3S - [fictional] Smart Sprinkler System

Driver	
Contributors	@ Gene Migirov @ Kavitha Ranganathan @ Richard Powell @ Jason Lorentzen

🚀 Project overview

Goal	
Opportunities	
Challenges	

🏆 Objectives

🎨 Creative requirements

Concept statement

Design and build a modern Smart Sprinkler System that consist of:

- IoT devices (sprinkler controller, solar-powered soil & weather sensor)
- Consumer Mobile App
- Single-Page Application (SAP) (maybe)
- Cloud-native backend components

Target audience

DIY-friendly Smart Sprinkler System is all-in-one offering that provides aspiring home owner opportunity to design, install and maintain their own lawn watering sprinkler system.

Consumer will purchased self-installed kit and subscribe to vendor's on-line services to setup their newly installed system to select operating conditions providing input data such as:

1. Grass Brand & Type
2. Soil Type & Conditions
3. Geo-Location

for backend services to calculate optimized watering cycle.

Acceptance Criteria

Smart Sprinkler Management System will take Grass Brand, Soil Type, Geo-Location and Season to calculate water usage for the next water cycle recommended by the grass manufacture.

- 1 residential home installable controller unit
  - receives commands from the server on scheduled watering
  - receives commands from the mobile app on schedule overwrite
  - controls sprinkler system to turn it on/off
- 4-8 solar-powered sensors installed along perimeter of the yard will
  - monitor soil condition and report to the server every 24 hours
  - report battery charge level the mobile app
- Backend services will
  - integrate with Weather Channel to retrieve weather data at system installation location
  - take current soil, grass, weather, geo-location information to calculate and schedule the next watering cycle
  - provide controller turn-on/-off schedule
- Homeowner mobile app will
  - status of sensors operating conditions and solar battery charge level
  - provide information on the next water cycle schedule and estimated water usage
  - enables user to overwrite upcoming watering cycle

🚚 Deliverables

ℹ️ Candidate free to select and submit any 3 to 5 artifacts from the list of required deliverables using tools of their choosing based on the role they apply for

Deliverable	Specifications	Format
Technical Design Artifacts		
Deployment Architecture		Diagram
Data Flow Sequence		Diagram
IoT Integration Architecture		Diagram
Integration Architecture		Diagram
Logical and/or Physical Data Model		Diagram
UX & UI Design Artifacts		
User Interaction		Diagram
Wireframe		
Code Artifacts		
National Weather Service Integration Module	Implement weather forecast integration service using https://www.weather.gov/documentation/services-web-api	Python or Java Module
Google Identity Service Integration	Implement Mobile App or SPA Registration and Sign-Up Module using https://developers.google.com/identity/gsi/web for web /android or https://developers.google.com/identity/sign-in/ios	JavaScript or Objective-C / Swift
Backend to Controller Integration Service	Adopt MQTT to publish next water cycle schedule from the backend service to a controller	Python or Java or JavaScript
Sensor to Controller Integration Service	Adopt MQTT to publish soil condition from the sensor(s) to a controller	Python or Java or JavaScript
Controller to Backend Integration Service	Adopt MQTT to publish soil condition to the backend service to calculate the next water cycle schedule	Python or Java or JavaScript

🕛 Timeline & Review process

Deliverables due date	
Review Notes	
Reviewers	