**Chapter 1**

**Smart Water Tank Monitoring System**

**Introduction**

Water is one of the most essential resources for daily life, and its efficient management is crucial to prevent wastage. With increasing demand for water in residential, commercial, and agricultural sectors, ensuring its proper utilization has become a priority. However, traditional water tank systems rely on manual monitoring, which can lead to various challenges such as overflow, shortages, and inefficient usage. These conventional methods often require human intervention, making them time-consuming, unreliable, and prone to human error. Such inefficiencies result in excessive water wastage, increased electricity consumption, and inconvenience to users.

To overcome these challenges, smart water management solutions are gaining popularity. With advancements in the Internet of Things (IoT), automation has made it possible to efficiently monitor and control water usage remotely. IoT-based water monitoring systems provide real-time data on water levels and automate the operation of water pumps, ensuring effective water management.

The Smart Water Tank Monitoring System is an IoT-based project designed to automate water level monitoring and optimize water usage. It uses sensors, microcontrollers, and cloud-based platforms to detect the water level inside a tank and provide users with real-time updates and alerts. The system allows users to remotely monitor and manage their water supply through a mobile application or web dashboard.

This system not only prevents water wastage but also ensures that there is an adequate supply of water at all times. By automating the pump operation, the system eliminates the need for manual intervention, reducing human effort and ensuring a cost-effective, efficient, and sustainable approach to water management.

**Key Features of the System**

The Smart Water Tank Monitoring System offers several key features, including:

* **Real-time Monitoring** – Displays live water levels via a web or mobile application.
* **Automated Pump Control** – Turns the water pump on or off based on predefined water levels.
* **Alert Notifications** – Sends alerts via SMS, email, or app notifications when water levels are too low or too high.
* **Cloud Integration** – Stores water usage data for long-term analysis and optimization.
* **Energy Efficiency** – Reduces unnecessary pump operation, saving electricity and reducing costs.
* **User-friendly Interface** – Provides a simple and accessible dashboard for monitoring and controlling water levels remotely.

1. **Data Collection**

Data collection is a crucial step in developing an IoT-based Smart Water Tank Monitoring System, as it ensures accurate measurement, monitoring, and control of water levels. The collected data helps in analysing water usage patterns, optimizing pump operations, and preventing wastage. This chapter discusses the types of data collected, sources of data, data collection methods, and the role of IoT sensors in gathering real-time information.

* **Water Level Data** – Measures the current water level inside the tank.
* **Pump Status Data** – Indicates whether the water pump is ON or OFF.
* **Flow Rate Data** – Tracks the rate at which water is being filled or consumed.
* **Temperature and Humidity Data (optional**) – Helps monitor environmental factors that may affect water quality.
* **Time-stamped Usage Data** – Records when water is used, aiding in consumption analysis.
* **Alert and Notification Logs** – Stores alerts sent to users regarding water level status.

**Sources of Data**

The data for this system is collected from various sources, including:

* **Ultrasonic Sensors** – Measure water levels in the tank.
* **Float Sensors** – Detect minimum and maximum water levels.
* **Water Flow Sensors** – Measure the flow rate of water entering or leaving the tank.
* **Microcontrollers (Arduino/ESP8266/NodeMCU)** – Process and transmit collected data.
* **Cloud Servers (Firebase, MQTT, Thing speak, etc.)** – Store and manage data for analysis.
* **User Input** – Manual inputs from users to configure settings like threshold levels for water levels.
  1. **Data Collection Methods**

The system employs different methods for collecting and transmitting data, such as:

* **Sensor-Based Data Collection**
  + Ultrasonic sensors continuously monitor water levels and send data to the microcontroller.
  + Float sensors detect when the water reaches a certain level and trigger actions accordingly.
  + Flow sensors measure water usage and detect leakages.
* **Automated Data Logging**
  + The microcontroller processes sensor data and logs it to a cloud-based platform for real-time monitoring.
  + Time-stamped data helps in analysing daily, weekly, or monthly water consumption patterns.
* **Wireless Data Transmission**
  + Data is transmitted using Wi-Fi or GSM modules to cloud platforms like Thing Speak, Firebase, or Blynk for remote access.
  + Users receive notifications and alerts on their smartphones or web dashboards.
* **User Feedback and Manual Inputs**
  + Users can manually input threshold levels for alerts (e.g., set minimum and maximum water level limits).
  + User feedback helps improve system efficiency and customization.

