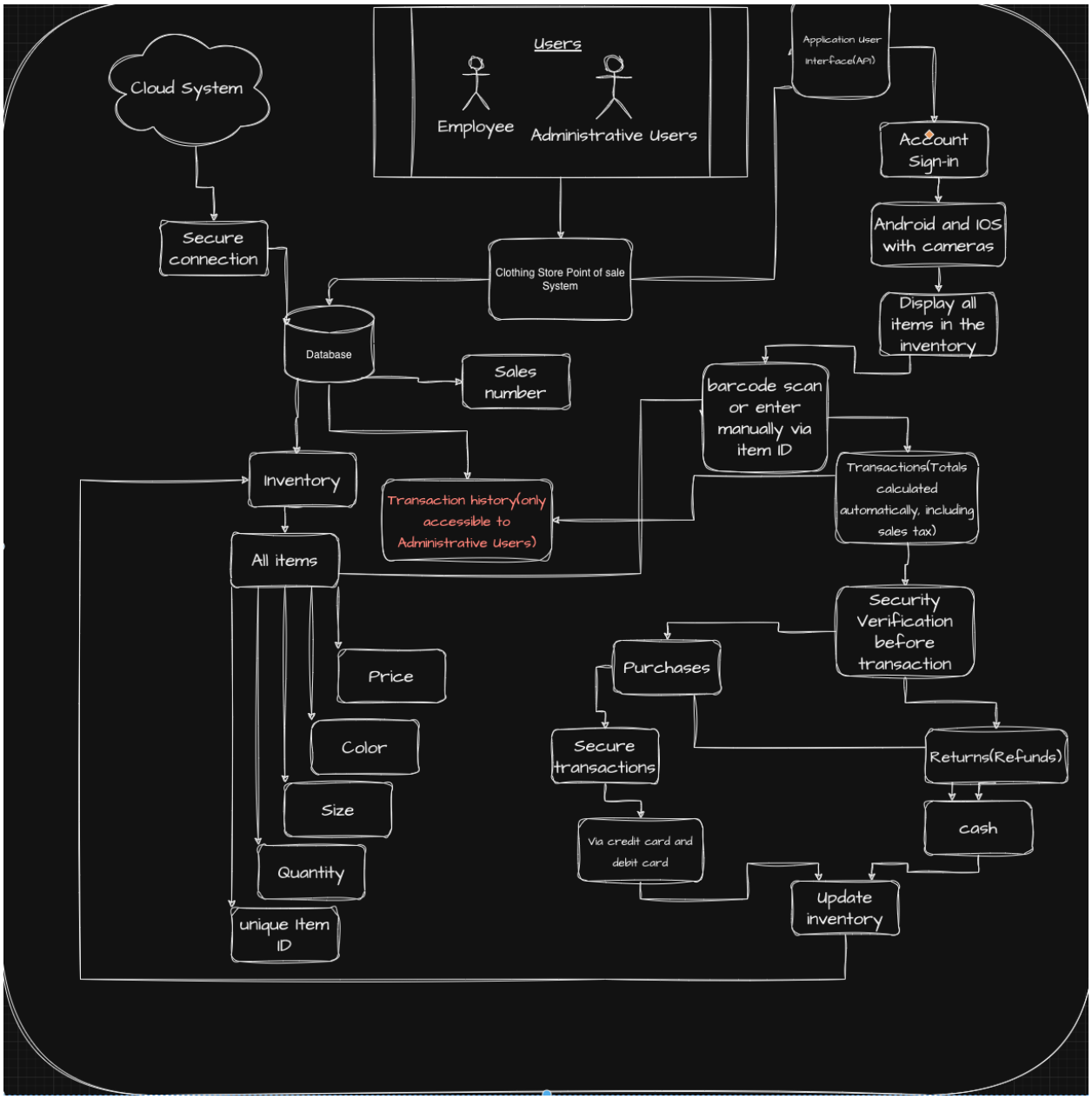
# Software Design 2.0

**Architectural Diagram:**




In this SWA diagram we can clearly see how the entire program is structured; only the employees and the administrative user will have access to this system. There will be a database connected to the cloud system. The database will manage the inventory, transaction history and the sales number. There will be an API for the system that will have things like account sign-in and will display all the items that are available in the inventory. There will be a transaction

feature added to the API. In this updated version of the SWA we can see that there are a lot of security steps being added to the system. Like security verification before any transaction, account sign-in, a secure connection between the cloud and the database.

**Data Management Strategy:**

Our data management strategy revolves around a relational database using SQL. We've chosen this approach due to the structured nature of our data, which involves items, transactions, and user information.

1) **Relational Database (SQL)**: We opted for a relational database to maintain the integrity and consistency of our structured data. SQL allows us to define relationships between different entities, ensuring efficient data retrieval and management.

2) **Single Database:** We decided to use a single database to simplify data management and maintainability. All relevant data, including inventory, transaction history, and user information, is stored in one central database.

3) **Table Organization:** Within the database, we have organized tables logically to represent key entities such as “Items”, “Transactions”, “Users”, and “SalesNumbers”. Each table contains fields corresponding to the attributes of the respective entity.

4) **Foreign Key Relationships:** To establish relationships between entities, we use foreign keys. For example, the `Transactions` table might have a foreign key referencing the `Items` table to associate transactions with specific items.

**Tradeoff Discussion:**

**Technology Choice (SQL vs. NoSQL):**
We chose SQL for its structured data model for its ease of maintaining relationships between different entities. SQL databases are well-suited for applications where data integrity is crucial, as in our point-of-sale system.

**Single Database vs. Multiple Databases:**
Opting for a single database simplifies data management and ensures consistency. However, there is a tradeoff in terms of scalability. If our system grows significantly, a shared or distributed database could be considered to distribute the load and enhance performance.

**Table Organization:**
The logical organization of tables ensures data consistency. We prioritized clarity in data organization over potential complexity.

**Possible Alternatives:**
NoSQL databases could be considered for more flexible data models, but the structured nature of our data makes SQL a more suitable choice. Multiple databases could be employed for scalability, but the added complexity seemed unnecessary for our current scale.

**Summary:**
Overall, our choice of a single, relational database using SQL aligns with the structured nature of our data and the requirements of our point-of-sale system. The tradeoffs involve considerations of scalability which we believe are acceptable given the current scope and anticipated usage of the system.