객체지향 설계 과제

- 설계 및 최종 보고서 -

과목명 : 객체지향 설계
담당교수 : 홍장의
학과 : 소프트웨어학과
팀명 : 익명의 오리
팀원 - 2017038079 차재현
- 2017038064 김동용
- 2017038054 박재원

- 목 차 -

시스템 정의서

프로젝트 팀명	익명의 오리	작성 일자	2021-03-20
시스템 명칭	가계부 관리 어플리케이션		
시스템 설명	종이 가계부에는 여러 한계점이 있습니다. 첫 번째, 일일이 수기로 수입, 지출을 통계하여야 한다는 번거로움. 두 번째, 물질적으로 장부를 잃어버리거나 훼손 시 소중한 정보를 확인하지 못한다는 위험성. 세 번째, 장부의 기록 공간 부족으로 인한 지속적인 관리/유지의 어려움과 물리적으로 보관하는데 있어 비효율적인 공간 차지 문제점. 위와 같은 종이 가계부의 한계점을 보완하여 평소 관리하기 힘들었던 개인 자산을 어플리케이션을 통해 더 효율적이고 편리하게 관리하기 위한 프로그램입니다. 프로그램의 큰 특징으로는 지출/수입 내역 기록, 예산 편성 기능, 기간 별 금액 흐름 검토 등의 기능이 있습니다.		
운영 환경	Android, ios 어플리케이션		
주요 기능	1. 수입, 지출 내역 입력 및 수정 2. 월간 재정 내역 한 눈에 보기 3. 주간별 수입/지출 그래프로 확인 4. 사용 내역 검색 기능 5. 예산 관리		

요구사항 정의서

프로젝트명	가계부 관리 프로그램		
프로젝트 팀명	익명의 오리	작성일	2021년 03월 22일
프로젝트 설명	종이 가계부의 한계점을 보완하여 평소 관리하기 힘들었던 개인 자산을 어플리케이션을 통해 더 효율적이고 편리하게 관리 하기 위한 프로그램입니다.		
사용 대상자	편리하고 깔끔한 정리를 원하는 가계부 사용자들		

□ 기능적 요구사항

주 기능	수입, 지출 내역 입력 및 수정	번호	F1	중요도	상
요구사항	F1-1. 사용자는 지출, 입금 내역의 금액을 입력할 수 있다. F1-2. 사용자는 지출, 입금 내역의 내용(용돈, 월급, 이월 등)을 입력할 수 있다. F1-3. 사용자는 지출, 입금 내역의 날짜를 입력할 수 있다. F1-4. 사용자는 기존 내역의 금액, 내용, 날짜를 수정할 수 있다.				

주 기능	기간별 내역 확인	번호	F2	중요도	상
요구사항	F2-1. 사용자는 인터페이스 화면에서 자신이 알고자 하는 내역들의 기간을 입력한다. F2-2. 사용자는 입력한 기간에 해당하는 수입/지출 내역들을 클릭하여 수입/지출 종류, 금액, 내용을 확인할 수 있다. F2-2-1. 사용자는 입력한 기간의 주간별 단위로 기록되어있는 지출/수입 내역을 꺾은선 그래프로 확인할 수 있다. F2-4. 사용자는 인터페이스의 상단 가운데 O월 버튼을 클릭하면 새로운 페이지가 나타나며 월간 내역을 확인할 수 있다.				

주 기능	수입/지출 내역 검색 기능	번호	F3	중요도	상
요구사항	F3-1. 사용자는 내용 키워드를 입력하여 수입/지출 내역을 검색할 수 있다. F3-2. 사용자는 검색 버튼을 누르면 조건에 맞는 수입/지출 내역을 확인할 수 있다. F3-2-1. 사용자로부터 입력받은 키워드가 '수입/지출 내역'의 입력해놓은 내용에 포함되어 있는 내역만을 출력한다. F3-2-2. 검색 결과 내역의 전체 수를 검색 버튼 왼쪽 아래에 Total $로 표시한다.				

주 기능	예산 편성/수정/확인	번호	F4	중요도	중
요구사항	F4-1. 사용자는 예산편성 기능을 통해 월 단위로 예산을 기록할 수 있다. F4-2. 사용자는 예산편성 버튼을 클릭하면 월 단위로 기간을 입력하고, 사용할 금액과 종류, 내용을 입력할 수 있다. F4-3. 사용자는 예산편성 수정 기능을 통해 기존의 편성된 예산 내역의 기간, 금액, 종류 , 내용을 수정할 수 있다. F4-4. 사용자는 기록했던 모든 예산 편성 내역을 확인할 수 있다.				

주 기능	사용자 인증	번호	F5	중요도	상
요구사항	F5-1. 사용자는 인증하기 기능을 통해 비밀번호를 입력한다. F5-2. 시스템은 DB에 저장되어있는 인증 비밀번호와 비교한다. F5-3. 시스템은 인증 비밀번호 비교하고 인증 결과를 출력한다.				

주 기능	과거 지출비교 및 경고메세지 전송	번호	F6	중요도	상
요구사항	F6-1. 사용자는 매달 30일날마다 경고메세지를 전송받는다. F6-2. 시스템은 지난달 지출량과 이번 달 지출량을 비교한다. F6-3. 시스템은 과거에 비해 증가된 지출량을 화면에 출력한다.				

□ 비기능적 요구사항

주 기능	인터페이스 요구사항	번호	NF1	중요도	중
요구사항	NF1-1. 글씨체, 글씨 크기 등의 폰트기능을 사용자가 설정할 수 있다. NF1-1-1. 설정 메뉴에서 글씨체를 선택할 수 있다. NF1-1-2. 설정 메뉴에서 글씨 크기를 대, 중, 소로 조절할 수 있다. NF1-2. 월간 데이터를 한눈에 보기 쉽게 이미지 파일을 사용할 수 있어야 한다. NF1-2-1. 내역 정보를 계산하여 통계 결과를 그래프로 출력한다. NF1-3. 내역 기록 중에 입력 오류가 발생하면 오류 내용 알림이 출력된다. NF1-3-1. 금액, 날짜 칸에 숫자 외의 문자 입력 시 오류를 알린다.				

주 기능	안전성 요구사항	번호	NF2	중요도	상
요구사항	NF2-1. 등록된 사용자만이 시스템에 접근할 수 있어야 한다. NF2-1-1. 어플리케이션 실행 시 개인 비밀번호를 입력한다. NF2-2. 서버에 보관할 때 자료의 정보를 암호화하여 보관한다. NF2-3 자료의 백업 주기는 12시간 이내에 되어야 한다.				

주 기능	성능 요구사항	번호	NF3	중요도	중
요구사항	NF3-1. 애플리케이션 실행 시, 로딩 시간은 10초 이내여야 한다. NF3-2. 화면 전환 시, 도표나 그래프와 같은 이미지 출력 로딩이 2초 내에 끝나야 한다. NF3-3. 내역을 입력하면 3초 내에 정보가 통계 결과에 반영되어야 한다.				

