SmartGarden – Brief complet pentru implementare

**0) Ce îmi dai ca input (obligatoriu)**

* **Mockup UI** (PDF/imagini) – flow: Login, Dashboard, Plant Detail, Add/Edit Plant, Calibrate, Settings.
* **Schema BD** (din brieful tău, vezi §1).
* **ZIP proiect curent** (dacă există).
* **Decizii de produs**:
  + Praguri default udare (ex. SoilMoistureMin=30%, Max=60%).
  + Algoritm udare automată (durată ON secunde, pauză minute, day-only?).
  + Interval citire senzori (default 15 min; 1s în calibrare).
* **Config dev**: URL API (ex. http://192.168.0.100:5000/api), DB dev (SQLite), secret JWT de test.

**1) Database (Entity Model)**

**Tabele & relații (final):**

1. Users (PK UserId) — 1–1 UserSettings, 1–N Plants
2. UserSettings (PK=FK UserId) — preferințe (auto-irrigation, thresholds, read interval)
3. Plants (PK PlantId) — N–1 Users, N–1 Species, N–1 SoilTypes, 1–N SensorReadings, 1–N WateringLogs
4. Species (PK SpeciesId) — presets: nume comun/științific + praguri implicite
5. SoilTypes (PK SoilTypeId) — nume + recomandări udare
6. SensorReadings (PK ReadingId) — N–1 Plants
7. WateringLogs (PK WateringId) — N–1 Plants

**Indexuri cheie:**  
SensorReadings(PlantId, Timestamp DESC), Plants(UserId), WateringLogs(PlantId, Timestamp).

**Seeding minim:** 6–10 Species, 4–6 SoilTypes, 1 user demo + 1–2 plants.

**2) SmartGarden.Core (Domain & Contracts)**

**Ce livrez:**

* **Models**: User, UserSettings, Plant, Species, SoilType, SensorReading, WateringLog, PlantNote (opțional).
  + Enum-uri: Placement {Indoor, Outdoor}, WateringMode {Manual, Auto}.
  + Invariante minime (ex. Plant.Validate() pentru intervale corecte).
* **DTOs** (record types):
  + Auth: LoginRequest, LoginResponse.
  + Plants: PlantCreateRequest, PlantUpdateRequest, PlantResponse.
  + Sensors: SensorReadingRequest, SensorReadingDto, HistoryQuery.
  + Watering: WaterCommandRequest{Command:"ON"|"OFF", DurationSec?}, WateringLogDto.
  + Stats: DailyStatsDto, WeeklyStatsDto.
* **Interfaces (service contracts)**:  
  IAuthService, IPlantService, ISensorService, IWateringService, IStatsService.
* **Shared**: Result<T>, Error, PagedResult<T>.

**Definition of Done (Core):**

* Toate entitățile + DTO-uri + interfețe compilabile, cu **XML summaries**.
* Nicio dependență pe EF/ASP.NET/MAUI.

**3) SmartGarden.Data (EF Core)**

**Ce livrez:**

* SmartGardenDbContext cu toate DbSet<>-urile.
* **Fluent API Configurations** per entitate (chei, relații, lungimi, default values, indexuri).
* **Migrations** (Init) + scripturi.
* **Data seeding** (Species/SoilTypes + user demo).
* **DI extension**: services.AddDataLayer(connectionString, useSqliteForDev).

**Comenzi:**

dotnet ef migrations add Init -p SmartGarden.Data -s SmartGarden.API

dotnet ef database update -p SmartGarden.Data -s SmartGarden.API

**Definition of Done (Data):**

* ApplyConfigurationsFromAssembly activ.
* SQLite dev OK, SQL Server prod OK.
* Read/write de probă (integration test sau seed vizibil în API).

**4) SmartGarden.API (ASP.NET Core Web API)**

**Stack & setup:**

* .NET 8, **JWT Auth**, **Swagger**, **CORS** (MAUI + ESP32), **AutoMapper**, **Serilog (opțional)**, **Global error middleware**.

**Config (appsettings.Development.json):**

{

"ConnectionStrings": { "Default": "Data Source=smartgarden.db" },

"Jwt": { "Key": "dev-secret-change-me", "ExpiresMinutes": 120 },

"Irrigation": { "DefaultOnSeconds": 5, "DefaultPauseMinutes": 2, "DayOnly": true },

"Cors": { "Origins": [ "http://localhost", "http://10.0.2.2", "\*" ] }

}

**Controllers & Endpoints (contract final):**

**AuthController (/api/auth)**

* POST /register → LoginRequest → int userId
* POST /login → LoginRequest → LoginResponse{token,userId}

**PlantsController (/api/plants) *(Authorize)***

* GET / → List<PlantResponse>
* GET /{plantId} → PlantResponse
* POST / → PlantCreateRequest → int newId
* PUT /{plantId} → PlantUpdateRequest → 204
* DELETE /{plantId} → 204
* PATCH /{plantId}/auto?enabled=true|false → 204

