This document outlines the complete Feature List, App Specifications, and Data Schema for **BIOME v2.0**, structured around the new "Data Sovereignty" architecture.

---

# BIOME v2.0: App Specification & Feature Requirements

## I. Core Philosophy: Data Sovereignty

The core philosophy for BIOME v2.0 is that the **project folder is the project**. The application's role is to act as a lightweight, fast *lens* through which users view, manage, and analyze self-contained project data.

---

## II. Architectural Blueprint: The Hybrid Model (App Specs)

To achieve both portability and performance, v2.0 switches from a central local database to a **Hybrid Model** that splits data storage based on its nature.

### 1. Data Storage & Architecture

| Component | Function | Location (The "Source of Truth") | Contents | Benefit |
|---|---|---|---|---|
| **Project Data** | **Source of Truth** (Project | **biome.json** file *inside* | Project Metadata, | **Portability:** Projects are |

| | | | | |
|---|---|---|---|---|
| | Details) | every project folder. | **Journal** (Full History), **Time Logs**, **Resource Links**, and **Protocol State**. | self-contained. The folder can be moved, archived, or shared without breaking history or links. |
| **Global Data** | The "Registry" (Performance Index) | **database.sqlite** (local %APPDATA%) | Users, Groups, App Settings, and a **Lightweight Index** of all *known* project folders (paths, status, last update time). | **Performance:** Enables **instant searching** and facility-wide statistics without scanning all project folders on every app launch. |

## 2. The "Brain": Sync & Watcher Logic

This is the critical new logic that bridges the two data stores.

| Feature | Description | Trigger |
|---|---|---|
| **Project Discovery** | The app "connects" by pointing to a root directory and recursively scans for biome.json files to populate its initial index. | User specifies a Project Root Folder. |
| **The Watcher** | A crucial background service that continuously monitors active project folders for changes to their | App is running and has an active project root path. |

| | | |
|---|---|---|
| | biome.json file. | |
| **Index Synchronization** | Automatically triggers an update to the database.sqlite Lightweight Index whenever a change is detected in a project's biome.json (e.g., a new Time Log is added). | biome.json file is modified externally or internally. |
| **Orphaned Project Handling** | If a project folder is deleted outside the app, BIOME flags the project as **"Missing"** in the global index and provides an option to clear the entry. | Project folder at an indexed path is not found. |

---

# III. Feature List (Functional Requirements)

## 1. Core Project Management & Tracking

| Feature | Requirement | Notes/Source |
|---|---|---|
| **Project Indexing** | Dynamically list and connect to projects by scanning user-specified directories for biome.json files. | New v2.0 architecture (User Request) |
| **Change Tracking** | Full history of project progress and documentation must be maintained and easily | Implemented via the **Journal** array within biome.json (User Request). |

| | | |
|---|---|---|
| | viewable. | |
| **Time Tracking** | Ability to log time spent on a project, including activity type and user. | **Time Logs** array stored directly in biome.json (User Request). |
| **Protocol Management** | Track the current step and completion status of experimental protocols within the project. | **Protocol State** object in biome.json (LIMS-like feature). |
| **Resource Linking** | Centralized location to link to external files (raw data, PDFs, final reports). | **Resource Links** array in biome.json. |

## 2. Dashboard & Stats (Analytics)

The Dashboard and Analytics features will be unified and rely on the Lightweight Index for quick filtering and aggregation.

| Metric Category | Feature Name | Description | Source |
|---|---|---|---|
| **Personal Stats** | **My Time this Period** | Aggregated time logs for the current user over a specified duration (e.g., Week/Month/Year). | Time Logs in biome.json (Aggregated by Index) |
| **Personal Stats** | **Projects Due for Review** | List of projects flagged as requiring attention (e.g., status is Review or no entry in the **Journal** for > 30 days). | Project Metadata in biome.json (Indexed) |

| | | | |
|---|---|---|---|
| **Facility Stats** | **Group/Team Usage** | Aggregate statistics for teams or groups based on active projects and logged time. | Lightweight Index in database.sqlite |
| **Facility Stats** | **Instrument Utilization** | Statistics derived from project metadata, allowing filtering by Protocol State (e.g., projects that have used "Microscope X"). | Project Metadata in biome.json (Indexed) |

---

# IV. Data Specification: Proposed biome.json Schema

This is the definition for the **Source of Truth** file for every project, which lives inside the project directory.

JSON

```
{
  // Recommended for use with VS Code Agent Mode for validation and intellisense
  "$schema": "https://universalbuilder/biome/v2/schema.json",

  // CORE METADATA
  "projectId": "string", // Unique ID for internal indexing
  "projectName": "string",
  "projectStatus": "string", // e.g., 'Active', 'On Hold', 'Archived', 'Review'
  "description": "string",
  "creationDate": "string (ISO 8601)",
  "lastUpdated": "string (ISO 8601)",

  // PROTOCOL/EXPERIMENT STATE (LIMS-like)
  "protocolState": {
```

```
    "currentStep": "string",
    "lastRunDate": "string (ISO 8601)",
    "completedSteps": "[array of strings]"
  },

  // RESOURCE LINKS (External Files/Folders)
  "resourceLinks": [
    {
      "url": "string (local path or URL)",
      "name": "string (display name)",
      "type": "string (e.g., 'Protocol PDF', 'Raw Data Folder')"
    }
  ],

  // TIME TRACKING
  "timeLogs": [
    {
      "logId": "string",
      "startTime": "string (ISO 8601)",
      "endTime": "string (ISO 8601)",
      "duration_minutes": "integer",
      "activityType": "string (e.g., 'Experiment', 'Analysis', 'Writing')",
      "user": "string (User ID)"
    }
  ],

  // CHANGE TRACKING / DOCUMENTATION (Journal 2.0)
  "journal": [
    {
      "journalId": "string",
      "timestamp": "string (ISO 8601)",
      "author": "string (User ID)",
      "entryType": "string (e.g., 'Note', 'Checkpoint', 'Data Upload')",
      "content": "string (Markdown or plain text for the documentation)",
      "changesetHash": "string (Optional: for linking to a specific file or repo state)"
    }
  ]
}
```