Development Process Document

Design Decisions:

1. **Bridge Contract Structure:** Utilized OpenZeppelin contracts for security and standardized functionality.
2. **Safe Token Handling:** Incorporated SafeERC20 library to ensure secure token transfers and prevent potential vulnerabilities.
3. **Reentrancy Protection:** Implemented ReentrancyGuard to prevent reentrancy attacks during critical contract functions.
4. **Transaction ID Management:** Used a mapping structure to manage transaction IDs and prevent duplicate transactions, enhancing reliability.

Challenges Faced:

1. **Security:** Ensuring secure token transfers and preventing reentrancy attacks were primary concerns throughout contract development.
2. **Interoperability:** Ensuring compatibility and seamless interaction between different blockchain networks posed challenges.
3. **Transaction Verification:** Verifying transaction signatures and ensuring the authenticity of triggering transactions required careful implementation.

Addressing Challenges:

1. **SafeERC20 Integration:** Utilized SafeERC20 library to mitigate security risks associated with ERC20 token transfers and ensure robustness .
2. **Chain Type Handling:** Incorporated chain type parameters to differentiate transactions and ensure proper routing between blockchain networks.
3. **Transaction Signature Verification:** Implemented transaction signature verification using UtilityHelper contract to ensure transaction authenticity and prevent unauthorized triggering.

Detailed Guide for Interaction:

1. **Deposit Function:**
    - Call the deposit function with the token address, recipient address on the other chain, amount, and destination chain type.
    - Ensure the caller has sufficient token balance and allowance.
    - If using native currency, send ETH along with the function call.
    - Monitor emitted events for transaction status and confirmation.
2. **Trigger Function:**
    - Use the trigger function to unlock bridged tokens on the other chain.
    - Provide the token address, recipient address, amount, transaction ID, source chain type, and currency type.
    - Sign the triggering transaction with the owner's private key and provide the signature.
    - Monitor events for successful token transfer and confirmation.

Conclusion:

The Bridge contract enables seamless token transfer between different blockchain networks while addressing challenges related to security, interoperability, and transaction verification. By following the detailed guide, users can confidently interact with the contract to lock and unlock tokens across supported chains.