**프로젝트 기획안**

**작성일 : 2022. 01. 26**

**작성자 : 황 명하**

1. 프로젝트 명 : Money Manager Project
2. 기획 의도
   1. 가계부를 인터넷으로 구현
      1. 평소 사용하는 가계부가 맘에들지 않아 직접 만들어 보다가 프로그래밍에 접목을 진행해보려고 진행 시작
      2. 누구나 자기 입맛에 맞는 가계부를 가질 수 있도록 계획.
      3. 쉽고 편하게 쓸 수 있도록 고안
   2. 보드게시판을 넣어서 사람들과 의사소통
3. 벤치마킹
   1. 후잉가계부
      1. 복식부기 방식으로 자산관리하는데 편리함 – 다만 사용법을 모른다면 어려움.
   2. 편한가계부
      1. 앱으로만 이용가능하며 파일화를 통해 excel 등에 연동을 할 수 있음.
4. 주요 기능
   1. 회원가입
      1. 로그인, 수정, 삭제, 전체보기(관리자), 로그인, 상세보기
   2. 계좌
      1. 전체보기, 수정, 삭제, 상세보기, 계좌생성
   3. 사용
      1. 전체보기, 수정, 삭제, 내역생성(계좌와 연동)
   4. 게시판
      1. 글쓰기, 내용보기, 수정하기, 삭제하기 등.
5. 기타
   1. 관련 필드 등에 대한 정보

*## MoneyManager Project*STS 에서 작업했던 것을 SpringBoot로 변경해서 진행 및 추가 내용 업데이트  
  
1. 기존에 했던 방식을 그대로 사용  
2. 모든 데이터는 가입된 사람을 기준으로 나눔  
 1. 첫 화면은 모든 데이터를 보여줌  
 2. 계좌, 현금, 카드 로 나누고 해당 데이터를 관리  
 3. 도표로 데이터를 함  
3. 해당 자료 + 물건을 사고 팔 수 있는 홈페이지 추가  
4. 글 작성 및 댓글 페이지 작성  
  
------  
초안  
  
*## MoneyManager project*- 가계부 기능을 하는 홈페이지 제작  
  
----  
*## 기능 설명*Base : 시간 기록을 위한 설정  
- startTime : 첫 기록 시간  
- nowTime : 수정 기록 시간  
  
---  
  
Member : 계좌 정보를 고객마다 다르게 관리하기 위한 기능  
- 목적  
 - 회원 관리를 하여 회원마다 정보를 나눠서 관리할 수 있게 하기 위함  
 - 회원마다 데이터가 섞이지 않으며 회원을 통해서 공유를 하여 내용을 볼 수 있게 설정(옵션)  
  
- 첨부 기능  
 - 회원가입(Join) : 회원 가입 양식을 통해 회원가입  
 - 아이디 중복 확인  
 - 비밀번호 두 번 입력을 통해 재화인  
 - 정규식을 통해 비밀번호 정규 화  
 - 로그인(Login)  
 - 회원 가입을 통해 저장된 데이터 베이스에서 아이디 검색  
 - 상세보기(findById)  
 - 로그인 한 상대가 자신의 정보 보기  
 - 수정(update)  
 - 자신의 정보를 수정할 수 있게 함  
 - 비밀번호 확인을 통해서 맞을 경우에 통과 시킴  
 - 탈퇴(delete)  
 - 회원이 자신의 정보를 남기고 싶지 않을때 탈퇴 가능.  
 - 비밀번호 확인을 통해서 확인  
 - 유예기간을 줄 수 있으면 주기(옵션)  
  
- 관리자만 가능한 기능  
 - 회원삭제(delete)  
 - 회원 정보 관리 및 회원 삭제를 통해서 데이터 정리 가능  
 - 유예기간을 줄 수 있도록 설정해보기 (옵션)  
 - 전체보기(findAll)  
 - 현재 전체 회원의 정보를 한눈에 보기 (비밀번호 확인)  
   
필드명(MemberEntity) extends BaseEntity(시간)  
- id(pk) : 분류를 위한 관리 번호  
- memberName : 회원 명  
- memberEmail : 가입 아이디  
- memberPassword : 가입 비밀번호  
- memberNickName : 사용하기 위한 닉네임  
- memberPhone : 전화번호  
- memberAddress : 주소   
- memberLevel : 등급 설정 (기본으로는 0) 관리자만 4~5  
   
