```
#include <DHT.h>
#include <SoftwareSerial.h> // Library for software serial communication

// Constants
#define DHTPIN 2       // Pin where the DHT sensor is connected
#define DHTTYPE DHT22   // DHT 22 (AM2302), AM2321
#define dehumidifierPin 3 // Pin to control the dehumidifier relay
#define coldStoragePin 4  // Pin to control the cold storage relay
#define eNosePin 5       // Pin connected to E-Nose sensor input
#define storagePin 6     // Pin connected to storage relay
#define trashPin 7       // Pin connected to trash relay

// GSM module pins (SIM800L)
SoftwareSerial SIM800(8, 9); // RX, TX

// Create a DHT object
DHT dht(DHTPIN, DHTTYPE);

// Variables for tracking E-Nose readings
int previousENoseValue = LOW;

void setup() {
  Serial.begin(9600);

  pinMode(dehumidifierPin, OUTPUT);
  pinMode(coldStoragePin, OUTPUT);
  pinMode(eNosePin, INPUT);
  pinMode(storagePin, OUTPUT);
  pinMode(trashPin, OUTPUT);

  // Initialize the DHT sensor
  dht.begin();

  // Initialize GSM module
  SIM800.begin(9600);
  delay(1000);
  sendSMS("System initialized. Monitoring grain storage.");
}
void loop() {
  // Read temperature and humidity from the DHT sensor
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  // Check if any readings failed and exit early
  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  // Display readings in Serial Monitor
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.print(" °C ");
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.println(" %");

  // Check temperature and control cold storage
  if (temperature > 30) {
    digitalWrite(coldStoragePin, HIGH); // Turn on cold storage
    Serial.println("Cold storage ON");
  } else if (temperature < 25) {
    digitalWrite(coldStoragePin, LOW);  // Turn off cold storage
    Serial.println("Cold storage OFF");
  }

  // Check humidity and control dehumidifier
  if (humidity > 70) {
    digitalWrite(dehumidifierPin, HIGH); // Turn on dehumidifier
    Serial.println("Dehumidifier ON");
  } else if (humidity < 65) {
    digitalWrite(dehumidifierPin, LOW);  // Turn off dehumidifier
    Serial.println("Dehumidifier OFF");
  }
 // Read E-Nose sensor value
  int currentENoseValue = digitalRead(eNosePin);

  // If E-Nose detects a deterioration in grain quality, send an alert
  if (currentENoseValue == HIGH && previousENoseValue == LOW) {
    sendSMS("Alert: Grain quality deteriorating. Please check the storage conditions.");
    Serial.println("Alert sent: Grain quality deteriorating.");
  }

  // Update previous E-Nose value
  previousENoseValue = currentENoseValue;

  if (currentENoseValue == HIGH) {  // E-Nose detects good grain
    digitalWrite(storagePin, HIGH);  // Move to storage
    delay(2000);                    // Simulate time taken to move to storage
    digitalWrite(storagePin, LOW);   // Stop the storage process
    Serial.println("Grain moved to storage");
  } else {  // E-Nose detects rotten grain
    digitalWrite(trashPin, HIGH);    // Move to trash
    delay(2000);                   // Simulate time taken to move to trash
    digitalWrite(trashPin, LOW);     // Stop the trash process
    Serial.println("Grain moved to trash");
  }

  delay(5000);  // Wait before next reading
}

// Function to send an SMS
void sendSMS(String message) {
  SIM800.print("AT+CMGF=1\r");  // Set SMS to text mode
  delay(100);
  SIM800.print("AT+CMGS=\"+91xxxxxxxxxx\"\r");  // Replace with the farmer's phone number
  delay(100);
  SIM800.print(message);  // Message content
  delay(100);
  SIM800.write(26);  // ASCII code for Ctrl+Z (end of SMS)
  delay(1000);
}
```