**SensorController (/api/plants/{plantId}/sensor)**

* POST /readings (ESP32) → SensorReadingRequest → {status:"OK"}
* GET /latest → SensorReadingDto
* GET /history?from=ISO&to=ISO&bucket=hour|day → List<SensorReadingDto>

**WateringController (/api/plants/{plantId}/water) *(Authorize)***

* POST / body: {"command":"ON","durationSec":5} → {message:"WATER ON"}
* GET /status → {isOn:boolean, lastRunUtc:ISO}

**CalibrationController (/api/plants/{plantId}/calibration)**

* GET / → {mode:"ON"|"OFF"}
* POST / body: {mode:"ON"|"OFF"} → 204

**StatsController (/api/stats) *(Authorize)***

* GET /daily?date=ISO → DailyStatsDto
* GET /weekly?weekStart=ISO → WeeklyStatsDto

**Răspuns eroare (global):**

{ "error": { "code": "ValidationFailed|NotFound|Unauthorized", "message": "..." } }

**Business logic (Services):**

* AuthService: user create/login + token.
* PlantService: CRUD + ownership checks (user isolation).
* SensorService: save reading, latest, history (grouping by hour/day).
* WateringService: manual trigger (log + eventual “command queue” simplă), set auto flag.
* StatsService: agregări (cele mai udate plante, sub prag azi, etc.).
* **Alerts (opțional)**: background job (HostedService) care scrie AlertEvents la prag depășit.

**Definition of Done (API):**

* JWT pe rutele protejate, CORS OK, Swagger cu exemple.
* Toate endpoint-urile funcționale conform contractului.
* Teste de integrare pentru Auth/Plants/Sensor/Water.

**5) SmartGarden.App (.NET MAUI – Mobile)**

**Arhitectură:** MVVM (CommunityToolkit.Mvvm), HttpClient + JWT, Microcharts, Notifications.

**Structură proiect:**

Models/ // DTO-uri pentru UI

Services/ // ApiService, AuthService, PlantService, SensorService, NotificationService, StorageService(SQLite)

ViewModels/ // LoginVM, DashboardVM, PlantDetailVM, AddPlantVM, CalibrateVM, SettingsVM

Views/ // \*.xaml pages (Login, Dashboard, PlantDetail, AddPlant, Calibrate, Settings)

Utils/ // Constants, Converters, Validator

App.xaml, AppShell.xaml, MauiProgram.cs

**Servicii:**

* ApiService: GET/POST/PUT/DELETE + header Bearer din Preferences["token"].
* AuthService: login/register (salvează token, userId).
* PlantService: GetPlantsAsync, Create/Update/Delete, ToggleAutoAsync, WaterNowAsync.
* SensorService: GetLatestAsync, GetHistoryAsync.
* NotificationService: ScheduleAlert(title,msg) (Plugin.Maui.Notifications).
* StorageService (opțional): cache SQLite (plants + ultimele 200 puncte/plant).

**Pagini & ViewModels:**

* **LoginPage + LoginViewModel**
  + Validare, AuthService.LoginAsync, nav //Dashboard.
* **DashboardPage + DashboardViewModel**
  + Listă PlantResponse + Pull-to-Refresh (LoadCommand la 30–60s).
  + Buton “Water now” → PlantService.WaterNowAsync.
* **PlantDetailPage + PlantDetailViewModel**
  + Latest + grafic History (24h/7d) cu Microcharts.
  + Switch “Auto irrigation” → ToggleAutoCommand.
* **AddPlantPage + AddPlantViewModel**
  + Form: Name, PlantType picker, SoilType picker, thresholds, Placement, Room, Date.
* **CalibratePage + CalibrateViewModel**
  + Wizard (Light → Soil → Water), când ON → polling GET /calibration la 1s și POST /sensor/readings frecvent din ESP.
* **SettingsPage**
  + Dark/Light, notificări, logout.

**Navigație (Shell):**

* Dacă token valid → //Dashboard; altfel → //Login.

**Permissions & build:**

* Android: Internet permission, Notifications channel.
* Windows: AppManifest pentru internet.
* Target frameworks: net8.0-android, net8.0-windows10.0.19041.0 (iOS opțional).

**Definition of Done (App):**

* Login funcțional, liste plante, udare manuală, toggle auto.
* Grafic istoric cu Microcharts.
* Notificări locale la praguri (soil < min, temp > max, water tank low).
* UI responsive, loading/errors vizibile.

**6) ESP32 Firmware (IoT)**

**Librării & pini:** WiFi, HTTPClient, Wire, SHT2x, BH1750; pini conform sketch-ului tău (soil analog, pump, MQ135, HC-SR04).