**1.3 SDLC Model**

**Introduction to SDLC**

The Software Development Life Cycle (SDLC) is a structured approach used in the development of software projects to ensure systematic planning, design, implementation, and maintenance. It helps in efficiently managing project development, reducing errors, and improving quality.

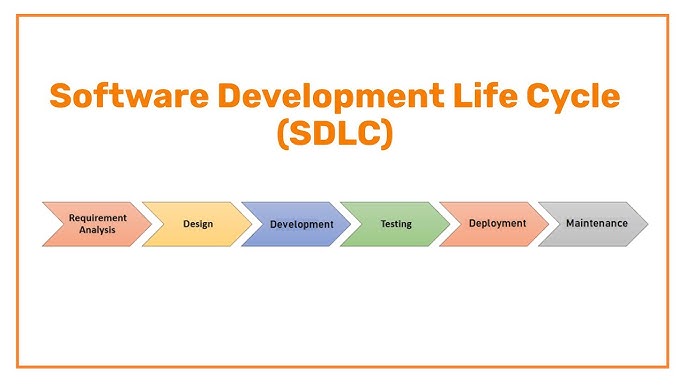
For the Smart Water Tank Monitoring System, an appropriate SDLC model is chosen to ensure smooth development, from requirements gathering to deployment.

SDLC Model Used in the Project

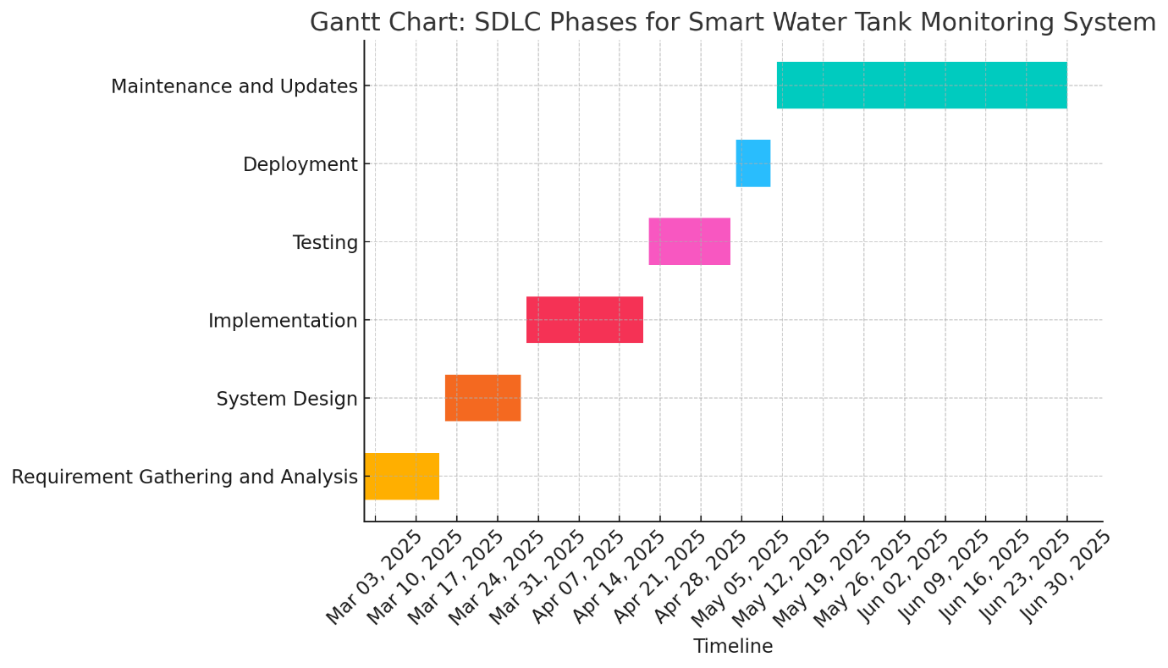
For this project, the Waterfall Model is used due to its sequential and structured approach. The development follows a step-by-step process where each phase is completed before moving to the next.