주 기능	조직 요구사항	번호	NF4	중요도	상
요구사항	NF4-1. 구글플레이 및 앱스토어에 무료 다운로드 배포를 할 수 있다. NF4-2. 소프트웨어 구현 시, Java 및 android SDK를 사용한다.				

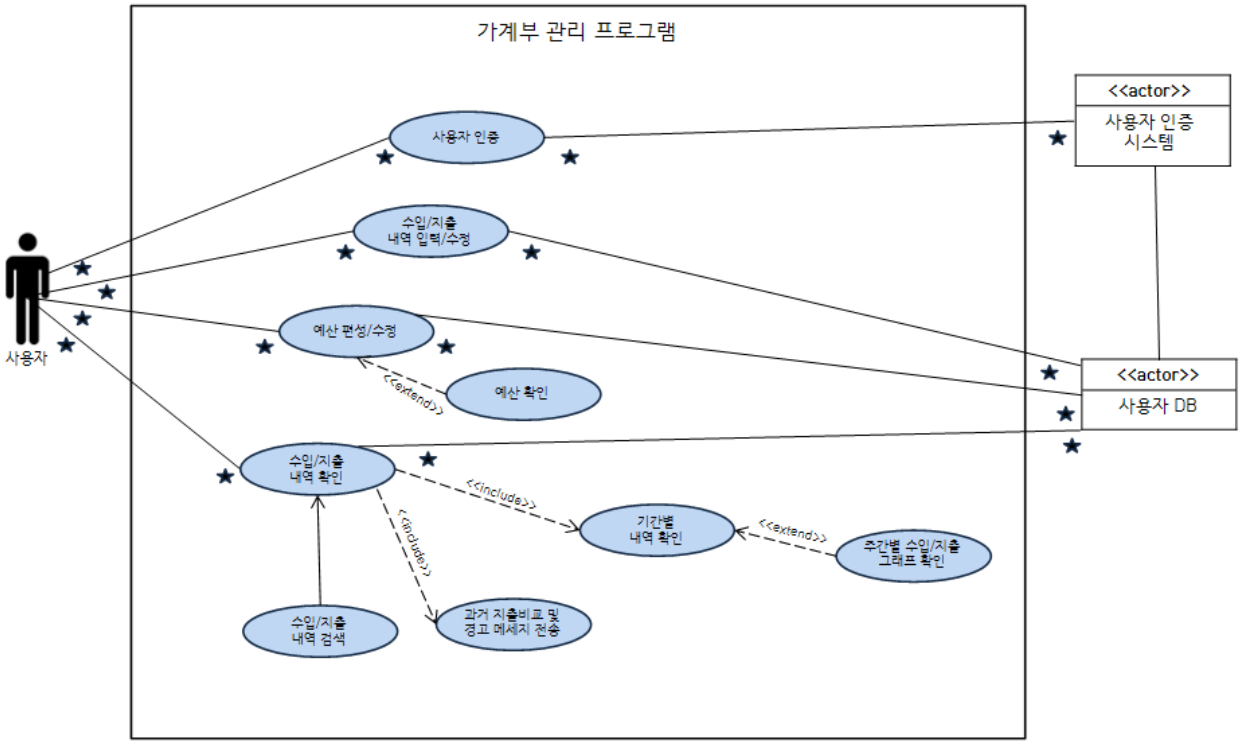
기능적 모델링 – 액터 리스트

번호	액터	설명
1	사용자	개인 자산을 애플리케이션을 통해 더 효율적이고 편리하게 관리하기 위한 도움이 필요한 사용자
2	사용자 인증시스템	사용자로부터 비밀번호를 입력 받아 처리하고 인증하는 시스템
3	사용자 DB	사용자의 수입/지출 내역, 예산 편성/수정 내역, 개인정보 등 모든 정보를 저장한다.

기능적 모델링 – Use Case 설명서

번호	Use Case	설명	관련 액터(Use Case) 및 방향
1	사용자 인증	사용자를 인증하는 Use Case	
2	예산 편성/수정	사용자의 예산을 관리하기 위해 계획을 편성하고 수정하는 Use Case	
3	예산 확인	사용자가 편성한 예산 계획 내역을 확인하는 Use Case	<<extend>> 예산 편성/수정
4	수입/지출 내역 입력 및 수정	수입/지출 내역을 입력하고 수정하는 Use Case	
5	주간별 수입/지출 그래프 확인	주간별 수입/지출에 대한 금액을 통계 그래프로 나타내어 사용자가 확인할 수 있도록 하는 Use Case	<<extend>> 기간별 내역 확인
6	기간별 내역 확인	일간, 주간, 월간, 연간 수입/지출에 대한 내역 및 종류를 확인하는 Use Case	
7	수입/지출 내역 검색	사용자의 수입/지출 내역을 검색하는 Use Case	Generalization 수입/지출 내역 확인
8	기간별 내역 확인	일간, 주간, 월간, 연간 수입/지출에 대한 내역 및 종류를 확인하는 Use Case	<<include>> 과거 지출 비교 및 경고메세지 전송 / 기간별 내역 확인
9	과거 지출 비교 및 경고메시지 전송	이전 기간과 지출을 비교하여 넘으면 경고메시지를 전송하는 Use Case	

기능적 모델링 - Use case Diagram



구조적 모델링 - CRC Card

Front :

Class Name : 사용자	**ID** : 1	**Type** : Concrete, Domain
Description : 사용자 개인 정보를 입력 및 수정하고 예산 편성/수정, 수입/지출 내역 생성 등 여러 기능을 호출하는 클래스입니다.		**Associated Use Cases** : 5

Responsibilities	Collaborators
- 인증하기 - 수입/지출 내역 생성 - 수입/지출 내역 수정 - 수입/지출 내역 확인 - 예산 편성 생성 - 예산 편성 확인 - 예산 편성 수정	- 사용자 인증 - 사용자 데이터베이스 - 예산 편성 내역 - 수입/지출 내역 - 수입/지출 내역 확인

Back :

Attributes :

- 이름 : string
- 이메일 : string
- 전화번호 : integer

Relationship :

Generalization (a-kind-of) :

Aggreagtion (has-parts) :

Other Association : 사용자 인증, 사용자 데이터베이스, 예산 편성 내역, 수입/지출 내역, 수입/지출 내역 확인

Front :

Class Name : 사용자 인증	ID : 2	Type : Concrete, Domain
Description : 사용자 정보를 바탕으로 올바른 사용자 여부를 판단하는 클래스입니다.		Associated Use Cases : 2
Responsibilities - 인증 비밀번호 비교 - 인증 결과 출력	Collaborators - 사용자 - 사용자 데이터베이스	

Back :

Attributes : - 사용자 입력 비밀번호 : string
Relationship : Generalization (a-kind-of) : Aggreagtion (has-parts) : Other Association : 사용자, 사용자 데이터베이스

Front :

Class Name : 사용자 데이터 베이스	ID : 3	Type : Concrete, Domain
Description : 사용자 개인 정보 및 수입/지출 내역 등 사용자의 정보를 보내고 저장하는 클래스입니다.		Associated Use Cases : 5

