Explanation Document

Eager Loading vs. Lazy Loading:

- **Eager Loading**: Fetches related entities immediately along with the parent entity. This can lead to unnecessary data being loaded, which might impact performance negatively.
- **Lazy Loading**: Defers the fetching of related entities until they are accessed. This can improve performance by loading only the necessary data when it is needed.

Transactional Context:

- When methods in the Account Service class are annotated with @Transactional, they are executed within a transactional context managed by Spring.
- Within this transactional context, the Hibernate session remains open and manages the persistence and fetching of entities. This ensures that lazy-loaded properties are fetched correctly when accessed.

Without @Transactional:

- Without a transactional context, the session is closed after the primary operation (e.g., findById) is completed.
- Attempting to access lazy-loaded properties after the session is closed results in LazyInitializationException, as there is no open session to fetch the data.

With @Transactional:

- The session remains open throughout the transactional method execution.
- Lazy-loaded properties can be accessed without issues, as the session fetches them as needed within the scope of the transaction.