

년월일: 2019-05-20	페이지: 1
--------------------	-----------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

상세 설계서

그룹 운동 커뮤니티 어플리케이션 (DIETA)

Ver. 2

2019.05.20

한국외국어대학교
융복합소프트웨어학과
2팀(Make U)

년월일: 2019-05-20	페이지: 2
--------------------	-----------



상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

문서 정보

구분	소속	성명	날짜	서명
작성자	한국외국어대학교	최진용	2019-05-19	
	한국외국어대학교	이정훈	2019-05-19	
	한국외국어대학교	윤지영	2019-05-19	
	한국외국어대학교	김영인	2019-05-19	
	한국외국어대학교	정진원	2019-05-19	
검토자	한국외국어대학교	이정훈	2019-05-20	
사용자				
승인자	한국외국어대학교	홍진표		

년월일: 2019-05-20	페이지: 3
--------------------	-----------



상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

머리말

본 문서는 장고 프레임워크로 제작할 운동 커뮤니티 웹 어플리케이션 서비스 소개와 목적을 설명하고, 본 시스템을 구축하기 위한 시스템 상세 설계를 기술한다.

년월일: 2019-05-20	페이지: 4
--------------------	-----------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

목차

1.개요	5
1.1 서비스 정의	5
1.2 문서의 목적	5
2. 시스템 구성요소 및 핵심기능	6
2.1 시스템 구성요소	6
2.2 핵심기능	7
3. 소프트웨어 설계	8
3.1 인터페이스 어플리케이션 sequence diagram	8
3.2 인터페이스 어플리케이션 상세설계 목록	9
3.3 인터페이스 어플리케이션 항목별 상세설계	10
4. 데이터베이스 설계	23
5. 상세설계 산출물	33
5.1 시스템 종합 설계도	33
5.2 Use case diagram	34
6. 담당업무	35

년월일: 2019-05-20	페이지: 5
--------------------	-----------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

1.개요

1.1 서비스 정의

▶ 자사가 서비스하는 'Dieta'는 GPS정보를 활용하여 그룹 매칭을 지원하고, 단체 운동에서 활동량을 기준으로 게임 랭킹 시스템을 적용한다. 이를 통해 함께 하는 다이어트의 즐거움과 지속성을 도모할 수 있는 환경을 제공한다.