**Config:**

* SSID/PASS (placeholder).
* backendUrl = "http://<server>:<port>/api".

**Loop logic:**

* **Interval normal:** readInterval = 900000 ms (15 min).
* **Calibration:** dacă GET /plants/{id}/calibration → {"mode":"ON"} → readInterval = 1000 ms.
* **Send readings:** POST /plants/{id}/sensor/readings cu payload:

{

"temperatureC": 23.15,

"humidityAir": 41.2,

"soilMoisture": 612, // raw sau % dacă calibrare e aplicată la device

"lightLux": 130.5,

"airQuality": 410,

"waterLevelCm": 12.7

}

* **Water commands:** aplicația → API POST /plants/{id}/water cu "ON", ESP poate:
  + fie **poll** la GET /plants/{id}/water/status,
  + fie primește flag în răspunsul la POST readings (simplu: rămânem pe poll GET).  
    La WATER ON: digitalWrite(PUMP\_PIN,HIGH) pentru durationSec apoi LOW; log local.

**Fail-safe:**

* Dacă WiFi down → skip HTTP, reîncearcă.
* Hard limit ON (ex. max 60s) să eviți flood.

**Definition of Done (Firmware):**

* Trimite payload corect, răspunde la calibrare, execută WATER ON/OFF.

**7) Non-functional & cross-cutting**

* **Security**: JWT pe API; fiecare user vede doar plantele proprii.
* **CORS**: permite MAUI + ESP.
* **Observability**: logging cereri/erori; GET /health simplu (200 OK).
* **Performance**: AsNoTracking la query-uri read-only; batch insert for readings.
* **Retention**: job de arhivare/ștergere citiri > 12 luni (crontab/HostedService).

**8) Acceptanță (E2E)**

1. **Auth**: register + login → primești token, navighezi la Dashboard.
2. **Plants**: creezi plantă → apare în listă; editezi → valorile se văd în detalii.
3. **Sensors**: ESP postează reading → GET latest reflectă imediat; istoric grafic OK.
4. **Water**: “Water now” → pompă ON (simulare OK) + WateringLogs în DB.
5. **Auto**: activezi Auto → la soil < min, apare WateringLog auto.
6. **Calibration**: POST mode=ON → ESP trece la 1s; OFF → revine la 15 min.
7. **Stats**: endpointurile zilnic/săptămânal returnează agregări non-zero.
8. **Notificări**: app trimite alertă locală la prag depășit.

**9) Conectarea proiectelor (Dependencies)**

* .Core: fără dependențe → referit de .API și (pentru DTO-uri) de .App.
* .Data: referă .Core.
* .API: referă .Core + .Data.
* .App: folosește DTO din .Core + REST către .API.

**10) Ce să specifici extra în brief (dacă vrei precizie maximă)**

* Valorile exacte pentru: DefaultOnSeconds, DefaultPauseMinutes, DayOnly.
* Lista Species și SoilTypes de seed (CSV mic).
* Politica de notificări (când, text mesaj).
* Target OS (Android only? + Windows?).
* Dacă vrei **SignalR** (push) vs. polling (MVP = polling).

Altele:

**1) Irrigation defaults — DefaultOnSeconds, DefaultPauseMinutes, DayOnly**

**Ce sunt:**

* DefaultOnSeconds – durata unui impuls de udare (pompa ON) în secunde.
* DefaultPauseMinutes – pauza dintre impulsuri, să lase apa să se distribuie în sol.
* DayOnly – dacă udarea auto e permisă doar ziua (silence/no noise noaptea).

**Recomandări practice (safe pentru majoritatea ghivecelor):**

* DefaultOnSeconds: **5–10s** (start cu **5s**; pompe hobby ~150–250 ml/min).
* DefaultPauseMinutes: **2–5 min** (start cu **2 min**).
* DayOnly: **true** (08:00–22:00, Europe/Bucharest).

**Parametri suplimentari utili (opțional, dar recomand):**

* StartHour/EndHour: **8 / 22**
* HysteresisPercent: **5** (oprești când depășești pragul minim cu +5%)
* MaxConsecutiveCycles: **6** (hard stop anti-inundare)
* CooldownMinutes: **30** (pauză lungă dacă s-au atins ciclurile maxime)
* MinGapBetweenRunsMinutes: **15** (anti-flooding între rulari)

