System Requirements Specification Index

For

Plant Care Advisory System Console Application

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

1 Project Abstract

BloomWise Solutions, a growing agritech startup based in Pune, requires a smart gardening system with a plant care advisory program. This Python console application helps users determine watering schedules, sunlight requirements, and care instructions based on plant type, environmental conditions, and season. The system uses conditional statements (if, if-else, if-elif-else) to provide specific care recommendations. As urban gardening gains popularity across Indian metropolitan areas, BloomWise Solutions recognized the need for accessible plant care guidance for novice gardeners living in apartments with limited space and varying light conditions. Their customer research revealed that many first-time plant owners struggle with establishing proper care routines, leading to plant health issues and customer frustration. This application aims to bridge the knowledge gap by offering customized care recommendations that adapt to seasonal changes and specific plant varieties common in Indian homes.

2 Business Requirements:

Screen Name	Console input screen
Problem Statement	1. Application must determine watering frequency based on plant type using if-elif-else 2. System should adjust care based on season using if-else 3. Program should calculate sunlight requirements using if statements 4. Console should recommend plant care instructions based on conditions 5. Program should provide care warnings based on temperature and humidity

3 Constraints

3.1 Input Requirements

1. Plant Type:
   - Must be stored as integer in variable **plant_type**
   - 1: Succulent
   - 2: Tropical
   - 3: Flowering
   - 4: Herb
   - Example: 2
2. Current Season:
   - Must be stored as integer in variable season
   - 1: Spring
   - 2: Summer
   - 3: Fall
   - 4: Winter
   - Example: 1
3. Temperature:
   - Must be stored as float in variable temperature
   - Must be between -10.0 and 50.0 Celsius
   - Example: 25.5
4. Humidity:
   - Must be stored as integer in variable humidity
   - Must be between 0 and 100 percent
   - Example: 60

3.2 Calculation Constraints

1. Watering Schedule (if-elif-else):
   - Succulent (plant_type 1): Return exactly 14 days
   - Tropical (plant_type 2): Return exactly 3 days

- Flowering (plant_type 3): Return exactly 2 days
- Herb (plant_type 4): Return exactly 1 day

2. Seasonal Adjustment (if-else):
   - Summer (season 2): Decrease watering days by 1 (minimum 1 day)
   - Winter (season 4): Increase watering days by 1
   - Spring/Fall (season 1/3): No change to watering days
3. Temperature Warning (if-elif-else):
   - Too Hot: When temperature > 30.0, return exactly: "Temperature too high Risk of heat stress"
   - Too Cold: When temperature < 10.0, return exactly: "Temperature too low Risk of cold damage"
   - Optimal: Otherwise return exactly: "Temperature optimal for plant growth"
4. Humidity Requirements (if-elif-else):
   - Low: When humidity < 30, return tuple: ("Low", "Increase humidity with misting")
   - Medium: When 30 <= humidity <= 60, return tuple: ("Medium", "Humidity is optimal")
   - High: When humidity > 60, return tuple: ("High", "Monitor for fungal growth")
5. Sunlight Requirements (if-elif-else):
   - Succulent (plant_type 1): Return exactly "Full sun to partial shade"
   - Tropical (plant_type 2): Return exactly "Bright indirect light"
   - Flowering (plant_type 3): Return exactly "Full sun"
   - Herb (plant_type 4): Return exactly "At least 6 hours of direct sunlight"

3.3 Output Constraints

1. Display Format:
   - Show "Watering Schedule: Every {X} days"
   - Show "Sunlight Requirement: {requirement}"
   - Show "Temperature Status: {status}"
   - Show "Humidity Level: {level}"
   - Show "Special Care Instructions:" followed by plant-specific instructions

2. Plant-Specific Care Instructions:
    - Succulent (plant_type 1): Include "Avoid overwatering"
    - Tropical (plant_type 2): Include "Maintain high humidity"
    - Flowering (plant_type 3): Include "Remove dead flowers regularly"
    - Herb (plant_type 4): Include "Harvest regularly to promote growth"
3. Seasonal Care Tips:
    - Summer (season 2): Include "Increase watering frequency"
    - Winter (season 4): Include "Reduce watering frequency"
4. Environmental Advice:
    - High Temperature (> 30.0): Include "Provide shade and increase watering"
    - Low Temperature (< 10.0): Include "Protect from cold and reduce watering"
    - Include the humidity advice from determine_humidity_needs()

4. TEMPLATE CODE STRUCTURE:

1. Conditional Functions:

- calculate_watering_schedule() [if-elif-else]
- adjust_for_season() [if-else]
- check_temperature() [if-elif-else]
- determine_humidity_needs() [if-elif-else]
- generate_care_instructions() [nested if]
- get_sunlight_requirement() [if-elif-else]

2. Input Section:

- Get plant type (int)
- Get current season (int)
- Get temperature (float)
- Get humidity (int)

3. Processing Section:

- Calculate base watering schedule
- Apply seasonal adjustments
- Check environmental conditions
- Generate care recommendations

4. Output Section:

- Display watering schedule
- Show environmental status
- List care instructions
- Display warnings

5. Execution Steps to follow:

1. Run the program
2. Enter plant type
3. Enter current season
4. Enter temperature
5. Enter humidity level
6. View complete plant care advisory report