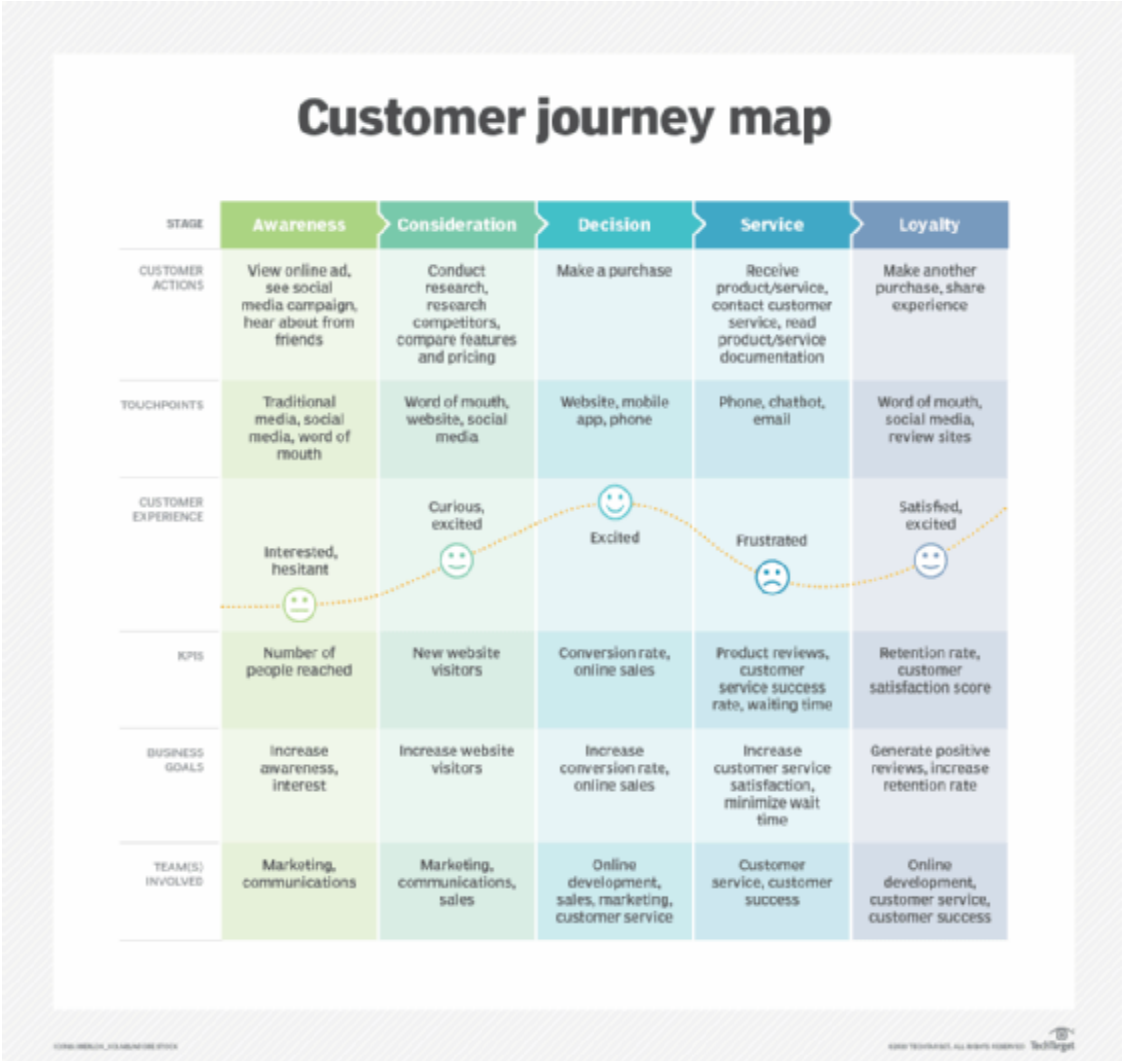
Project Design Phase
Part-2

Customer Journey Map:



Functional Requirements:

1. Data Capture and Recording: The system should be able to capture and record relevant data at each stage of the food supply chain, such as production, processing, packaging, and transportation.