**Config API (appsettings.json):**

"Irrigation": {

"DefaultOnSeconds": 5,

"DefaultPauseMinutes": 2,

"DayOnly": true,

"StartHour": 8,

"EndHour": 22,

"HysteresisPercent": 5,

"MaxConsecutiveCycles": 6,

"CooldownMinutes": 30,

"MinGapBetweenRunsMinutes": 15

}

**Logic server (pseudo):**

if (cfg.DayOnly && !IsBetween(nowLocal, cfg.StartHour, cfg.EndHour)) return;

if (soil < min && MinutesSince(lastRun) > cfg.MinGapBetweenRunsMinutes) {

for (var i = 0; i < cfg.MaxConsecutiveCycles; i++) {

PumpOn(cfg.DefaultOnSeconds); // comanda ESP

await Task.Delay(TimeSpan.FromMinutes(cfg.DefaultPauseMinutes));

soil = await GetLatestSoil(); // re-evaluează

if (soil >= min + cfg.HysteresisPercent) break; // condiție de ieșire

}

if (i == cfg.MaxConsecutiveCycles) await Delay(cfg.CooldownMinutes);

}

**TZ:** folosește **Europe/Bucharest** pentru DayOnly/quiet hours.

**2) Seed data — liste Species & SoilTypes (CSV)**

**De ce:** preseturi pentru UI și praguri default/UX bun + demo imediat.

**Structură CSV propusă:**

species.csv

SpeciesId,CommonName,ScientificName,SoilMoistureMinPct,SoilMoistureMaxPct,TempMinC,TempMaxC,LightLuxMin,LightLuxMax,WateringPerWeek

1,Monstera,Monstera deliciosa,30,60,15,30,500,5000,2

2,Cactus,Cactaceae sp.,10,25,18,35,1000,20000,0

3,Areca Palm,Dypsis lutescens,35,65,18,30,800,8000,2

4,Snake Plant,Sansevieria trifasciata,20,40,12,30,200,5000,1

5,Peace Lily,Spathiphyllum wallisii,40,70,18,28,200,4000,3

soiltypes.csv

SoilTypeId,Name,Description,DefaultOnSeconds,DefaultPauseMinutes,DrainageFactor

1,Well-draining mix,Perlit + turbă,5,2,0.8

2,Cactus mix,Nisipos,3,3,0.9

3,Peat rich,Reține apă,6,3,0.5

4,Universal,Standard,5,2,0.7

**Implementare seed:**

* **Simplu:** HasData(...) în EF Core pentru dev/demo.
* **Flexibil:** loader CSV la startup (doar dacă tabelele sunt goale).

**Ce îmi dai:** CSV-urile finale sau confirmarea că merg cu lista propusă.

**3) Notification policy — când, ce text, rate limits**

**Tip notificări (local, în App):**

* **Soil low:** soil < min menținut **≥10 min**.
* **High temp:** temp > 40°C menținut **≥5 min**.
* **Tank low:** nivel apă < **20%** (sau distanță ultrasonic > prag).
* **Device offline:** fără citiri **>30 min**.
* **(Opțional) Auto-run summary:** după o secvență de udare.

**Throttling & dedup:**

* **Max 1 alert / tip / plantă / 2h**.
* **Daily summary** dacă problema persită >24h.
* **Quiet hours:** 22:00–08:00 (nu vibra/sunet; sau amână notificarea).

**Texte propuse (scurte, clare):**

* Soil low: 🌱 {Plant}: Umiditatea solului e scăzută (<{Min}% ). Water now?
* Temp high: 🔥 {Plant}: Temperatura e prea mare ({Temp}°C).
* Tank low: 💧 Refill: {Plant} – rezervorul de apă e aproape gol.
* Offline: 📶 {Plant}: no sensor data in last 30 min.

**Implementare (App – pseudo):**

var alerts = detector.Check(latest, thresholds, lastNotified);

foreach (var a in alerts)

if (!Throttled(a.Type, a.PlantId)) NotificationService.ScheduleAlert(a.Title, a.Message);

Persistă lastNotifiedAt per (plantId, alertType) în Preferences/SQLite.

**Ce îmi dai:** praguri finale, texte (dacă vrei alt wording), quiet hours (da/nu).

**4) Target OS — Android, Windows, iOS**

| **Platform** | **Recomandare** | **Note** |
| --- | --- | --- |
| **Android** | ✅ MVP | Ușor de testat; emulator folosește 10.0.2.2 către host; notificări ok. |
| **Windows** | ✅ Dev/Demo | Build rapid pe laptop; atenție la AppManifest pentru internet. |
| **iOS** | da | Necesită cont dev + provisioning; background limits mai stricte. |

**Setup:**

* TargetFrameworks: net8.0-android;net8.0-windows10.0.19041.0 (iOS dacă vrei).
* Permisiuni: Android INTERNET + Notifications channel; Windows internet capability.
* Icons/splash: un set comun, vector/sf.  
  **Ce îmi dai:** platformele țintă, minSdk (Android ≥24), dacă vrei și .msix pentru Windows, și dacă includem iOS.