**Phases of SDLC in This Project**

1. **Requirement Analysis**
   * Identifying the need for an IoT-based water monitoring system.
   * Understanding hardware and software requirements.
   * Defining functional and non-functional requirements.
2. **System Design**
   * Designing the system architecture, including sensor placement, microcontroller selection, and cloud integration.
   * Creating a block diagram of the system components.
   * Planning the user interface for monitoring water levels.
3. **Implementation**
   * Integrating hardware components such as ultrasonic sensors, flow sensors, microcontrollers, and Wi-Fi modules.
   * Writing the software code for sensor data collection and cloud communication.
   * Developing a mobile app or web dashboard for user interaction.
4. **Testing**
   * Conducting unit testing on individual components (sensors, microcontrollers).
   * Performing integration testing to ensure seamless communication between hardware and software.
   * Running system testing to validate functionality and performance.
5. **Deployment**
   * Installing the system in a real-world environment (home, industry, or farm).
   * Configuring cloud connectivity for remote monitoring.
   * Ensuring proper data logging and notifications.
6. **Maintenance & Upgrades**
   * Monitoring system performance and fixing bugs.
   * Updating software for better efficiency and security.
   * Adding new features such as AI-based water usage predictions.



**1.4 GANTT CHART**

A Gantt chart is a powerful tool for project management, particularly when visually representing a project’s timeline and the various phases involved. In the context of the Smart Water Tank Monitoring System, using the Software Development Life Cycle (SDLC) model, the Gantt chart helps break down the project into its key phases, from initiation to completion, and shows the tasks involved in each phase. Here’s how the Gantt chart can be used to represent the timeline for this system development:

**Chapter 2**

**Software Requirement Specification (SRS)**

**2.1 Introduction**

The IoT-Based Smart Water Tank Monitoring System is an advanced solution designed to monitor and manage water levels in tanks in real time. By leveraging Internet of Things (IoT) technology, sensors, and automation, the system ensures efficient water usage, prevents overflows, and alerts users when the water level is too low or too high. This system is ideal for households, apartments, industries, and agricultural water management.

**2.2 Purpose**

The purpose of this document is to define the functional and non-functional requirements for the Smart Water Tank Monitoring System. This system provides real-time water level tracking, notifications, and automation to improve water conservation and management.

**2.3 Scope**

The Smart Water Tank Monitoring System will include the following features:

* Real-Time Water Level Monitoring: Continuously measure and display water levels.
* Automatic Pump Control: Turn the pump ON/OFF based on predefined water levels.
* IoT Connectivity: Send real-time updates to a cloud platform or mobile app.
* User Alerts & Notifications: Notify users via mobile apps, SMS, or email about critical water levels.
* Data Logging & Analysis: Store historical data for future analysis and optimization.

**2.4 Definitions, Acronyms, and Abbreviations**

* IoT (Internet of Things): A network of smart devices communicating via the internet.
* MCU (Microcontroller Unit): The processing unit controlling the system.
* Wi-Fi Module: ESP8266 or ESP32 used for IoT connectivity.
* HC-SR04 Sensor: An ultrasonic sensor used to measure water levels.
* Relay Module: A switching device to control the pump.

**2.5 References**

* IEEE 830-1998 Software Requirements Specification Standard
* Embedded System Design & IoT Development Guidelines

**2.6 Overview**

This document outlines the functional, non-functional, hardware, and software requirements for the IoT-Based Smart Water Tank Monitoring System. It also describes the system's architecture, constraints, dependencies, and user expectations to ensure efficient development.

**3. Overall Description of the Proposed System**

**3.1 Product Perspective**

The Smart Water Tank Monitoring System integrates IoT-based real-time monitoring, water level sensors, and automated motor control. The system consists of a microcontroller, ultrasonic sensors, relay module, Wi-Fi module, mobile/web application, and cloud storage for data logging.

**3.1.1 System Interfaces**

The system interacts through various interfaces:

* User Interface: Mobile app/web dashboard for real-time monitoring and notifications.
* Hardware Interface: Sensors, microcontroller, and relay module.
* Communication Interface: Data transmission via Wi-Fi to a cloud-based platform.

