

기획 설계 문서가 준비되었습니다

아이디어가 구체적인 실행 계획으로 변환되었습니다.

[PRD \(요구사항\)](#)[TRD \(기술\)](#)[User Flow](#)[DB 설계](#)[디자인 시스템](#)[개발 태스크](#)[가로](#)

4. Database Design (논리적 데이터 모델)

본 프로젝트는 초기 단계에서 서버 DB를 사용하지 않고 클라이언트 상태 관리와 URL 파라미터를 사용합니다. 아래는 애플리케이션 내부에서 다루는 데이터 객체의 구조입니다.

4.1 데이터 구조 설명

- **ProductItem**: 사용자가 입력한 개별 금융 상품의 정보입니다. 예금과 적금을 구분하는 타입 필드를 가집니다.
- **CalculationResult**: 입력값을 바탕으로 계산된 결과 데이터입니다. 화면 표시를 위해 동적으로 생성됩니다.

4.2 Mermaid ERD (Client-side Model)

```
PRODUCT_ITEM {
```

```
    string id PK "고유 식별자(UUID)"
```

```
    string type "예금(Deposit) or 적금(Savings)"
```

```
    string name "상품명(선택)"
```

```
    number principal "원금/월납입액"
```

```
    number rate "연 이자율(%)"
```

```
int period "기간(개월)"
```

```
boolean isCompound "복리 여부"
```

```
}
```

```
CALCULATION_RESULT {
```

```
string id PK "ProductItem ID 참조"
```

```
number taxAmount "이자 과세액(15.4%)"
```

```
number netInterest "세후 이자 총액"
```

```
number totalReceipt "만기 수령액"
```

```
number diffFromBest "1등 상품 대비 차액"
```

```
int rank "수익률 순위"
```

```
}
```

```
PRODUCT_ITEM ||--|| CALCULATION_RESULT : "calculates"
```

PDF로 저장하기

새로운 기획 시작