2. Transparency and Accessibility: The traced food data should be easily accessible to consumers, producers, regulators, and other stakeholders, promoting transparency and accountability.

3. Real-time Tracking: The system should enable real-time tracking of food products, allowing stakeholders to monitor their movement and status throughout the supply chain.

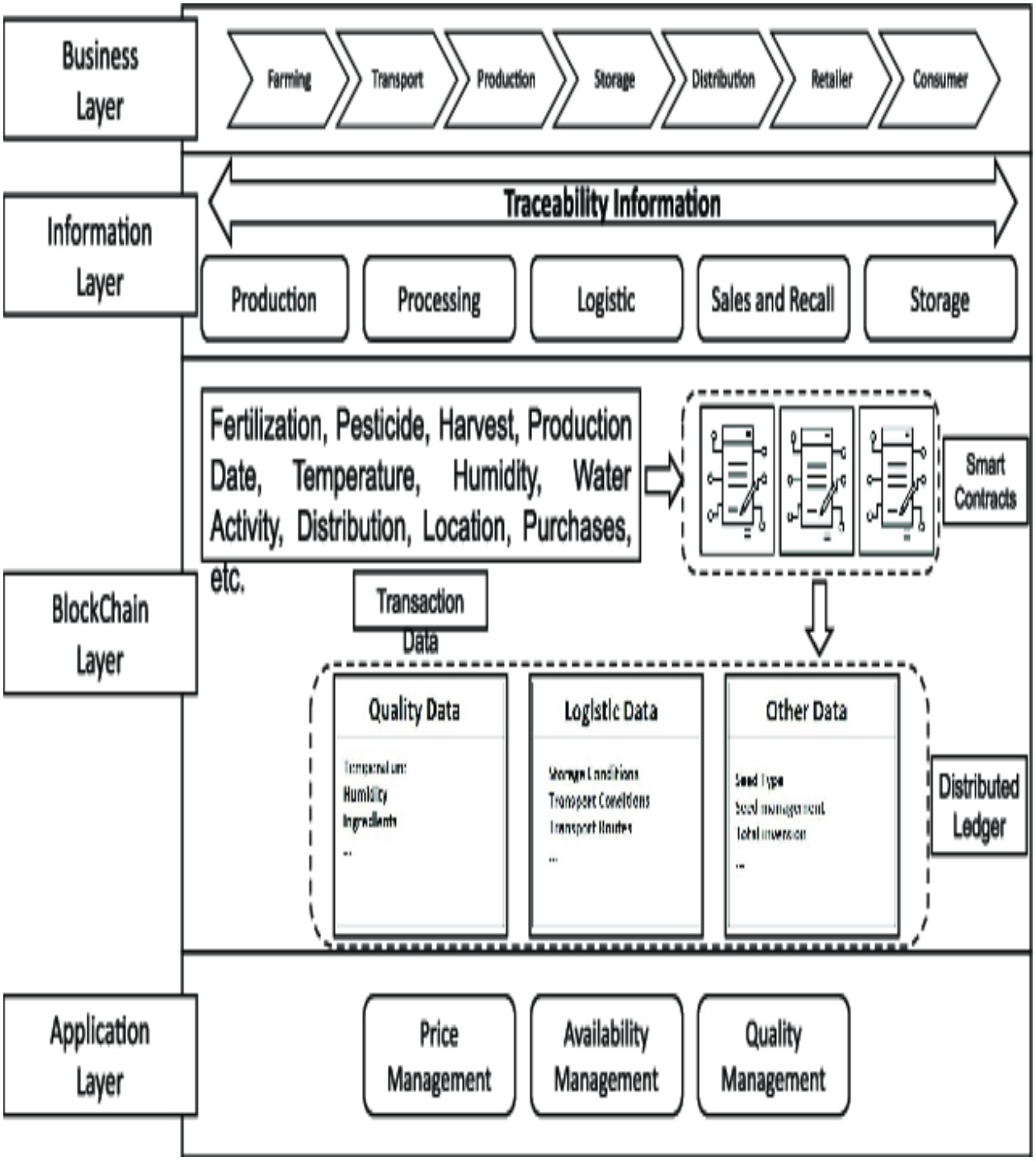
4. Traceability and Product Identification: The solution should provide unique identifiers for each food product, enabling easy traceability and identification in case of recalls or safety concerns.