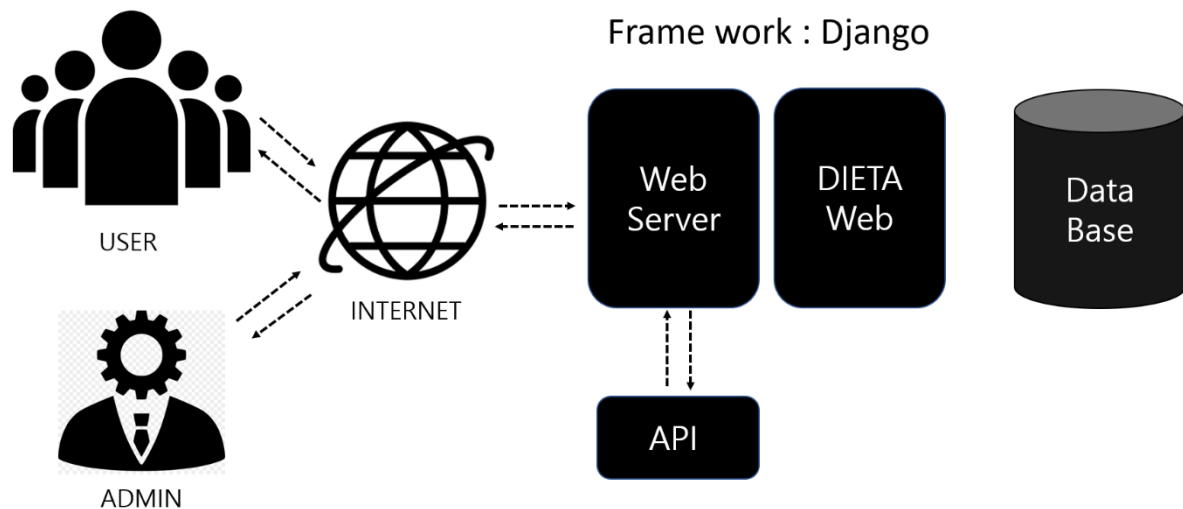
1.2 문서의 목적

▶ 본 문서는 자바 프로그래밍 기반 운동 커뮤니티 'Dieta' 어플리케이션 개발 사업에 관련해 시스템, 기능, 성능, 인터페이스, 데이터, 테스트, 보안, 품질, 제약, 관리 및 지원 등 개발관련 요구사항을 토대로 설계 시 필요한 상세설계 부분에 대하여 중점적으로 작성한다.

2. 시스템 구성요소 및 핵심기능

2.1 시스템 구성요소

본 System의 구성요소에는 User, web Interface, Database가 포함된다.



아래에 구성요소별로 정리한 표가 있다.

Dieta system 구성요소		
User	Web interface	Database
<ul style="list-style-type: none"> ▶ User는 그룹 운동 매칭 app 'Dieta'를 이용하는 대상이다. ▶ User는 본 System을 이용함으로써 그룹 매칭을 통한 집단 운동으로 운동의 지속성을 도모한다. 	<ul style="list-style-type: none"> ▶ 웹 브라우저를 기반으로 제작하여 각종 기능과 interface를 담당하고 있다. ▶ 자세한 기능으로 운동 진행, 칼로리 계산, gps 정보를 활용한 장소 추천, 그룹 매칭 등 각종 구성요소가 있다. 	<ul style="list-style-type: none"> ▶ Server에서 운용되는 Database로 Interface App과 Data Language를 통해 정보를 주고 받아야 한다. ▶ 사용자 정보, GPS 정보, 날씨 정보 데이터 등이 필요하다.

년월일: 2019-05-20	페이지: 7
--------------------	-----------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