Responsibilities	Collaborators
- 사용자 정보 보내기	- 사용자 - 사용자 인증 - 예산 편성 내역 - 수입/지출 내역 - 수입/지출 내역 확인

Back :

Attributes :
- DB_인증 비밀번호 : string
- 기간 : int
- 사용자 : object
- 수입/지출 내역 : object list
- 예산 편성 내역 : object list

Relationship :

Generalization (a-kind-of) :

Aggreagtion (has-parts) :

Other Association : 사용자, 사용자 인증, 예산 편성 내역, 수입/지출 내역, 수입/지출 내역 확인

Front :

Class Name : 예산 편성 내역	ID : 4	Type : Concrete, Domain
Description : 예산 금액, 종류, 기간, 내용을 입력하여 예산 편성을 계획하는 클래스입니다.		Associated Use Cases : 2
Responsibilities - 예산 내용 및 기간 입력 - 예산 정보 요청 - 예산 출력 - 예산 종류 및 금액 입력		Collaborators - 사용자 - 사용자 데이터베이스

Back :

Attributes : - 예산 금액 : int - 예산 종류 : bool - 기간 : int - 내용 : string
Relationship : Generalization (a-kind-of) : Aggreagtion (has-parts) : Other Association : 사용자, 사용자 데이터베이스

Front :

Class Name : 수입/지출 내역	ID : 5	Type : Concrete, Domain
Description : 사용자가 자신의 수입/지출 내역을 입력하는 클래스입니다.		Associated Use Cases : 2
Responsibilities - 종류 및 금액 입력 - 내용 및 날짜 입력	Collaborators - 사용자 - 사용자 데이터베이스	

Back :

Attributes : - 수입/ 지출 : bool - 수입/지출 내용 : string - 금액 : int - 날짜 : date
Relationship : Generalization (a-kind-of) : Aggreagtion (has-parts) : Other Association : 사용자, 사용자 데이터베이스

Front :

Class Name : 수입/지출 내역 확인	ID : 6	Type : Concrete, Domain
Description : 사용자가 자신의 수입/지출 내역을 확인하는 클래스입니다.		Associated Use Cases : 4
Responsibilities - 전체 내역 확인 - 기간 입력 - 내역 검색 - 내역 출력	Collaborators - 사용자 - 사용자 데이터베이스 - 내역 그래프 출력 - 경고 메세지 전송	

Back :

Attributes : - 수입/지출 내역 : object list - 검색용 내용 키워드: string - 보유 자산 금액 : int - 기간 : int
Relationship : Generalization (a-kind-of) : Aggreagtion (has-parts) : 내역 그래프 출력, 경고 메세지 전송 Other Association : 사용자, 사용자 데이터베이스

Front :

Class Name : 내역 그래프 출력	ID : 7	Type : Concrete, Domain
Description : 수입/지출 내역 정보를 그래프의 형식으로 출력하는 클래스입니다.		Associated Use Cases : 1
Responsibilities - 그래프 정보 요청 - 내역 그래프 출력	Collaborators - 수입/지출 내역 확인	

Back :

Attributes : - 수입/지출 내역 리스트 : object list
Relationship : Generalization (a-kind-of) : Aggreagtion (has-parts) : 수입/지출 내역 확인 Other Association :

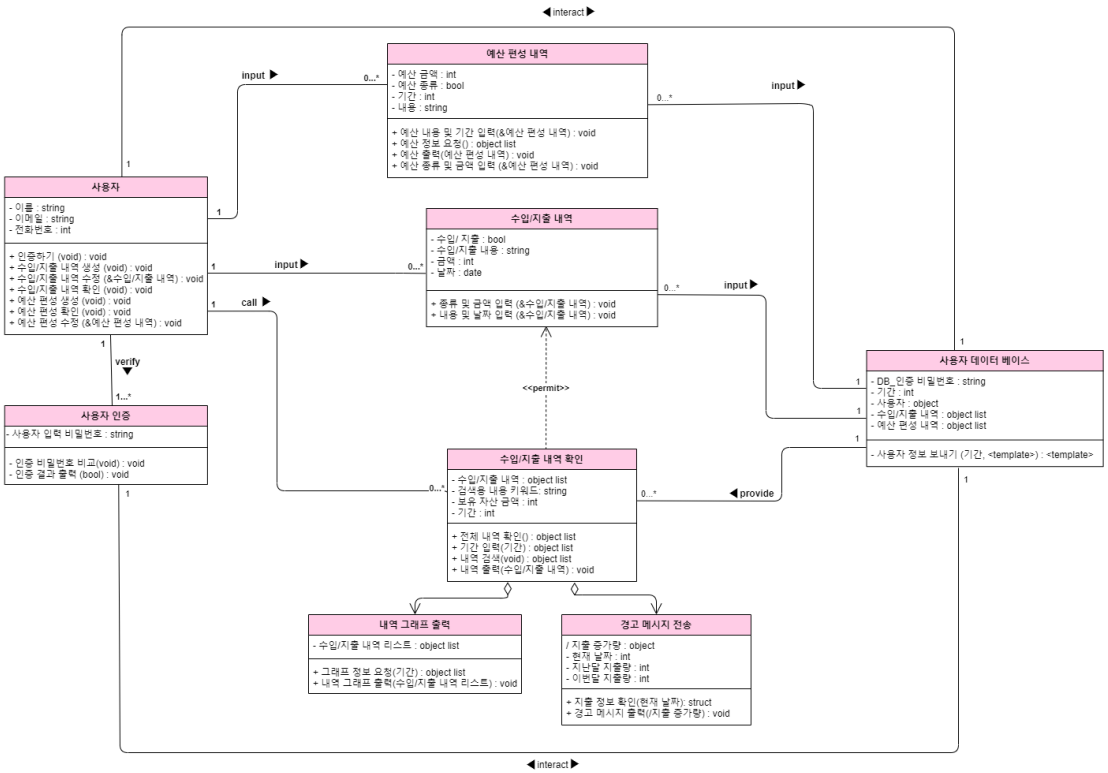
Front :

Class Name : 경고 메시지 전송	ID : 8	Type : Concrete, Domain
Description : 이전 기간과 지출을 비교하여 지출량이 이전보다 높으면 경고 메시지를 전송하는 Use Case		Associated Use Cases : 1
Responsibilities - 지출 정보 확인 - 경고 메시지 출력	Collaborators - 수입/지출 내역 확인	

Back :