---  
Bank : 계좌 정보를 기록 하기 이전에 분류  
- 목적  
 - 계좌 생성 이전에 은행별 분류를 통해 정보 정리를 편하게 하기 위함  
 - 이름을 통해 전체적인 통계를 볼 수 있게 설정  
 - 총자산을 한눈에 볼 수 있도록 표시. (Bank 상위 탭에 전체 자산의 합도 표시)  
 - 계좌 (입출금, 적금 등의 기능을 가지고 있음)  
 - 이율 설정 가능하여 해당 이율일 경우 시간에 따라서 이윤을 더하게 해주는 기능도 포함시키기(옵션)  
  
- 첨부 기능  
 - 생성(save)  
 - 계좌 정보 생성  
 - 총 자산은 처음에 입력할 수 있게 설정 (미입력시 0으로 설정)  
 - 하기 내역들의 정보를 모두 받아와서 해당 정보를 기반으로 데이터 호출  
 - 전체보기(findAll)  
 - 내 bank 관련 계좌 관련 목록 보기  
 - 해당 bank 계좌에 있는 돈들을 각각 볼 수 있으며 bank 만 따로 총합 계산을 하여 돈을 볼 수 있음 (옵션)  
 - 상세보기(findById)  
 - 한 계좌 내부에 있는 돈의 입출금 내역을 볼 수 있음  
 - balance와 연동하여 balance의 내역을 볼 수 있도록 할 수 있음.  
 - 수정(update)  
 - 총자산 또는 계좌 명을 변경 시킬 수 있음.  
 - 삭제(delete)  
 - 해당 계좌를 삭제할 수 있음.  
필드명(BankEntity) extends BaseEntity  
- id(pk) : 분류를 위한 관리 번호  
- MemberEntity (fk) : 회원 연동  
- bankName(not null) : 계좌명  
- totalAsset : 계좌의 액수가 얼마인지 확인 (마이너스 통장 기능도 할 수 있도록 설정하기)  
- bankMemo : 계좌의 정보를 기록하기 위한 메모  
- bankRate : 계좌 이율을 계산하기 위함  
- bankPhoto : 계좌 사진 필요하면 입력할 수 있게 설정  
  
---  
  
account : 계좌에 연동되어 상세보기를 들어가면 볼 수 있는 내역.  
- 목적  
 - 계좌 세부보기에서 볼 수 있으며 해당 계좌의 상세 내역을 볼 수 있음.  
 - 각가의 항목에 금액이 표시되며 +, -를 표시할 방법이 필요  
 - 1안 : 색으로 구분 / 2안 : 위치로 구분  
 - 이미지를 통해서 영수증 첨부를 할 수 있도록 설정  
  
- 첨부 기능  
 - 생성(save)  
 - 세부 내역 생성  
 - 자산을 입력할 수 있게 (미입력서 0) -> 0은 실제 입력해봐야 소용없으니... 무조건 입력  
 - 전체보기(findAll)  
 - 내역을 bank의 상세내역에 들어갔을 떄 해당 정보가 모두 뜰 수 있도록 설정.  
 - bank는 상세 내역을 2번에 걸쳐서 볼 수 있도록 제작 할 필요성 있음. (수정에 들어갔을 때 볼 수 있도록)  
 - 세부 보기(findById)  
 - 내역을 선택시 다시 새로운 세부 내역이 볼 수 있도록 하기  
 - 수정(update)  
 - 세부 보기에 들어 갔을 때만 수정을 통해서 수정할 수 있도록 설정  
 - 삭제(delete)  
 - 세부 보기와 전체보기에서 지울 수 있도록 설정  
  
필드명(AccountEntity) extends BaseEntity  
- id(pk) : 분류 번호  
- BankEntity (fk) : 연동을 위한 은행 계좌  
- MemberEntity (fk) : 필요여부 불확실 -> 고민  
- accountName : 내역명  
- accountPhoto : 첨부 영수증  
- accountMemo : 메모  
- PlusAsset : 수입  
- MinusAsset : 지출  
-   
  