**5) SignalR (push) vs. Polling (MVP)**

**Polling (MVP)**

* **Pro:** simplu, robust, fără infrastructură extra.
* **Contra:** latență 30–60s; trafic periodic.
* **Setări recomandate:** Dashboard poll **30–60s**, Calibrate **1s**.

**SignalR (push)**

* **Pro:** update instant (sub-secundă), excelent pentru calibrare/alerte live.
* **Contra:** complexitate mai mare (hub, reconectare, auth), impact baterie.

**Plan recomandat:**

* MVP = **Polling** (interfață IDataUpdates în App -> ușor de schimbat).
* Milestone ulterior = **SignalR**:
  + **Server:** AddSignalR(), app.MapHub<SgHub>("/hubs/sg").
  + **Evenimente:** readingUpdated, wateringStarted/Finished, alertRaised, calibrationModeChanged.
  + **Client MAUI:**
  + var conn = new HubConnectionBuilder()
  + .WithUrl($"{apiBase}/hubs/sg", o => o.AccessTokenProvider = () => Task.FromResult(token))
  + .WithAutomaticReconnect()
  + .Build();
  + conn.On<SensorReadingDto>("readingUpdated", dto => /\* update UI \*/);
  + await conn.StartAsync();

**Ce îmi dai:** confirmare că rămânem pe polling la MVP; dacă vrei push, îl planificăm ca feature.

**Mini-checklist de decizii (completează și e totul clar pentru implementare)**

**Irrigation defaults**

* DefaultOnSeconds = \_\_ (recomand 5)
* DefaultPauseMinutes = \_\_ (recomand 2)
* DayOnly = true/false (recomand true)
* StartHour/EndHour = \_\_ / \_\_ (recomand 08–22)
* HysteresisPercent = \_\_ (recomand 5)
* MaxConsecutiveCycles = \_\_ (recomand 6)
* CooldownMinutes = \_\_ (recomand 30)
* MinGapBetweenRunsMinutes = \_\_ (recomand 15)

**Seed data**

* Species CSV furnizat? da/nu (dacă nu, folosesc lista propusă)
* SoilTypes CSV furnizat? da/nu

**Notification policy**

* Soil low hold (min) = \_\_ (recomand 10)
* Temp high hold (min) = \_\_ (recomand 5, prag 40°C)
* Tank low prag (%) = \_\_ (recomand 20)
* Offline timeout (min) = \_\_ (recomand 30)
* Quiet hours 22:00–08:00 = da/nu
* Rate limit = 1 alert / tip / plantă / 2h (OK?)

**Target OS**

* Android (minSdk?) = \_\_ (recomand 24+)
* Windows = da/nu (recomand da pentru demo)
* iOS = da

**Updates model**

* MVP: Polling (30–60s dashboard, 1s calibrare) ✅
* Push (SignalR): milestone ulterior ⏳

**Definition of Done (CSV import)**

* Species & SoilTypes au câmpuri Code (UNIQUE) + IsActive.
* Service de import **CsvHelper** configurat InvariantCulture, Delimiter=",", UTF-8.
* Endpoint POST /api/admin/reference/import-csv (Admin-only).
* Seed automat (dev) din SmartGarden.Data/Seeds/\*.csv (opțional).
* Raport JSON cu Inserted/Updated/Errors.

**Plan complet – SmartGarden (de la 0 la produs)**

**0) Setup & organizare**

* **Repo mono** cu 4 foldere: /backend (API), /mobile (MAUI), /firmware (ESP32), /ops (CI/CD, Docker, doc).
* **Issue board** (Backlog/Doing/Review/Done) + template PR + convenții (branching, cod, commit).
* **.editorconfig** + stil C# + .clang-format (firmware).
* **Secret management**: .env (dev), variabile de mediu (prod).

**1) Baza de date & model (Data)**

* Definește **entități**: Users, UserSettings (1–1), Plants, Species, SoilTypes, SensorReadings, WateringLogs.
* **EF Core DbContext** + configurări Fluent (chei, FK, indexuri: SensorReadings(PlantId, CreatedAt desc)).
* **Migrație inițială** + script de creare DB.
* **Seed minim** (opțional): 1 user demo, 1–2 plante demo (pt. test rapid).
* **Upsert import CSV/Excel** (tool mic sau endpoint admin) pentru species.csv și soiltypes.csv.

**Livrabile:** proiect SmartGarden.Data, migrații, script import, README „Cum rulezi DB în 1 minut”.  
**Gata când:** migrația creează tabelele, importul CSV populă corect, indexurile apar în DB.

**2) Contracte & core (Domain)**

* **Modele domeniu** (POCO): User, Plant, Species etc.
* **DTO-uri**: LoginRequest/Response, PlantCreate/Update/Response, SensorReadingRequest/Dto, WaterCommandRequest, Daily/WeeklyStats.
* **Interfețe servicii**: IAuthService, IPlantService, ISensorService, IWateringService, IStatsService (+ IAlertService opțional).
* **Reguli business** (invariante simple: praguri min<max, proprietate plantă aparține userului etc.).

