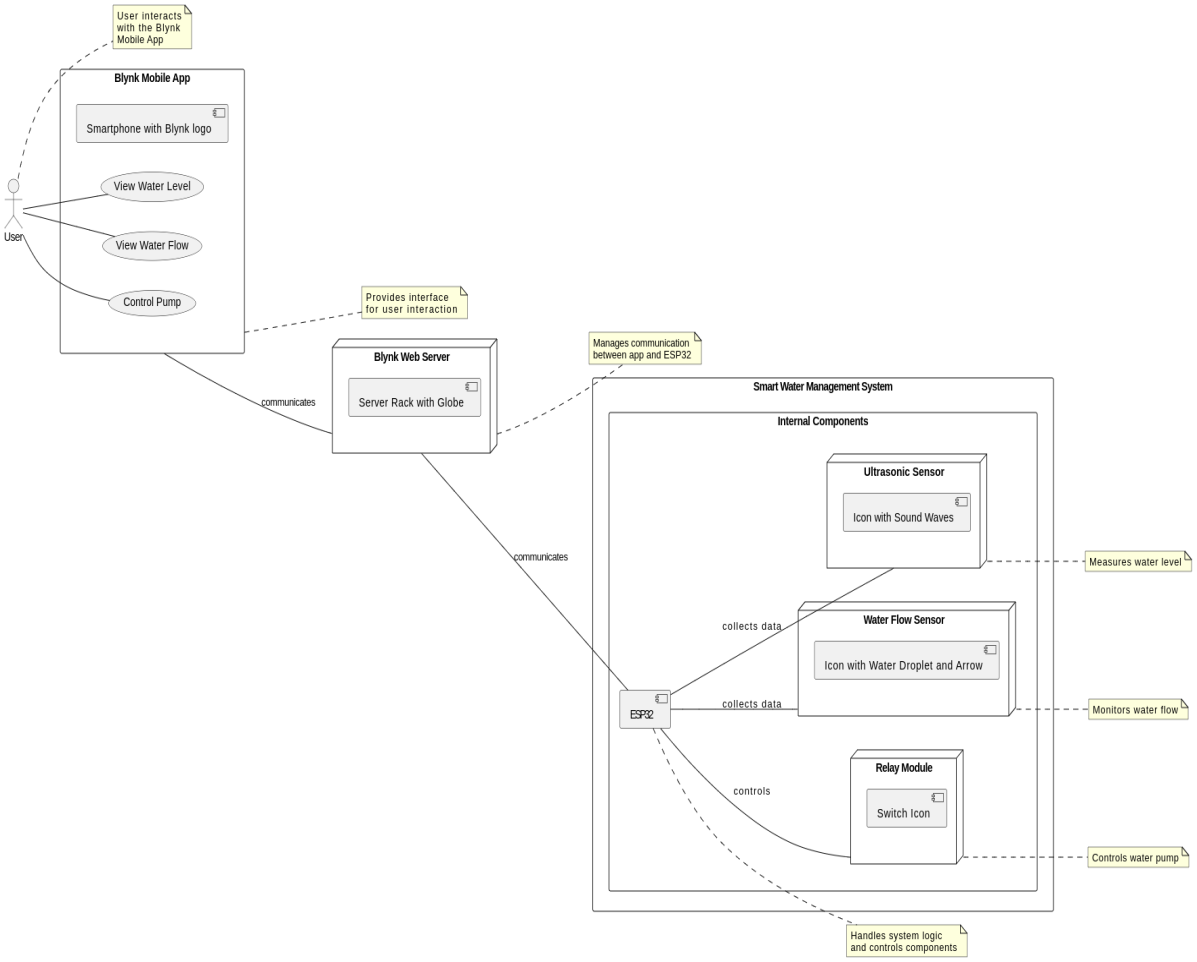


Figure 1

1. **Usage Scenarios:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Use Case Title | Use Case Id | Actions | Description | Alternative Paths | Pre-Conditions | Post-Conditions | Author | Exceptions |
| Monitor water level | UC-1 | 1. The ultrasonic sensor measures the water level in the roof tank. | 2. The ESP32-Micro-Controler reads the sensor data and sends it to the Blynk mobile app. | N/A | The water level is displayed on the mobile app. | The water level is displayed on the mobile app. | BC200406015 | N/A |
| Start water pump | UC-2 | 1. The water level falls below a predefined threshold. | 2. The ESP32-Micro-Controler sends a signal to the relay module to turn on the water pump. | N/A | The water pump turns on and starts filling the ground tank. | The ground tank is filled with water. | BC200406015 | N/A |
| Stop water pump (no flow) | UC-3 | 1. The water flow sensor detects no water flow in the pipe. | 2. The ESP32-Micro-Controler sends a signal to the relay module to turn off the water pump. | N/A | The water pump turns off. | The water pump is off. | BC200406015 | N/A |
| Stop water pump (overflow) | UC-4 | 1. The water level rises above a predefined threshold. | 2. The ESP32-Micro-Controler sends a signal to the relay module to turn off the water pump. | N/A | The water pump turns off. | The water tank is not overflowing. | BC200406015 | N/A |
| Turn on water pump manually | UC-5 | 1. The user presses the "Turn on pump" button on the mobile app. | 2. The Blynk mobile app sends a signal to the ESP32-Micro-Controler. | 3. The ESP32-Micro-Controler sends a signal to the relay module to turn on the water pump. | N/A | The water pump turns on. | BC200406015 | N/A |
| Turn off water pump manually | UC-6 | 1. The user presses the "Turn off pump" button on the mobile app. | 2. The Blynk mobile app sends a signal to the ESP32- Micro-Controller. | 3. The ESP32-Micro-Controler sends a signal to the relay module to turn off the water pump. | N/A | The water pump turns off. | BC200406015 | N/A |

1. **Adopted Methodology**

As per requirements we will adopt the methodology named. “VU process model”.

The description which is given below.

**VU Process Model**

We use the **VU Process Model** and it contains seven steps or phases.