---  
  
Cash : 현금 관리를 위한 계좌  
- 목적  
 - 계좌 세부보기를 통해 내역 확인 가능  
 - 전체 계좌의 총 자산 볼 수 있음   
 - 해당 계좌의 자산만 볼 수 있음  
- 첨부 기능   
 - 생성(save)  
 - 세부 내역 생성   
 - 자산을 입력할 수 있게 (미입력서 0) -> 0은 실제 입력해봐야 소용없으니... 무조건 입력   
 - 전체보기(findAll)  
 - 상세내역에 들어갔을 떄 해당 정보가 모두 뜰 수 있도록 설정.   
 - 상위 계좌는 상세 내역을 2번에 걸쳐서 볼 수 있도록 제작 할 필요성 있음. (수정에 들어갔을 때 볼 수 있도록)  
 - 세부 보기(findById)  
 - 내역을 선택시 다시 새로운 세부 내역이 볼 수 있도록 하기  
 - 수정(update)  
 - 세부 보기에 들어 갔을 때만 수정을 통해서 수정할 수 있도록 설정  
 - 삭제(delete)  
 - 세부 보기와 전체보기에서 지울 수 있도록 설정  
  
필드명(CashEntity) extends BaseEntity  
- id(pk) : 분류를 위한 관리 번호  
- MemberEntity (fk) : 회원 연동  
- cashName(not null) : 계좌명  
- totalAsset : 계좌의 액수가 얼마인지 확인 (마이너스 통장 기능도 할 수 있도록 설정하기)  
- cashMemo : 계좌의 정보를 기록하기 위한 메모  
- cashPhoto : 계좌 사진 필요하면 입력할 수 있게 설정  
-  
  
---  
  
balance : 계좌에 연동되어 상세보기를 들어가면 볼 수 있는 내역.  
- 목적  
 - 계좌 세부보기에서 볼 수 있으며 해당 계좌의 상세 내역을 볼 수 있음.  
 - 각가의 항목에 금액이 표시되며 +, -를 표시할 방법이 필요  
 - 1안 : 색으로 구분 / 2안 : 위치로 구분  
 - 이미지를 통해서 영수증 첨부를 할 수 있도록 설정  
  
- 첨부 기능  
 -생성(save)  
 - 세부 내역 생성  
 - 자산을 입력할 수 있게 (미입력서 0) -> 0은 실제 입력해봐야 소용없으니... 무조건 입력  
 - 전체보기(findAll)  
 - 상세내역에 들어갔을 떄 해당 정보가 모두 뜰 수 있도록 설정.  
 - 상세 내역을 2번에 걸쳐서 볼 수 있도록 제작 할 필요성 있음. (수정에 들어갔을 때 볼 수 있도록)  
 - 세부 보기(findById)  
 - 내역을 선택시 다시 새로운 세부 내역이 볼 수 있도록 하기  
 - 수정(update)  
 - 세부 보기에 들어 갔을 때만 수정을 통해서 수정할 수 있도록 설정  
 - 삭제(delete)  
 - 세부 보기와 전체보기에서 지울 수 있도록 설정  
  
필드명(balanceEntity) extends BaseEntity  
- id(pk) : 분류 번호  
- cashEntity (fk) : 연동을 위한 은행 계좌  
- memberEntity (fk) : 필요여부 불확실 -> 고민  
- balanceName : 내역명  
- balancePhoto : 첨부 영수증  
- balanceMemo : 메모  
- plusAsset : 수입  
- minusAsset : 지출  
  
---  
  
Card : 현금 관리를 위한 계좌  
- 목적  
 - 계좌 세부보기를 통해 내역 확인 가능  
 - 전체 계좌의 총 자산 볼 수 있음  
 - 해당 계좌의 자산만 볼 수 있음  
- 첨부 기능  
 - 생성(save)  
 - 세부 내역 생성  
 - 자산을 입력할 수 있게 (미입력서 0) -> 0은 실제 입력해봐야 소용없으니... 무조건 입력  
 - 전체보기(findAll)  
 - 상세내역에 들어갔을 떄 해당 정보가 모두 뜰 수 있도록 설정.  
 - 상위 계좌는 상세 내역을 2번에 걸쳐서 볼 수 있도록 제작 할 필요성 있음. (수정에 들어갔을 때 볼 수 있도록)  
 - 세부 보기(findById)  
 - 내역을 선택시 다시 새로운 세부 내역이 볼 수 있도록 하기  
 - 수정(update)  
 - 세부 보기에 들어 갔을 때만 수정을 통해서 수정할 수 있도록 설정  
 - 삭제(delete)  
 - 세부 보기와 전체보기에서 지울 수 있도록 설정  
  
