Context view:

TL;DR: The app allows users to manage inventory securely and efficiently on mobile devices, with admin oversight. It has limitations, such as no offline mode or export options, and is for testing purposes only.

The context view for this application showcases the interactions and collaboration between users and the system while managing inventory records:

* Users desire a secure login to an inventory form, allowing them to add inventory details and images with ease. They should have the ability to manage their records, search for specific items, and remove outdated information.
* Administrators require access to all records within the application to oversee inventory levels and ensure accuracy. They aim to limit access to authorized users only and permit each user to update or delete their records, maintaining data security.

We can summarize each of these scenarios with user story statements:

* As a user, I want to securely log in with my credentials, ensuring that only authorized users can access the inventory form and application.
* As a user, I want to effortlessly fill out the inventory form, enabling me to input all necessary information about my inventory.
* As a user, I want to capture images of my inventory items using my device's camera and attach them to the form, providing visual documentation of the items.
* As a user, I want to review my inventory records, update them if required, and delete them if no longer relevant, allowing me to effectively manage my inventory.
* As a user, I want to search for specific inventory records, helping me find the information I need quickly and easily.
* As an administrator, I want to view all records in the application, enabling me to monitor inventory levels and verify accuracy.
* As an administrator, I want to ensure that only authorized users can access the application and that each user can only update or delete their records, maintaining data security.

Advantages of the system:

* Functionality: Implementing Test-Driven Development (TDD) on the backend ensures solid functionality.
* Security: Adhering to Django's deployment guidelines guarantees robust security measures.
* Scalability: Employing Object-Oriented Programming (OOP) allows for scalable architecture, documented using the 4+1 view pattern.
* Compatibility: The application is designed to be compatible with mobile devices.

Limitations of the system:

* Functionality: -
* Security vulnerabilities: The app won't be containerized, which may expose it to some risks.
* Lack of scalability: -
* Compatibility issues: The app will exclusively support mobile devices, limiting accessibility.
* Reliance on input: The app will use fixed values in forms, constraining flexibility.
* Limited support: The absence of user documentation and customer assistance.
* Export: Users won't be able to export inventory data to spreadsheets or other file formats.
* Notifications: The app will not notify users about updated inventory records.
* State: The application is intended for testing purposes only and won't be deployed in production.
* Mode: The app won't have an offline mode, requiring constant internet access.

Minimum Functional Requirements:

* User authentication and authorization: The application should mandate users to authenticate with valid credentials before granting access to the inventory form, and only authorized users should view, update, or delete their inventory records.
* Inventory form: The application should offer a user-friendly interface for users to input inventory information, including fields for relevant details such as item name, description, quantity, and price.
* Image upload: The application should enable users to upload pictures of their inventory items, either through direct upload or by using their device's camera.
* Search functionality: The application should provide a search function allowing users to filter and sort inventory records based on specific criteria, such as item name, price, or quantity.
* CRUD operations: The application should permit users to create, read, update, and delete their inventory records.
* Data security: The application should incorporate strong security measures to prevent unauthorized access, such as encryption, user authentication, and access control.

Verbs (function, method, keyword names): securely log in, fill out, input, capture, attach, review, update, delete, manage, search, view, monitor, verify, access, maintain.

Nouns (variable, class , object names): user, credentials, inventory form, application, information, inventory, images, device, camera, records, administrator, levels, data security,

Summarize:

The context view describes a mobile inventory management application that allows users to securely log in, add, update, and delete inventory records with images. Administrators can oversee all records and ensure data security. The system has functional, secure, and scalable features but has limitations, such as lack of export options, notifications, and offline mode. The app is for testing purposes only and requires continuous internet access.