**Livrabile:** proiect SmartGarden.Core.  
**Gata când:** compilă separat; DTO-urile acoperă toate endpoint-urile.

**3) API (ASP.NET Core)**

**3.1 Config & infrastructură**

* **Program.cs**: AddControllers, Swagger (dev), AddDbContext, AddCors, AddAuthentication(JWT), AddAuthorization.
* **appsettings** (dev): ConnectionStrings:Default, Jwt:{Key,ExpiresMinutes}, Irrigation:{DefaultOnSeconds,DefaultPauseMinutes,DayOnly}, Cors:Origins.
* **Error middleware** (răspuns JSON unificat la excepții).

**3.2 Endpoints (REST)**

* /auth: POST /register, POST /login (JWT).
* /plants (🔒): GET list/one, POST, PUT, DELETE, PATCH /{id}/auto?enabled=bool.
* /plants/{id}/sensor:
  + POST /readings (AllowAnonymous + **validare deviceKey**),
  + GET /latest (🔒),
  + GET /history?from&to&bucket=hour|day (🔒).
* /plants/{id}/water:
  + (🔒) POST / (manual ON/OFF + durată),
  + (device) GET /status?deviceKey=... (polling comandă; răspunde eventual cu {isOn, durationSec}).
* /plants/{id}/calibration: POST {mode:ON|OFF} (🔒), GET /?deviceKey=... (device poll).
* /stats (🔒): GET /daily, GET /weekly.
* (opțional) /species, /soiltypes (read-only pentru aplicație).

**3.3 Servicii & logică**

* **AuthService**: parolă hash, JWT cu claim userId.
* **PlantService**: CRUD + verificare proprietate (user isolation).
* **SensorService**: salvare reading, cache „ultima citire”, agregări history.
* **WateringService**:
  + manual: log + „queue” comandă pt. device,
  + auto: la salvare reading dacă AutoWateringEnabled și SoilMoisture < Threshold + respectă PauseBetweenWaterMin și DayOnly.
* **StatsService**: agregări zilnice/săptămânale.
* **DeviceKey**: câmp Plants.DeviceKey + validare pe endpoint-urile device.

**3.4 Teste & Docker**

* **Teste integrare** (WebApplicationFactory + SQLite): autentificare, CRUD plante, POST readings, water command, auto-irrigation rule.
* **Dockerfile** API + **docker-compose** (API + SQLite/SQL Server dev).
* **README** „run în 1 comandă”.

**Livrabile:** proiect SmartGarden.API + Swagger + Docker.  
**Gata când:** toate rutele răspund, validări ok, teste verzi, container pornește local.

**4) Firmware (ESP32)**

**4.1 Config & stabilizare**

* Setări: ssid, password, backendUrl, **plantId**, **deviceKey** (trimis ca ?deviceKey=...).
* **Pini corecți**: SOIL\_MOISTURE\_PIN, PUMP\_PIN, MQ135\_PIN, TRIG\_PIN, ECHO\_PIN.
* **Read interval**: normal 15 min, în calibrare 1s.
* **Calibrare**: poll GET /plants/{id}/calibration?deviceKey=... → dacă „ON” setează interval 1s; când „OFF” revine la 15 min.
* **Trimitere date**: POST /plants/{id}/sensor/readings?deviceKey=... cu JSON (temp, humAir, soilMoisture [% sau raw], lightLux, airQuality, waterLevel [%/cm]).
* **Comenzi udare**: GET /plants/{id}/water/status?deviceKey=... → dacă isOn:true, pornește pompa durationSec (max fail-safe 60s); nu pornește dacă rezervor gol.
* **Safety**: limită hard 60s, protejează pompa dacă nivel apă < 5% (nu porni).
* (Opțional) **NVS** (persistă valori de calibrare dry/wet, dist empty/full).
* **Reconectare Wi-Fi** robustă; log pe Serial.

**Livrabile:** folder /firmware cu sketch final + instrucțiuni flash.  
**Gata când:** device trimite la 15 min, intră în calibrare 1s, execută WATER ON/OFF, respectă safety.

**5) Aplicația mobilă (MAUI)**

**5.1 Schelet & servicii**

* **MauiProgram**: DI pentru ApiService, AuthService, PlantService, SensorService, NotificationService.
* **Config**: Constants.ApiBaseUrl (dev/prod), Preferences pentru JWT.
* **Service API**: HttpClient + Bearer token auto.