5. Verification and Authentication: The system should incorporate mechanisms to verify the authenticity and integrity of traced data and ensure that it cannot be altered or tampered with.

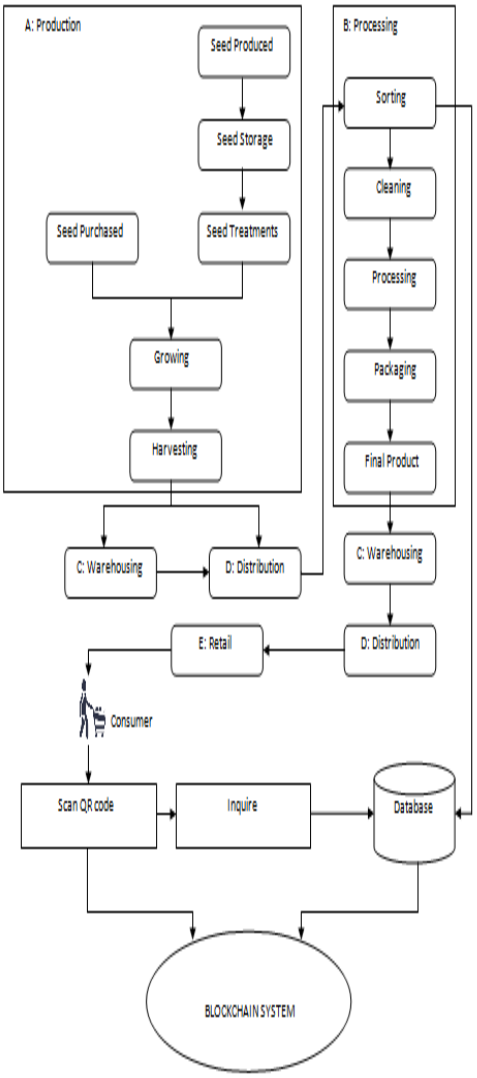
6. Integration with Existing Systems: The solution should have the ability to integrate with existing databases, information systems, and technologies already used in the food industry for streamlined data sharing and interoperability.

7. Reporting and Analytics: The system should generate meaningful reports and analytics based on the traced data, providing valuable insights for decision-making and identifying areas for improvement.