필드명(CardEntity) extends BaseEntity  
- id(pk) : 분류를 위한 관리 번호  
- MemberEntity (fk) : 회원 연동  
- cardName(not null) : 계좌명  
- totalAsset : 계좌의 액수가 얼마인지 확인 (마이너스 통장 기능도 할 수 있도록 설정하기)  
- cardMemo : 계좌의 정보를 기록하기 위한 메모  
- cardPhoto : 계좌 사진 필요하면 입력할 수 있게 설정  
-   
  
---  
  
Debit : 계좌에 연동되어 상세보기를 들어가면 볼 수 있는 내역.  
- 목적  
 - 계좌 세부보기에서 볼 수 있으며 해당 계좌의 상세 내역을 볼 수 있음.  
 - 각가의 항목에 금액이 표시되며 +, -를 표시할 방법이 필요  
 - 1안 : 색으로 구분 / 2안 : 위치로 구분  
 - 이미지를 통해서 영수증 첨부를 할 수 있도록 설정  
  
- 첨부 기능  
 -생성(save)  
 - 세부 내역 생성  
 - 자산을 입력할 수 있게 (미입력서 0) -> 0은 실제 입력해봐야 소용없으니... 무조건 입력  
 - 전체보기(findAll)  
 - 상세내역에 들어갔을 떄 해당 정보가 모두 뜰 수 있도록 설정.  
 - 상세 내역을 2번에 걸쳐서 볼 수 있도록 제작 할 필요성 있음. (수정에 들어갔을 때 볼 수 있도록)  
 - 세부 보기(findById)  
 - 내역을 선택시 다시 새로운 세부 내역이 볼 수 있도록 하기  
 - 수정(update)  
 - 세부 보기에 들어 갔을 때만 수정을 통해서 수정할 수 있도록 설정  
 - 삭제(delete)  
 - 세부 보기와 전체보기에서 지울 수 있도록 설정  
  
필드명(DebitEntity) extends BaseEntity  
- id(pk) : 분류 번호  
- cardEntity (fk) : 연동을 위한 은행 계좌  
- bankEntity (fk) : 계좌에서 돈이 바로 출금될 수 있도록 설정  
- memberEntity (fk) : 필요여부 불확실 -> 고민  
- debitName : 내역명  
- debitPhoto : 첨부 영수증  
- debitMemo : 메모  
- debitMinusAsset : 지출  
- debitGet : 할인 또는 캐쉬백 항목  
 - 두가지 타입으로 나눠야함 / 정액, 퍼센트  
 - 방법고안 필요 1안 : select로 해서 적용 / 필드 2개 생성  
   
---  
  
Credit : 계좌에 연동되어 상세보기를 들어가면 볼 수 있는 내역.  
- 목적  
 - 계좌 세부보기에서 볼 수 있으며 해당 계좌의 상세 내역을 볼 수 있음.  
 - 각가의 항목에 금액이 표시되며 +, -를 표시할 방법이 필요  
 - 1안 : 색으로 구분 / 2안 : 위치로 구분  
 - 이미지를 통해서 영수증 첨부를 할 수 있도록 설정  
  
- 첨부 기능  
 -생성(save)  
 - 세부 내역 생성  
 - 자산을 입력할 수 있게 (미입력서 0) -> 0은 실제 입력해봐야 소용없으니... 무조건 입력  
 - 전체보기(findAll)  
 - 상세내역에 들어갔을 떄 해당 정보가 모두 뜰 수 있도록 설정.  
 - 상세 내역을 2번에 걸쳐서 볼 수 있도록 제작 할 필요성 있음. (수정에 들어갔을 때 볼 수 있도록)  
 - 세부 보기(findById)  
 - 내역을 선택시 다시 새로운 세부 내역이 볼 수 있도록 하기  
 - 수정(update)  
 - 세부 보기에 들어 갔을 때만 수정을 통해서 수정할 수 있도록 설정  
 - 삭제(delete)  
 - 세부 보기와 전체보기에서 지울 수 있도록 설정  
  
필드명(CreditEntity) extends BaseEntity  
- id(pk) : 분류 번호  
- cardEntity (fk) : 연동을 위한 은행 계좌  
- bankEntity (fk) : 계좌에서 돈이 바로 출금될 수 있도록 설정  
- memberEntity (fk) : 필요여부 불확실 -> 고민  
- creditName : 내역명  
- creditPhoto : 첨부 영수증  
- creditMemo : 메모  
- creditMinusAsset : 지출  
- creditRate : 할부시 이자 표시하기 위해서  
- creditMonth : 할부 설정 (최대 60)  
- creditGet : 캐쉬백이나 할인 부분 설정  
 - 두가지 타입으로 나눠야함 / 정액, 퍼센트  
 - 방법고안 필요 1안 : select로 해서 적용 / 필드 2개 생성  
  