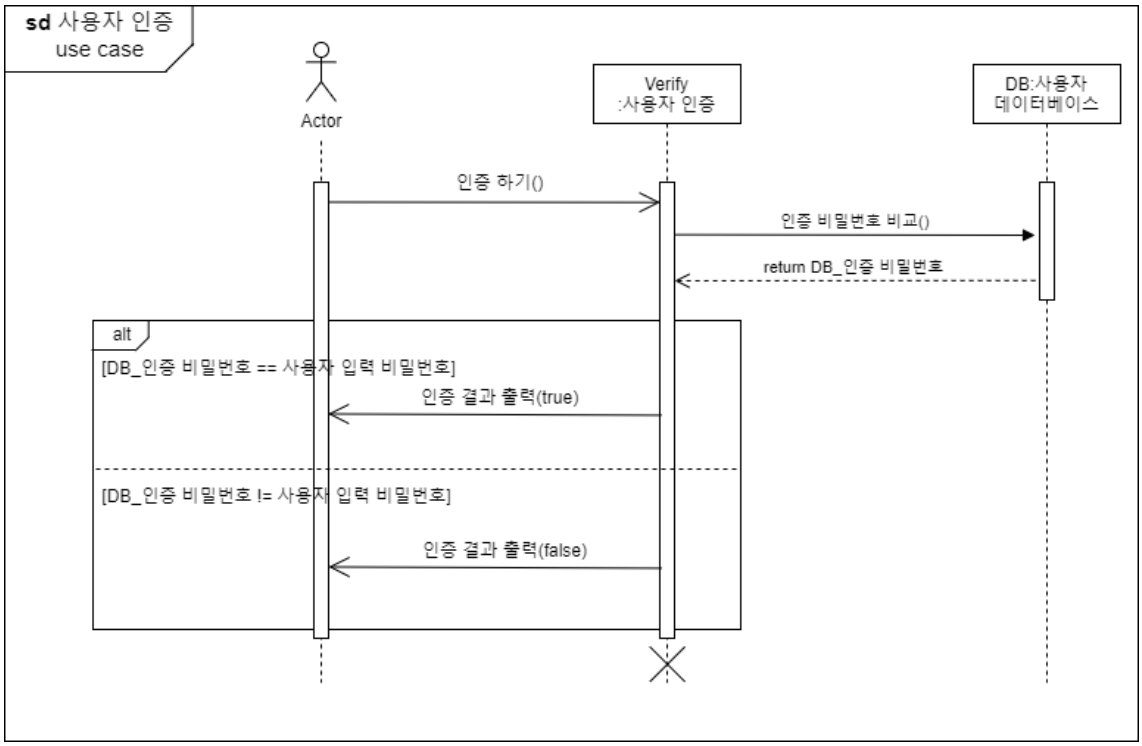
Attributes : / 지출 증가량 : object - 현재 날짜 : int - 지난달 지출량 : int - 이번달 지출량 : int
Relationship : Generalization (a-kind-of) : Aggreagtion (has-parts) : 수입/지출 내역 확인 Other Association :

