**Requirements and Goals of the App**

The app was designed as a cataloging tool that included authentication and system notifications. To extend its usefulness, I explored consuming API data from IoT devices, aligning the project with my long-term goals of integrating with Prismatica. The app’s purpose was to provide users with a structured, reliable way to store and retrieve data while maintaining scalability for future sensor-based integrations.

**User-Centered Design and UI Considerations**

The design emphasized logical navigation to ensure the interface felt intuitive. Following spatial hierarchy along the z, y, and x-axes, pop-ups were given top priority on the z-axis, while y and x-axis cues guided movement and interaction. This approach grounded the user in the app’s flow, reducing confusion and ensuring consistency. The designs were successful because they balanced simplicity with usability, keeping the end user at the center of the experience.

**Coding Approach and Strategies**

The development process adhered to standard best practices. I employed clear naming conventions and modular components, which was especially important given the MVC structure residing in a single directory. Resources were abstracted into subfiles for easier referential management. This modular, MVC-based approach not only enhanced readability but also promoted maintainability. These same strategies could be applied in future projects to streamline collaboration, testing, and scalability.

**Testing Methods and Importance**

Testing was primarily manual, which allowed for basic functionality checks but revealed the limitations of not having automated diagnostics. In hindsight, incorporating functional and end-to-end testing would have been valuable, especially in a project dependent on multiple libraries and frameworks. Such testing would have created a stronger foundation for identifying bugs and maintaining stability later in the lifecycle.

**Overcoming Challenges and Innovation**

One major challenge was my initial goal of enabling users to define their own objects and store them dynamically in SQLite. Since relational databases rely on fixed schemas and discourage multivalued fields, this approach proved problematic. I had to pivot to adjust the design while still delivering meaningful features. This experience became a critical learning point about aligning design ambitions with technological constraints.

**Demonstrated Skills and Success**

The device list component, along with its supporting service files, represents the strongest demonstration of my skills. It serves as the nexus of the UI, enabling core CRUD operations and forming the foundation of the app’s functionality. If the tech stack were expanded, this component could cascade into more advanced subservices for handling devices and device data. As it stands, it highlights my ability to implement and integrate key application features successfully.