PHASE 1

PROBLEM DEFINITION AND DESIGN THINKING

Problem Definition:

The problem of "smart water fountains" can be defined as designing and implementing water fountains equipped with advanced technology and sensors to optimize water usage, monitor water quality, and enhance user experience. Key aspects of this problem include, Water Conservation, Water Quality Monitoring,User Interaction,Maintenance and diagnostics,Sustainability,Data Management and Accessibility. This system should address various challenges commonly associated with Smart Water fountains.

Challenges:

- Cost
- Technical Complexity
- Power Supply
- Data security
- User adoption
- Maintenance
- Accessibility
- Environmental impact
- Regulations
- Interoperability

Solution Objectives: Smart water fountains have several objectives aimed at improving water consumption and sustainability. Here are a few key points:

1. **Water Conservation:** Smart water fountains are designed to minimize water wastage by offering precise water dispensing options, such as adjustable flow rates and automated shut-off timers.
2. **Hydration Monitoring:** Some smart fountains can track users' water intake and provide reminders to encourage regular hydration, promoting better health.
3. **Water Quality Monitoring:** They may include sensors to monitor water quality, ensuring that the water is safe and free from contaminants.
4. **Energy Efficiency:** Smart water fountains often incorporate energy-efficient features, such as LED lighting and low-power components, to reduce energy consumption.
5. **Maintenance Alerts:** These fountains can generate alerts or notifications when maintenance, such as filter replacement or cleaning, is required, ensuring the fountain remains in good working condition.

6. **User Experience:** Improved user interfaces, touchless operation, and accessibility features can enhance the overall experience for users.
7. **Data Collection:** Smart fountains can collect data on water usage patterns, helping organizations make informed decisions about water supply and infrastructure.
8. **Remote Management:** Many smart fountains allow for remote monitoring and management, enabling administrators to control settings and diagnose issues remotely.
9. **Sustainability:** They may be designed with sustainable materials and practices, promoting environmentally friendly choices.
10. **Integration:** Integration with other smart building systems, like HVAC or lighting, can optimize resource usage and improve building efficiency.

Design Thinking:

1.Empathize:

- Conduct user research to understand the needs and pain points of individuals using water fountains.
- Identify common issues such as hygiene concerns, accessibility, and water wastage.

2.Define:

- Clearly define the problem you want to address, e.g., creating a more hygienic and efficient water fountain.

3.Ideate:

- Brainstorm ideas for smart features like touchless operation, water purification, and user-friendly interfaces.
- Consider sustainability aspects such as reducing plastic waste from disposable bottles.

4.Prototype:

- Create low-fidelity prototypes to visualize and test your ideas.
- Include features like motion sensors, UV water purification, and adjustable water temperature.

5.Test:

- Gather user feedback by letting individuals interact with the prototype.
- Analyze their reactions and suggestions for improvement.

6.Refine:

- Based on user feedback, refine the design and features of the smart water fountain.
- Ensure it addresses the identified problems effectively.

7.Develop:

- Build a functional prototype or minimum viable product (MVP) of the smart water fountain.
- Collaborate with engineers and designers for the technical aspects.

8.Test Again:

- Test the functional prototype in real-world environments, such as public places or offices.
- Collect feedback on usability, reliability, and performance.

9.Iterate:

- Continuously refine and iterate on the design and functionality based on ongoing user testing and feedback.

10.Launch:

- Prepare for the commercial launch, including manufacturing, marketing, and distribution.

11.Monitor and Improve:

- After launch, collect user data and monitor the performance of the smart water fountains.
- Use this data to make continuous improvements and updates.

Coding and Explanation:

Creating Python code for a smart water fountain would depend on the specific hardware and features you have in mind. However, I can provide you with a simple example of Python code to control a basic smart water fountain using a Raspberry Pi and a relay module to control the water pump. Please adapt this code to your specific hardware and requirements:

```
import RPi.GPIO as GPIO

import time

# Define GPIO pins

pump_pin = 17  # Adjust to your setup

# Initialize GPIO

GPIO.setmode(GPIO.BCM)

GPIO.setup(pump_pin, GPIO.OUT)

def turn_on_fountain():

    GPIO.output(pump_pin, GPIO.HIGH)

    print("Fountain is ON")

def turn_off_fountain():

    GPIO.output(pump_pin, GPIO.LOW)

    print("Fountain is OFF")
```

```
try:
    while True:
        # You can implement logic here to control the fountain based on sensors or user input
        # For example, turn_on_fountain() when water level is low or based on a schedule.
        # Use turn_off_fountain() to stop the fountain when needed.
 time.sleep(1)
 # Adjust the interval as needed except KeyboardInterrupt:
    print("Exiting...")
    GPIO.cleanup()
```

This code sets up a GPIO pin on the Raspberry Pi to control the water pump. It provides functions to turn the fountain on and off. You would need to integrate this code with sensor inputs or user interface elements as per your specific requirements.Please remember to install the RPi.GPIO library on your Raspberry Pi if it's not already installed:

```
pip install RPi.GPIO
```

Additionally, ensure that your Raspberry Pi is appropriately wired to control the water pump via the relay module. Always follow safety guidelines when working with electrical components and water to avoid accidents.