1. First phase is the requirement and verify in which requirement is gathering and for gathering requirement meeting is conduct with the Customer of IoT Project.
2. Second step is planning and designing of IoT Project in which plane the development of IoT Project and it complete on given time.
3. Third phase is the analyzing and designing phase so it analyzes and designing the all development of IoT Project.
4. Fourth step is the implementation and testing phases so implement the designing and planning of IoT Project and also test.
5. Fifth phase is the integration and system testing so in which integrate the development and test every IoT Project parts.
6. Sixth step is operational phase in which IoT Project is send for operation.
7. Seventh step is the maintenance phase in which maintain the operation according to the need.

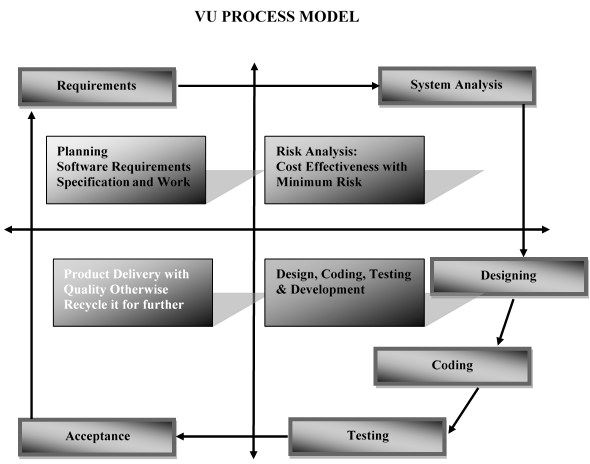


Figure 2

**Water Fall Model:**

In this approach, we see that processes flows in a downward fashion from requirement phase to accepting phase which meets to client satisfaction.  **Requirements phase** collects IoT Project specifications, functional and non-functional requirements in SRS (software requirement specification) document of VU final project.  **Analysis phase** analyzes cases and work plan on the basis of IoT Project requirement phase that covers use case scenarios up to methodology and work plan.  **Design phase** goes through with flow charts, algorithms and GUI on the basis of analysis phase. **Coding phase** generates the actual logic of the application and provide functionality to the system.  The system logic is tested if it is working as it is as in designing phase. Finally, client acceptance phase verifies if application meets to the client’s requirement.

Operation and Maintenance

Integration and System Testing

Implementation and Unit Testing

System Design

Requirement Definition

**Figure 3**

**Requirement Analysis**

This is the first phase of waterfall model which includes a meeting with the customer to understand his requirements. It is very important to understand the customer requirements and expectations so that the end product meets his specifications. All the requirements related to system to be developed are analyze in this place.