**5.2 UI & MVVM**

* **Login / Register** (LoginViewModel): validează, salvează JWT, navighează la Dashboard.
* **Dashboard** (DashboardViewModel): listă plante (GET /plants); buton „Water now”; filtre (All/Indoor/Outdoor/cameră).
* **Plant Detail** (PlantDetailViewModel): GET /sensor/latest + grafic GET /history; switch Auto (PATCH /auto); Water now (POST /water).
* **Add Plant** (AddPlantViewModel): form cu pickere Species/Soil (GET), POST /plants.
* **Calibrate** (CalibrateViewModel): POST mode ON → poll latest 1s → pași ghidați (light dry/medium/bright, soil dry/wet, water empty/full) → POST mode OFF.
* **Settings**: Dark/Light, notificări on/off, logout.

**5.3 Notificări & offline**

* **Plugin.Maui.Notifications**: canal Android; afișează notificări locale pentru: sol sub prag (când Auto=off), temperatură în afara intervalului, rezervor apă scăzut.
* (Opțional) **cache local** (SQLite) pentru lista de plante și ultima citire.

**Livrabile:** proiect SmartGarden.App (Android + Windows), APK debug.  
**Gata când:** toate ecranele funcționează, comenzi udare/auto merg, notificările se declanșează corect.

**6) Integrare end-to-end**

* Conectează device la Wi-Fi + setări backendUrl/keys.
* Rulează API (docker-compose) + import CSV (Species/SoilTypes).
* App: login, add plant, vizualizare latest/history.
* Testează **calibrarea** (mod ON/OFF, frecvență 1s).
* Testează **Water now** (manual) – pompa pornește X sec, log salvat.
* Testează **Auto-irrigation**: forțează soil < prag → server marchează → device pornește → pauză minimă respectată.
* Testează **notificări**: sol uscat (Auto off), temperatură out-of-range, water tank low.

**Gata când:** toate fluxurile funcționează fără intervenții manuale în Postman.

**7) Calitate, securitate, observabilitate**

* **Validări input** (DTO) + coduri HTTP corecte (400/401/403/404).
* **JWT**: cheie secretă sigură (prod), expirare rezonabilă, clock-skew 0.
* **DeviceKey**: obligatoriu pe rutele device; rate-limit simplu (minimal).
* **CORS**: dev permisiv, prod restrâns.
* **Logs** (Serilog sau built-in): core endpoints + erori; corelare request-id.
* **Health check** /api/health.
* **Backups & retenție**: job de curățare SensorReadings > 12 luni (opțional).

**8) Deploy & CI/CD**

* **Dockerfile** API + **compose** (API + DB).
* **CI**: build + test (API, mobile unit), publish artifact (image).
* **CD**: mediul *staging* (VM sau container host), variabile securizate (JWT key, conn string).
* **DNS/SSL** (dacă expunem public) – altfel rămâne LAN.
* **Mobile release**: profiluri build, semnare APK/AAB (dacă e cazul).

**9) Documentație & handover**

* **README root** (structură, cum rulezi tot).
* **API docs** (Swagger + exemple payload).
* **Ghid firmware** (flash, setări, pairing deviceKey).
* **Ghid import CSV** (format coloane + comandă rulare).
* **Checklist QA** (scenarii e2e) și **Known Issues**.
* **Plan extensii** (OTA, push notifications, per-plant thresholds, role admin).

**Acceptanță (Definition of Done – scurt)**

* **Firmware**: trimite citiri la 15 min; în calibrare 1s; răspunde la /calibration; execută WATER ON cu limită; nu pornește când tank low.
* **API**: toate rutele funcționale; auto-irrigation declanșează corect; deviceKey validat; statistici returnate.
* **Mobile**: login, listă plante, detalii & grafice, add plant, water now, toggle auto, calibrare ghidată, notificări praguri.
* **Ops**: pornește local cu docker-compose up; DB migrată; CSV importat; health OK.
* **E2E**: de pe telefon, văd date live de la device real, pot porni pompa, primesc notificări când e cazul.

soiltypes.csv (preview)

| **Key** | **Name** | **Description** | **DefaultWaterOnSeconds** |
| --- | --- | --- | --- |
| cactus\_mix | Cactus & Succulent Mix | Very fast-draining substrate for cacti/succulents | 3 |
| universal\_potting | Universal Potting Mix | Peat-based indoor plant mix with moderate retention | 5 |
| airy\_loam | Airy Loam + Perlite | Balanced loamy mix with extra aeration | 5 |

species.csv (preview)

| **CommonName** | **ScientificName** | **SoilTypeKey** | **WateringFrequencyPerWeek** |
| --- | --- | --- | --- |
| Monstera Deliciosa | Monstera deliciosa | airy\_loam | 1.0 |
| Cactus (Echinopsis) | Echinopsis sp. | cactus\_mix | 0.25 |
| Rubber Plant | Ficus elastica | universal\_potting | 1.0 |

perfect — ți le-am „scris” eu, cu tot ce-ți trebuie ca input, gata de folosit. am inclus și fișiere descărcabile (CSV + .env de exemplu).