**3.1.2 Hardware Interfaces**

* Microcontroller: ESP8266, ESP32, or Arduino.
* Ultrasonic Sensor (HC-SR04): Measures water levels.
* Relay Module: Controls the water pump based on sensor data.
* Power Supply: 5V/12V adapter or battery backup for uninterrupted operation.

**3.1.3 Software Interfaces**

* Mobile/Web App: Displays real-time data and sends alerts.
* Cloud Integration: Blynk, Firebase, or ThingSpeak for remote access.

**3.1.4 Communication Interfaces**

* Wi-Fi (ESP8266/ESP32): Connects the system to the cloud for real-time monitoring.
* MQTT/HTTP Protocols: Used for data transmission between devices and the cloud.

**3.1.5 Operations**

* Continuous Monitoring: Tracks water level in real time.
* Pump Automation: Automatically controls the pump based on water level.
* User Notifications: Alerts users via the mobile app when the tank is empty or full.
* Data Logging: Stores water level history for future analysis.

**3.1.6 Site Adaptation Requirements**

The system is adaptable for homes, apartments, industries, and agriculture, with considerations for tank size, connectivity options, and environmental conditions.

**3.2 Product Functions**

* Water Level Measurement: Ultrasonic sensors detect and transmit real-time data.
* Pump Control Automation: The pump switches ON/OFF based on water level thresholds.
* Alerts & Notifications: Users receive alerts.
* Remote Monitoring: Users can check the water level from anywhere using an IoT-enabled dashboard.
* Data Storage: Cloud-based logging of historical water level data.

**3.3 User Characteristics**

The system is designed for various users:

* Homeowners: Monitor water levels and automate pump control.
* Building Managers: Manage water tanks for apartments and commercial buildings.
* Farmers: Optimize irrigation by monitoring water availability.
* Industrial Users: Ensure efficient water management in factories.

**3.4 Constraints**

* Power Dependency: Requires continuous power supply for operation.
* Internet Connection: Stable Wi-Fi is needed for IoT-based monitoring.
* Sensor Accuracy: Measurement errors may occur due to tank shape or environmental conditions.

**3.5 Assumptions and Dependencies**

* The user has a stable internet connection for real-time monitoring.
* The system is placed in a secure and dry location to prevent sensor damage.
* Users must configure threshold levels for notifications and pump automation.

**3.6 Future Enhancements**

Future versions of the system may include:

* AI-based Water Usage Predictions to optimize consumption.
* Battery Backup for operation during power outages.
* Solar-Powered Sensors for energy-efficient monitoring.

**4. Specific Requirements**

**4.1 External Interfaces**

**4.1.1 User Interfaces**

* Web Dashboard: Displays water level, system status, and alerts.
* Push Notifications & Alerts: Notifies users about low or high-water levels.

**4.1.2 Hardware Interfaces**

* Microcontroller (ESP8266/ESP32/Arduino)
* Ultrasonic Sensor (HC-SR04)
* Relay Module (For Pump Control)
* Power Supply (5V/12V Adapter or Battery Backup)

**4.1.3 Software Interfaces**

* Arduino IDE for Firmware Development
* Mobile App (Blynk, Firebase, or Custom Web App)
* Cloud Integration (Thing Speak, Firebase, AWS IoT)

**4.1.4 Communication Interfaces**

* Wi-Fi for IoT Connectivity
* MQTT / HTTP Protocol for Data Transmission

**4.2 Performance Requirements**

* Real-Time Response: Water level updates every 1-5 seconds.
* Low Power Consumption: Efficient operation for extended usage.
* Reliable Connectivity: Minimum 99% uptime for IoT-based monitoring.

**4.3 Design Constraints**

* Secure Data Transmission to prevent unauthorized access.

**4.4 Software System Attributes**

* **Reliability**: Continuous monitoring with minimal downtime.
* **Security**: Encrypted data transmission to ensure privacy.
* **Scalability:** Ability to support multiple tanks.

**4.5 Other Requirements**

* The system should work in both indoor and outdoor environments.
* Integration with smart home systems for automation.