  
---  
  
Board : 자유게시판  
- 목적  
 - 자유롭게 떠들기 위함  
 - Q&A등의 문답 가능  
- 첨부 기능  
 - 생성(save)  
 - 게시글 작성  
 - 닉네임을 작성자로 설정  
 - 전체보기(findAll)  
 - 전체 게시글 보기  
 - 상세내역에 들어갔을때 내용 볼 수 있게 설정  
 - 세부 보기(findById)  
 - 내용 보기 및 수정 버튼 삽입  
 - 수정(update)  
 - 세부 보기에 들어 갔을 때만 수정을 통해서 수정할 수 있도록 설정  
 - 삭제(delete)  
 - 세부 보기에서 지울 수 있도록 설정  
  
필드명(BoardEntity) extends BaseEntity  
- id(pk) : 분류를 위한 관리 번호  
- MemberEntity (fk) : 회원 연동  
- boardTitle(not null) : 제목  
- boardWriter : 작성자  
- boardContents : 게시글 내용  
- boardHits : 조회수  
- boardPhoto : 이미지 첨부  
-   
  
---  
  
Comment : 댓글 쓰기 기능  
- 목적  
 - 댓글 작성을 할 수 있음  
 - 좋아요 싫어요 ?  
  
- 첨부 기능  
 -생성(save)  
 - 댓글 생성  
 - 전체보기(findAll)  
 - board 와 notice에서 들어갔을 때 바로 볼 수 있게  
 - 수정(update)  
 - 댓글 수정  
 - 삭제(delete)  
 - 댓글 삭제  
  
필드명(CommentEntity) extends BaseEntity  
- id(pk) : 분류 번호  
- BoardEntity (fk) : 댓글 연동  
- NoticeEntity (fk) : 댓글 연동  
- commentWriter : 작성자 (로그인 닉네임 또는 미 로그인시 댓글 불가)  
- commentContents : 댓글 내용  
- commentHate : 싫어요  
- commentLike : 좋아요  
-   
  
---  
  
Notice : 공지게시판  
- 목적  
 - 곻지 적용  
 - Q&A등의 문답 가능  
- 첨부 기능  
 - 생성(save) (관리자 설정된 사람만 가능)  
 - 게시글 작성  
 - 닉네임을 작성자로 설정  
 - 전체보기(findAll)  
 - 전체 게시글 보기  
 - 상세내역에 들어갔을때 내용 볼 수 있게 설정  
 - 세부 보기(findById)  
 - 내용 보기 및 수정 버튼 삽입 (관리자 설정된 사람만 가능)  
 - 수정(update) (관리자 설정된 사람만 가능)  
 - 세부 보기에 들어 갔을 때만 수정을 통해서 수정할 수 있도록 설정  
 - 삭제(delete) (관리자 설정된 사람만 가능)  
 - 세부 보기에서 지울 수 있도록 설정  
  
필드명(NoticeEntity) extends BaseEntity  
- id(pk) : 분류를 위한 관리 번호  
- MemberEntity (fk) : 회원 연동  
- noticeTitle(not null) : 제목  
- noticeWriter : 작성자  
- noticeContents : 게시글 내용  
- noticeHits : 조회수  
- noticePhoto : 이미지 첨부  
  
---  
<이 기능은 고민>  
  
Shop : 쇼핑몰? 중고장터?  
목적  
- 제품을 보여줘서 해당 제품들이 제대로 나올 수 있게?  
  
- 기능  
 - 저장(save)  
 - 제품 목록화  
 - 전체보기(findAll)  
 - 저장된 모든 항목을 격자로 표현해서 표현  
 - 세부보기(findById)  
 - 세부 정보를 표현할 수 있게 함.  
 - 제품 삭제 (delete)  
 - 삭제  
 - 제품 수정(update)  
 - 수정  
  
필드  
- id (pk) : 고유번호  
- memberEntity (fk) : 제품을 올리기 위한 것?  
- shopTitle : 제목  
- shopPhoto : 제품 사진  
- shopContents : 내용을 넣기 위함  
- shopInfo : 판매 정보를 올리기 위함  
- shopPrice : 제품 가격  
  
  
---  
ShopList : 해당 세부 정보  
- 리뷰용?