구조적 모델링 - Class Diagram

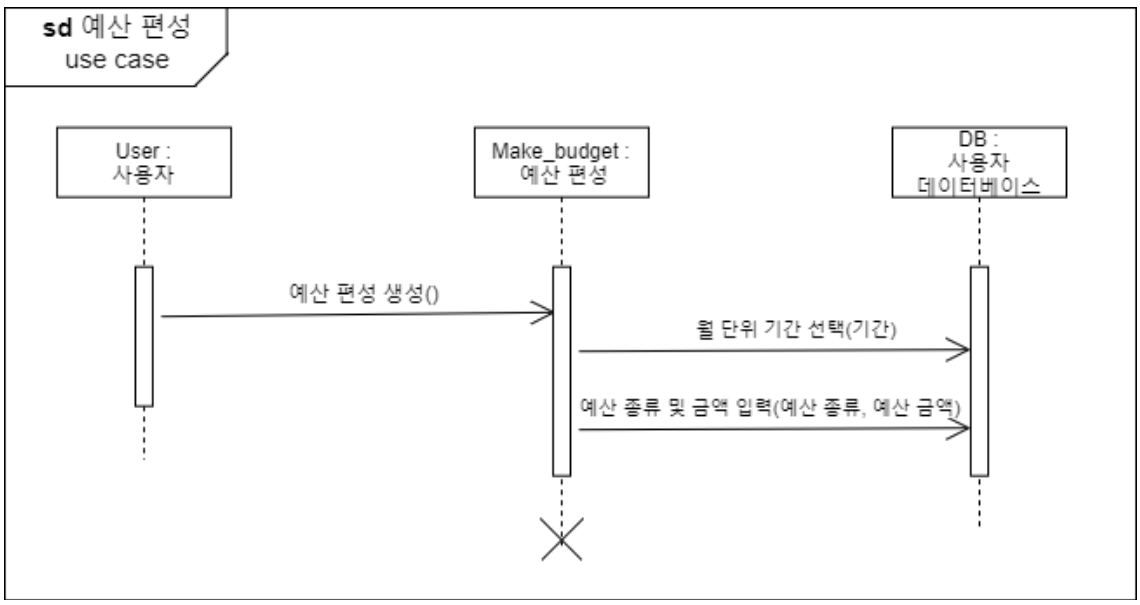


행위 모델링 – Sequence Diagram

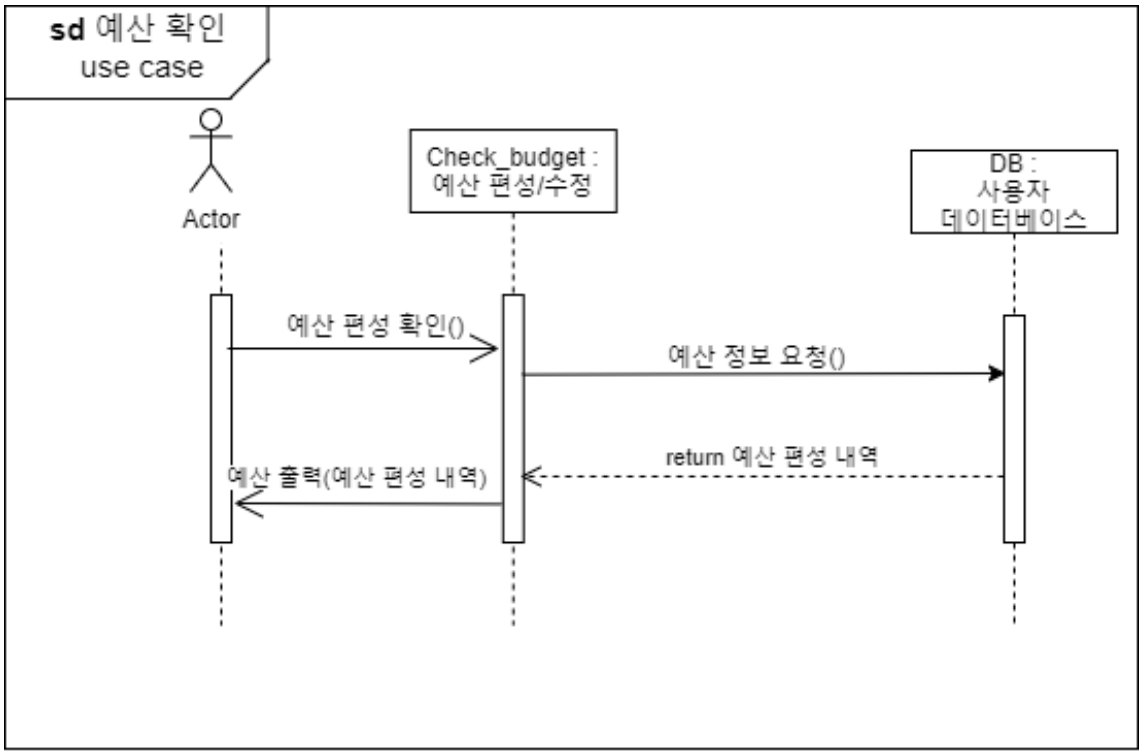
#1 사용자 인증



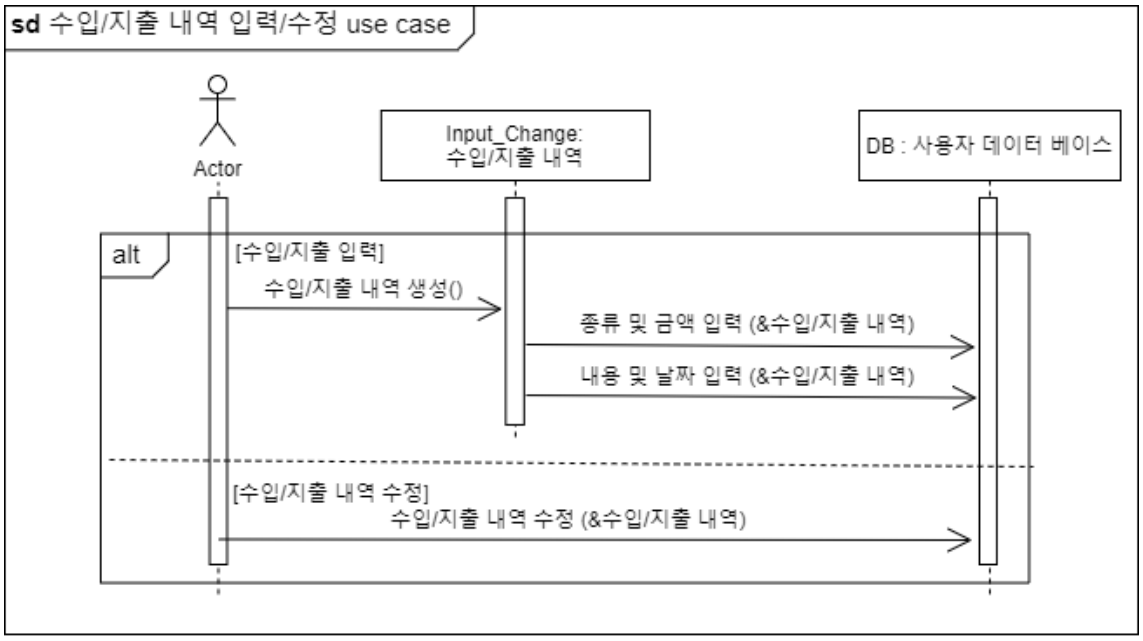
#2 예산 편성/수정



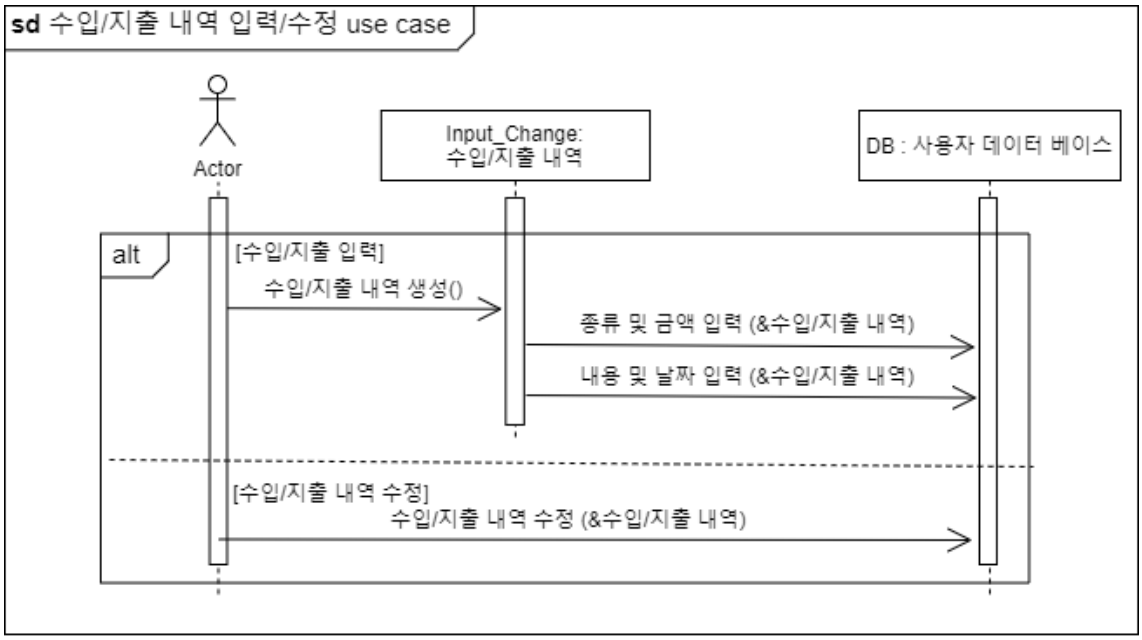
#3 예산 확인



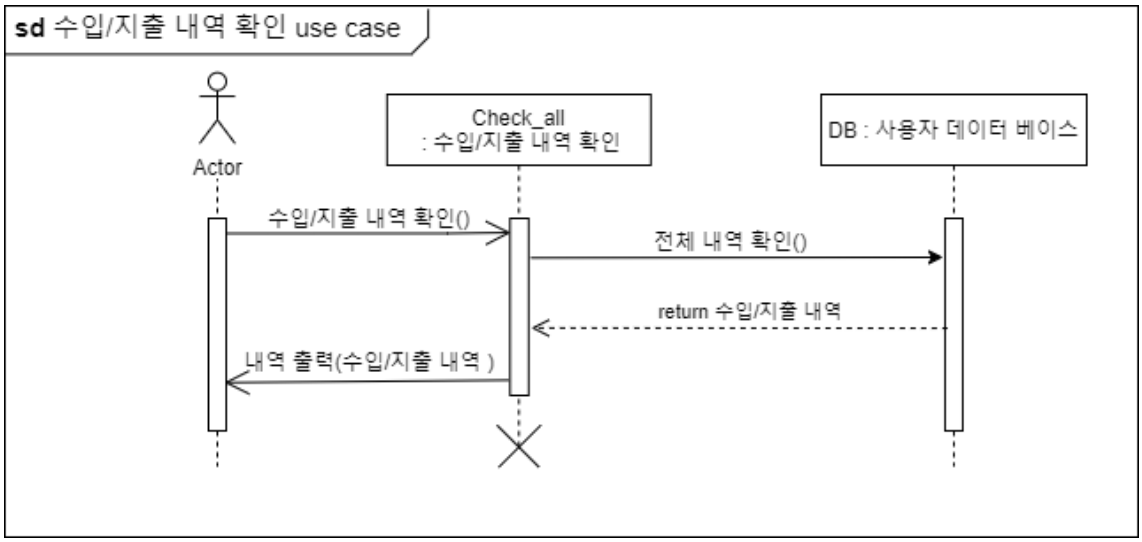
#4 수입/지출 내역 입력/수정



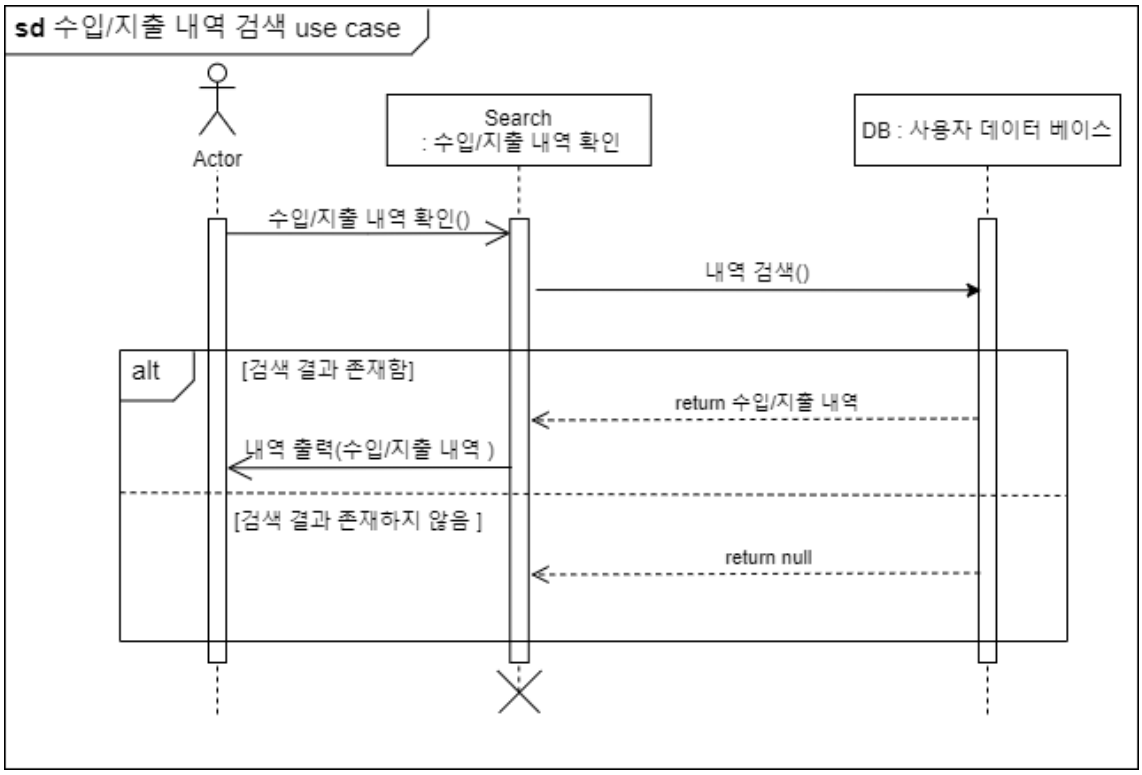
#5 주간별 수입/지출 그래프 확인



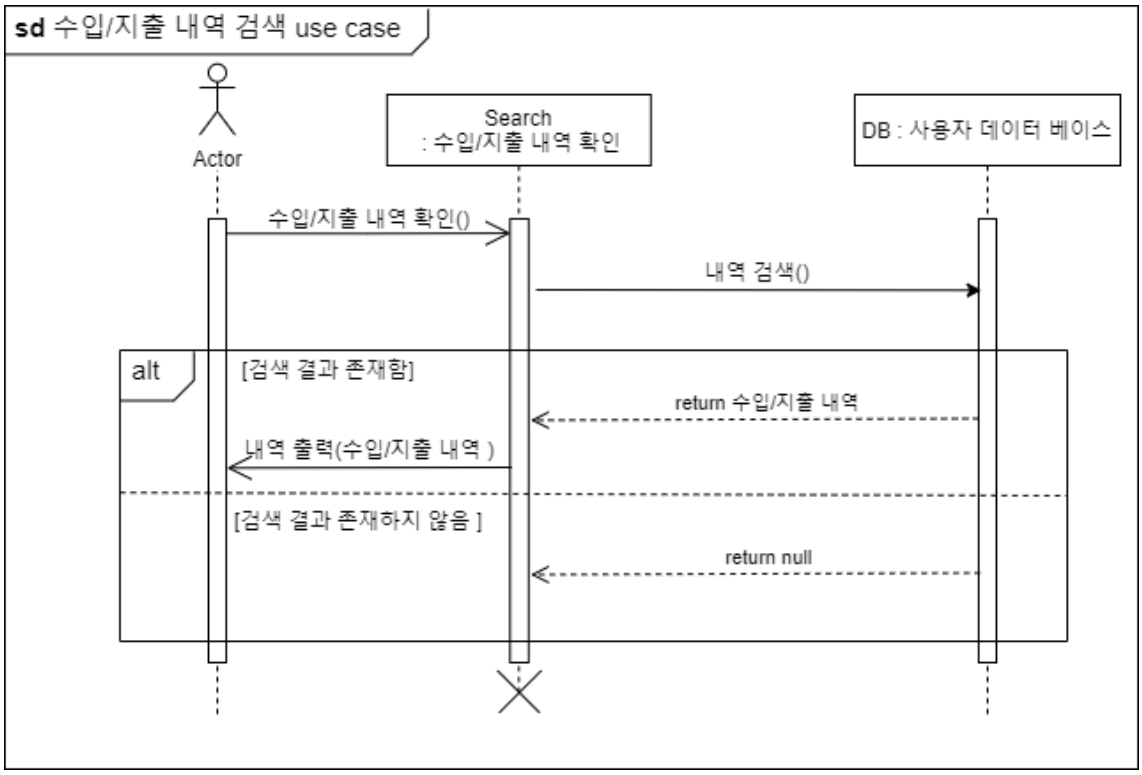
#6 수입/지출 내역 확인



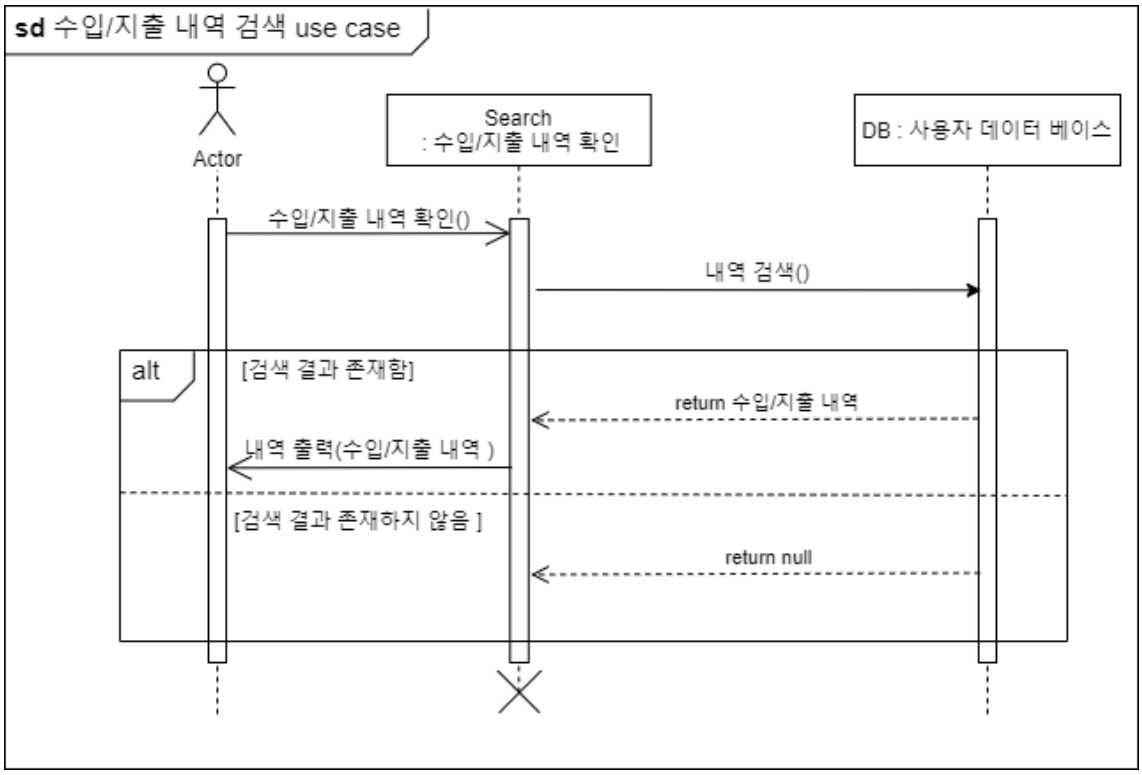
#7 수입/지출 내역 검색



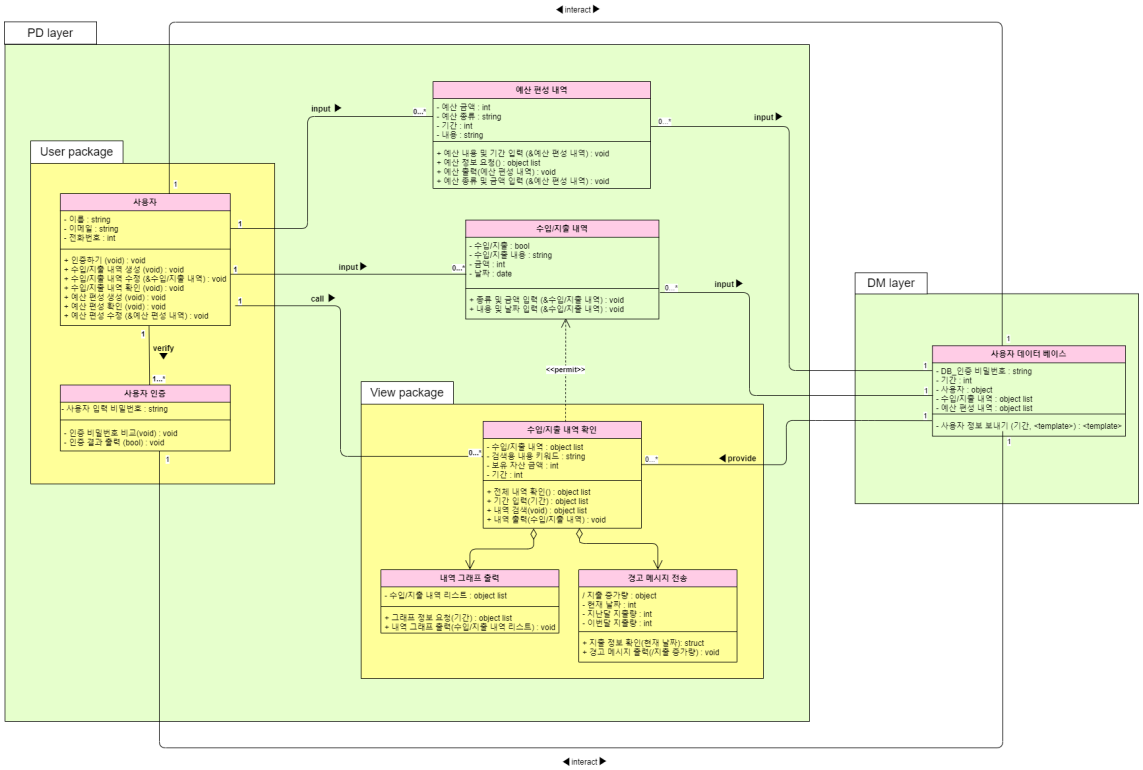
#8 기간별 내역 확인



#9 경고 메세지 전송


sd 수입/지출 내역 검색 use case
Actor
Search
: 수입/지출 내역 확인
DB : 사용자 데이터 베이스
수입/지출 내역 확인()
내역 검색()
alt
[검색 결과 존재함]
return 수입/지출 내역
내역 출력(수입/지출 내역)
[검색 결과 존재하지 않음]
return null