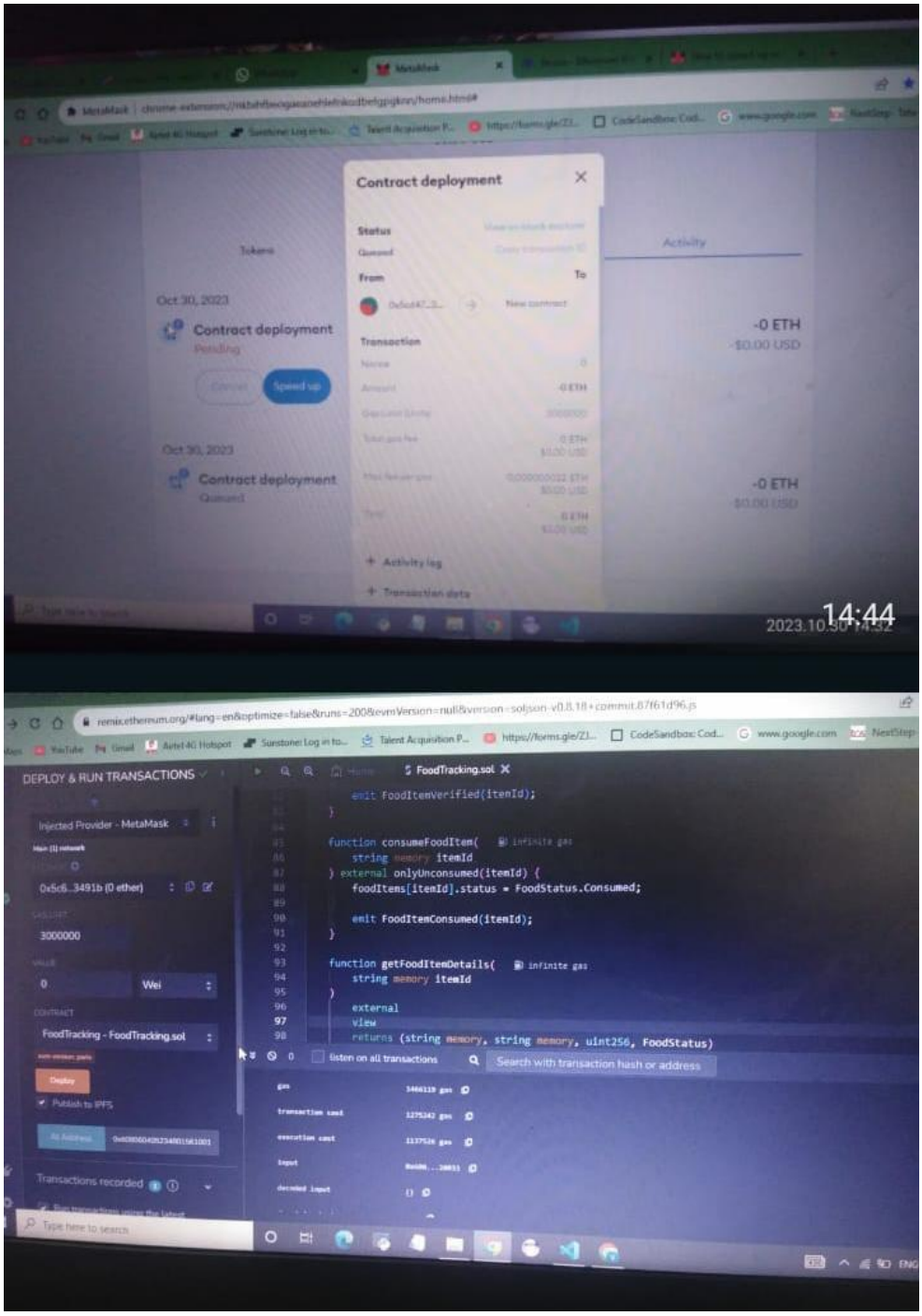
Technical Architecture:



Flow Chart:


A: Production
Seed Produced
Seed Storage
Seed Purchased
Seed Treatments
Growing
Harvesting
B: Processing
Sorting
Cleaning
Processing
Packaging
Final Product
C: Warehousing
D: Distribution
C: Warehousing
D: Distribution
E: Retail
Consumer
Scan QR code
Inquire
Database
BLOCKCHAIN SYSTEM

Deployment:


Contract deployment
Status
From
To
Transaction
Activity
Oct 30, 2023
Contract deployment
-0 ETH
Oct 30, 2023
Contract deployment
-0 ETH
14:44
DEPLOY & RUN TRANSACTIONS
FoodTracking.sol
emit FoodItemVerified(itemId);
function consumeFoodItem(
string memory itemId
) external onlyUnconsumed(itemId) {
foodItems[itemId].status = FoodStatus.Consumed;
emit FoodItemConsumed(itemId);
function getFoodItemDetails(
string memory itemId
external
view
returns (string memory, string memory, uint256, FoodStatus)
Injected Provider - MetaMask
3000000
Wei
FoodTracking - FoodTracking.sol
Deploy
Transactions recorded
Listen on all transactions
Search with transaction hash or address