**1) Mockup UI**

Ai deja mockup-ul (👍). Nu mai trebuie nimic aici — doar să-l atașezi împreună cu celelalte fișiere când îmi trimiți pachetul.

**2) CSV/Excel pentru preseturi (gata de import)**

Ți-am generat două fișiere template, cu coloanele corecte + câteva rânduri exemplu:

* **species.csv** – preseturi per specie (udare, praguri, lumină, temperatură, sol)
* **soiltypes.csv** – tipuri de sol (descriere + defaulturi de udare)

👉 Descarcă species.csv  
👉 Descarcă soiltypes.csv

Dacă vrei format Excel, le pot converti în .xlsx pe loc — zi-mi doar.

**Formatele (pe scurt)**

* species.csv – coloane:
  + CommonName, ScientificName, SoilTypeKey, WateringFrequencyPerWeek, DefaultWaterOnSeconds, DefaultPauseBetweenWaterMinutes, SoilMoistureMin, SoilMoistureMax, TempMinC, TempMaxC, LightMinLux, LightMaxLux, Notes
* soiltypes.csv – coloane:
  + Key, Name, Description, DefaultWaterOnSeconds, DefaultPauseBetweenWaterMinutes, DrainageScore

**Notă:** SoilTypeKey din species.csv trebuie să existe în soiltypes.csv (ex: airy\_loam, cactus\_mix).

**3) Rețeaua (SSID/PASS) pentru device – format standard**

Trimite-mi **valorile reale** (nu în clar în public), dar ca structură îți las un fișier exemplu de configurare:

👉 Descarcă network\_dev.example.env

Conține:

WIFI\_SSID=YourWiFiSSID

WIFI\_PASSWORD=YourWiFiPassword

API\_BASE\_URL=http://192.168.0.100:5000/api

PLANT\_ID=1

DEVICE\_KEY=<GUID generat>

READ\_INTERVAL\_MS=900000

CALIBRATION\_INTERVAL\_MS=1000

**4) PlantId & deviceKey (pairing device ↔ plant)**

Ai două opțiuni:

**A. Mi le trimiți tu**

* PLANT\_ID (int) – ID-ul plantei pe care o va controla device-ul.
* DEVICE\_KEY (string sigur, tip GUID). Poți folosi exemplul din .env sau îmi lași mie să generez.

**B. Le generez eu prin API**

* Fac POST /plants → primesc { id, deviceKey }.
* Îți dau valorile, iar tu le pui în firmware (vezi mai jos) sau în .env.

**Patch minim în firmware (să fie complet)**

Adaugă deviceKey și trimite-l pe endpoint-urile pentru device.

// ====== CONFIG EXTRA ======

const char\* deviceKey = "REPLACE-WITH-YOUR-DEVICE-KEY"; // ex. GUID

// ... în sendSensorData():

http.begin(String(backendUrl) + "/plants/1/sensor/readings?deviceKey=" + deviceKey);

// ... în checkWaterCommand():

http.begin(String(backendUrl) + "/plants/1/water/status?deviceKey=" + deviceKey);

// ... în checkCalibrationMode():

http.begin(String(backendUrl) + "/plants/1/calibration?deviceKey=" + deviceKey);

Recomandare: mută plantId într-o variabilă (const int plantId = 1;) și folosește-l în URL ca să nu fie hardcodat peste tot.

**5) IP/API base URL (unde rulează backend-ul)**

Am nevoie de **adresa la care vede device-ul și telefonul** API-ul:

* Dev local tipic: http://192.168.X.Y:5000/api (IP-ul PC-ului tău în LAN).
* Prod/LAN box: IP/hostname stabil sau domeniu (dacă expui extern), **fără HTTPS** la început (mai ușor pe LAN), cu CORS deschis pentru App.

**Cum îl găsești rapid:**

* Windows: ipconfig → IPv4 Address.
* macOS/Linux: ifconfig / ip addr.

Pune exact valoarea în:

* .env (device): API\_BASE\_URL=...
* SmartGarden.App → Constants.ApiBaseUrl (sau appsettings.Development.json dacă vrei centralizat).

**Recapitulare: ce îmi trimiți acum**

1. **Mockup UI** (imaginea pe care o ai).
2. **CSV**-urile:
   * ✅ species.csv — (folosește template-ul de mai sus)
   * ✅ soiltypes.csv — (la fel)
3. **Config rețea + pairing**
   * WIFI\_SSID / WIFI\_PASSWORD
   * API\_BASE\_URL
   * PLANT\_ID și DEVICE\_KEY (sau mă lași pe mine să le generez și ți le dau înapoi)
4. (Opțional) Orice **note speciale** pe preseturi (ex: „pentru Monstera vreau alt prag default”).