Move to Design – Package Diagram



Move to Design – Method Specification

Method Name	인증 비밀번호 비교	Class Name	사용자 인증
Brief Desc.	사용자로부터 인증 비밀번호를 입력 받고 DB에 있는 DB_인증 비밀번호를 요청하는 메소드		
Trigger	사용자의 인증하기 버튼 클릭		
Input Para.	X		
Return value	X		
Pre-Condition	Actor가 인증하기 버튼을 클릭		
Post-Condition	인증결과 출력		
세부 처리 로직			
Pseudocode			

```
void 인증 비밀번호 비교()
{
  input 사용자 입력 비밀번호
  string DB_pw = 사용자 데이터베이스 :: 사용자 정보 보내기(DB_인증 비밀번호)
  if(DB_pw == 사용자 입력 비밀번호)
    사용자 인증 :: 인증 결과 출력(true)
  else
    사용자 인증 :: 인증 결과 출력(false)
}
```

Method Name	사용자 정보 보내기	Class Name	사용자 데이터베이스
Brief Desc.	다른 객체가 본인들의 일처리를 위해 필요한 데이터를 받기 위해 데이터 요청을 하면 해당 데이터를 보내주는 메소드		
Trigger	다른 객체가 본인들의 일처리를 위해 필요한 데이터를 받기 위해 해당 메소드 호출		
Input Para.	기간(trigger가 전달하는 기간에 대한 변수), template 사용자 정보		
Return value	template 사용자 정보		
Pre-Condition	다른 객체가 본인들의 일처리를 위해 필요한 데이터를 받기 위해 해당 메소드 호출		
Post-Condition	메소드를 요청한 객체로 데이터를 보낸 후 해당 객체에서 데이터 사용		

세부 처리 로직

Pseudocode

```
(template 사용자 정보) 사용자 정보 보내기(기간, template 사용자 정보)
{
  if(기간==현재 날짜){
    기간 = 지난달 1일 ~ 현재 날짜
    while(기간)
      list info += 수입/지출 내역
      return info
  }
  if(search(template 사용자 정보)){
    case : template 사용자 정보 == DB_인증 비밀번호
      return DB_인증 비밀번호
    case : template 사용자 정보 == 예산 편성/수정
      return 모든 예산 편성/수정
    case : template 사용자 정보 == 수입/지출 내역
      while(기간)
          list info += 수입/지출 내역
      return info
    case : template 사용자 정보 == 검색용 내용 키워드
        return 일치하는 모든 수입/지출 내역
  }
  else{
    return null
  }
}
```