**System Design**

System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

**Implementation**

First of all, designed the system small units and implemented for testing its functionality.

**Testing**

In this stage, both individual components and the integrated whole are methodically verified to ensure that they are error free and fully meet the requirements. All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

**Deployment**

Once the functional and nonfunctional testing is done, the product is deployed.

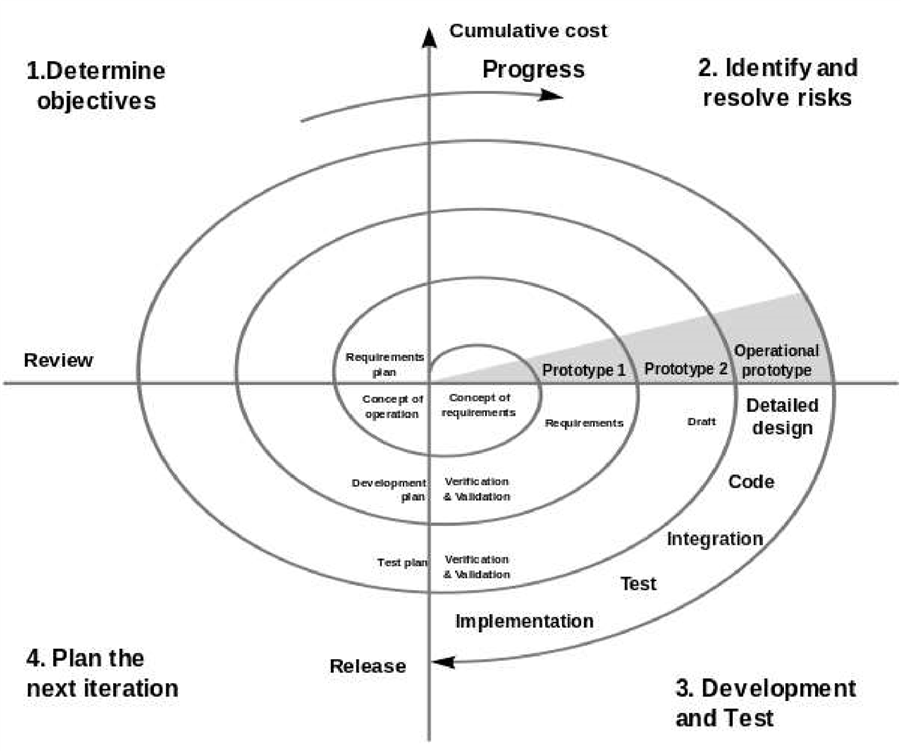
**Maintenance**

This is the final phase of the waterfall model, in which the completed IoT Project is handed over to the client after alpha, beta testing.

**Spiral model:**

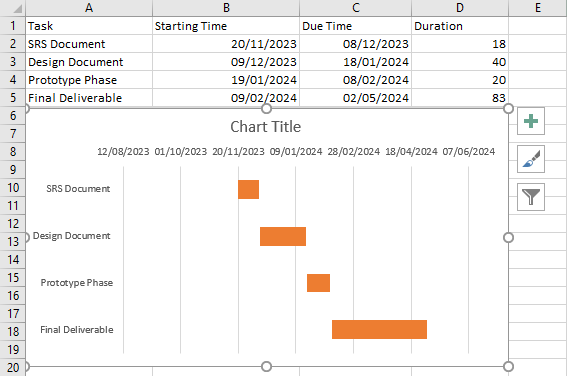
Spiral model overcomes the cons of waterfall model by adding risk analysis to its phase and these phases are iteratively followed by one after another in order to avoid maximum risk during IoT Project development. By Iteratively followed means that task move through four processes many times until quality ensured.

In the necessity stage the framework objectives, benefits and obliges are built up after talk with client. The arranging stage is synchronized push to conclusive all the potential barriers and drafts a vigorous arrangement to handle each. This stage generally includes a considerable measure of conceptualizing with in programming improvement group and with separate customer. It boosts the nature of framework and limits any hazard and mistake



**Figure 4**

1. Work Plan (Use MS Project to create Schedule/Work Plan)



**Figure 5**

End of SRS Document

Start of Design Document