Method Name	내역 출력 ()	Class Name	수입/지출 내역 확인
Brief Desc.	사용자 데이터베이스의 모든 수입/지출 내역을 출력하는 메소드		
Trigger	사용자 데이터 베이스의 '사용자 정보 보내기()'		
Input Para.	object list 수입/지출 내역 리스트		
Return value	X		
Pre-Condition	수입/지출 확인 클래스의 '전체 내역 확인()' 호출, 사용자 데이터베이스 클래스의 '사용자 정보 보내기()' 호출		
Post-Condition	X		
세부 처리 로직			
Pseudocode			

```
void 내역 출력(list 수입/지출 내역 리스트) {
  result = 수입/지출 내역 리스트;
  if(result == NULL)
  else{
    int i = 0;
    while(i < result.length)  {
      if(result[i].수입/지출 == true)
          printf("종류 : 수입\n");
      else(result[[i].지출 == false)
          printf("종류 : 지출\n);
      printf("날짜 : %d / %d -> 금액 : %d", result[[i].날짜월, result[[i].날짜일, reesult[[i].금액);
      printf("내용 : %s\n", result.[i].수입/지출 내용);
      i++;
    }
  }
}
```

Method Name	내역 검색 ()	Class Name	수입/지출 내역 확인
Brief Desc.	사용자가 내역 내용의 키워드를 입력하여 찾고자 하는 특정 내역을 검색하고 해당 정보를 출력하는 메소드		
Trigger	사용자 클래스의 '수입/지출 내역 확인()'		
Input Para.	X		
Return value	object list 수입/지출 내역 리스트		
Pre-Condition	사용자의 내역 내용 키워드 입력		
Post-Condition	사용자 데이터베이스 클래스의 검색조건에 해당하는 데이터 전송		
세부 처리 로직			
Pseudocode			

```
list 내역 검색() {
  scanf(검색용 내용 키워드);
  list search = 사용자 데이터 베이스 :: 사용자 정보 보내기 (내용 키워드);
  if(search == NULL)
    return null;
  return search;
  }
```

Method Name	수입/지출 내역 수정	Class Name	사용자
Brief Desc.	사용자가 수입/지출 내역에 대한 수정할 내용이 있을 시 호출하는 메소드		
Trigger	사용자가 수입/지출 내역을 보고 수정 버튼을 누른다.		
Input Para.	&수입/지출 내역		
Return value	void		
Pre-Condition	수정 버튼 클릭		
Post-Condition	X		
세부 처리 로직			
Pseudocode			

```
void 수입/지출 내역 수정(&수입/지출 내역)
{
  input(new_수입/지출)
  수입/지출 내역.수입/지출 = new_수입/지출
  input(new_수입/지출 내용)
  수입/지출 내역.수입/지출 내용 = new_수입/지출 내용
  input(new_금액)
  수입/지출 내역.금액 = new_금액
  input(new_날짜)
  수입/지출 내역.날짜 = new_날짜
}
```

Method Name	내역 그래프 출력( )	Class Name	내역 그래프
Brief Desc.	사용자가 어떤 기간 동안의 내역 그래프 출력을 요청하여 데이터베이스로부터 해당 데이터를 받아 사용자에게 그래프를 출력하는 메소드		
Trigger	사용자 데이터베이스 클래스의 사용자 정보 보내기()		
Input Para.	object list 해당 기간만큼의 수입/지출 내역 리스트(object들의 배열)		
Return value	X		
Pre-Condition	사용자의 기간 입력, 사용자 데이터베이스 클래스의 해당 데이터 전송이 필요		
Post-Condition	X		
세부 처리 로직			
Pseudocode			

```
void 내역 그래프 출력(수입/지출 내역 리스트){
    x_label = 기간
    y_label = 금액
    points[(수입/지출 내역 리스트).length]
    for(int i=0 ; i < (수입/지출 내역 리스트).length ; i++)
        points[i] = (x_label, y_label)
        print(points[i])
}
```

Method Name	지출 정보 확인( )	Class Name	경고 메시지 전송
Brief Desc.	사용자의 요청 없이 시스템 상에서 월 말(매달 30일)이 되면 사용자 데이터베이스 클래스에 지출 정보 확인을 위한 데이터를 요청하는 메소드		
Trigger	월 말(매달 30일)이 되는 날짜에 경고 메시지 전송 클래스의 생성자 호출		
Input Para.	date 현재 날짜(매달 30일)		
Return value	struct 이번달 지출량, 지난달 지출량		
Pre-Condition	월 말(매달 30일)이 되는 날짜		
Post-Condition	X		
세부 처리 로직			
Pseudocode			

```
struct 지출 정보 확인(현재 날짜(매달 30일)){
    this_date = 현재 날짜(매달 30일)
    past_date = 지난달 1일
    지출량 = (이번달 지출량, 지난달 지출량)
    지출량 = 사용자 데이터베이스 :: 사용자 정보 보내기(past_date, this_date)
    return 지출량
}
```

Method Name	예산 종류 및 금액 입력()	Class Name	예산 편성/수정
Brief Desc.	예산 편성 생성 시, 예산 계획의 내용을 수입/지출 종류와 금액을 입력하는 메소드		
Trigger	사용자로부터 예산 편성 생성() 호출		
Input Para.	&예산 편성 내역		
Return value	X		
Pre-Condition	예산 편성 생성() 호출 필요		
Post-Condition	X		
세부 처리 로직			
Pseudocode			

```
// budget : 사용자::예산 편성 생성()에 의해 만들어진 DB에 존재하는 예산 편성/수정 object 선언
void 예산 종류 및 금액 입력(& 예산 편성 내역) {
  bool inout;
  int money;
  budget = 예산 편성/수정;
  choice 수입 or 지출;
  switch(종류) {
   case : 수입
     inout = true;
     break;
   case : 지출
     inout = false;
     break;
  }
  scanf(money);
  budget . 종류 = inout;
  budget . 금액 = money;

}
```

Method Name	예산 편성 수정	Class Name	사용자
Brief Desc.	사용자가 예산편성 대한 수정할 내용이 있을 시 호출하는 메소드		
Trigger	사용자가 예산 편성을 보고 수정버튼을 누른다		
Input Para.	&예산 편성 내역		
Return value	void		
Pre-Condition	수정 버튼 클릭		
Post-Condition	X		
세부 처리 로직			
Pseudocode			

```
void 예산 편성 수정(&예산 편성 내역)
{
 input(new_기간)
 예산 편성 내역 . 기간 = new_기간
 input(new_예산 종류)
 예산 편성 내역 . 예산 종류 = new_예산 종류
 input(new_예산 금액)
 예산 편성 내역. 예산 금액 = new_예산 금액
 input(new_내용)
 예산 편성 내역 . 내용 = new_내용;
}
```