2.2 핵심 기능

1. 게시판 및 캘린더 기능

- 다이어트 웹 기반 커뮤니티로서의 기능 수행하기 위함.
- 그룹 운동 날짜를 선정하기 위함

2. Gps 및 날씨예보 api를 활용한 의사결정 지원

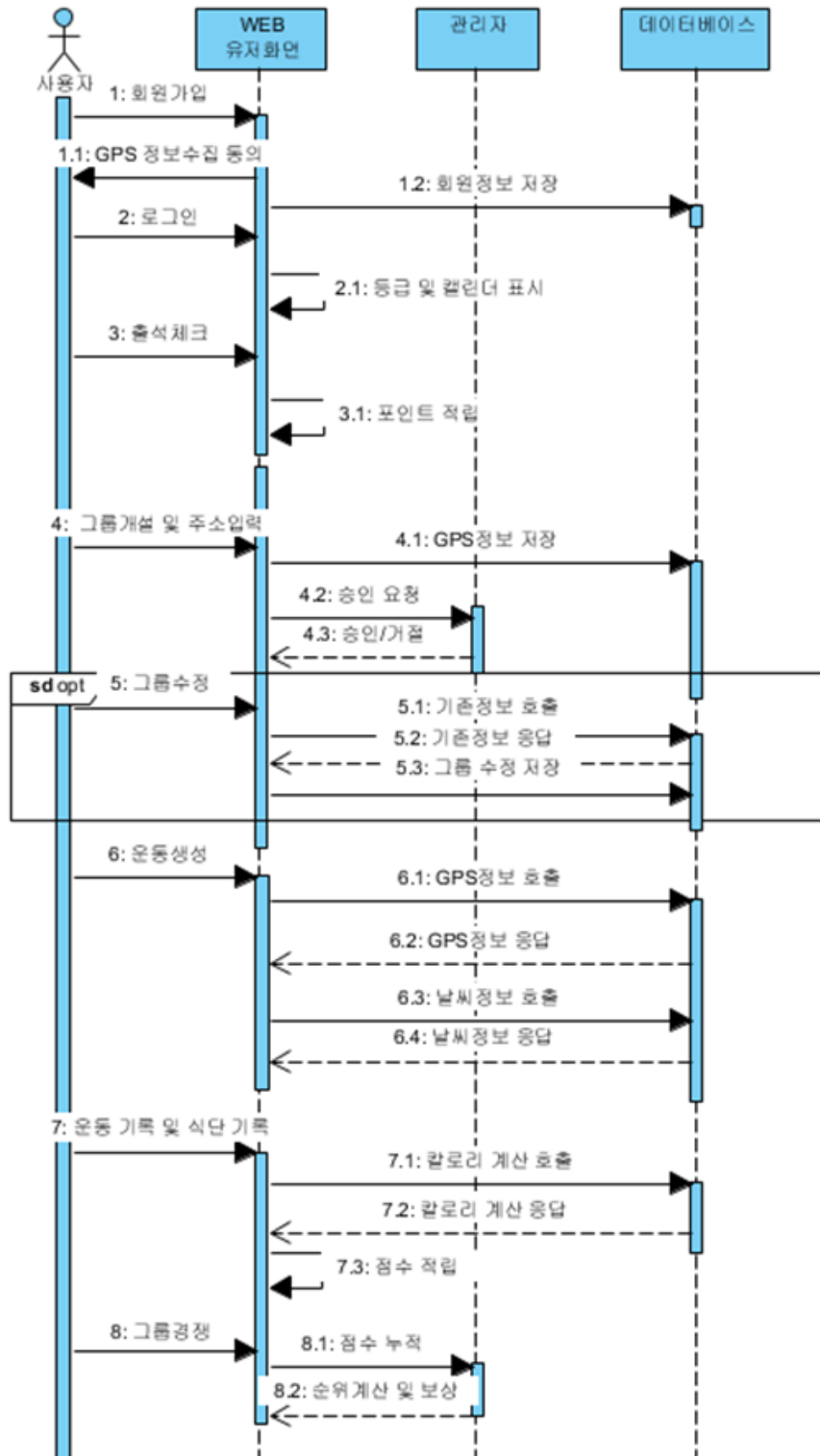
- 사용자가 운동 및 그룹 선택시 고려하는 환경 제공

3. 칼로리 계산을 통한 운동 평가 기능

4. 포인트 누적을 통한 그룹 경쟁 기능

3. 소프트웨어 설계

3.1 인터페이스 어플리케이션 sequence diagram



상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

3.2 인터페이스 어플리케이션 상세설계 목록

상세유형	요구사항 명	상세설계 ID	비고
인터페이스 부 상세 설계 내역 (SFD-I)	홈 메인 화면 제공	SFR-S-1	
	회원가입 기능 제공	SFR-S-2	
	로그인 기능 제공	SFR-S-3	
	로그아웃 기능 제공	SFR-S-4	
	게시판 글 목록 기능 제공	SFR-S-5	
	게시판 글 상세보기 기능 제공	SFR-S-6	
	게시판 글 작성 기능 제공	SFR-S-7	
	캘린더 기능 제공	SFR-S-8	
	캘린더 전, 후 달 이동 기능 제공	SFR-S-9	
	캘린더 이벤트 추가 기능 제공	SFR-S-10	
	회원 그룹 만들기 기능 제공	SFR-S-11	
	그룹 회원 추가 기능 제공	SFR-S-12	
	운동 모임 개설 기능 제공	SFR-S-13	
	GPS/날씨예보 api 활용 의사결정 지원	SFR-S-14	
	칼로리 계산을 통한 운동 평가 기능	SFR-S-15	
	포인트 누적을 통한 그룹 경쟁 기능	SFR-S-16	

년월일: 2019-05-20	페이지: 10
--------------------	------------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

3.3 인터페이스 어플리케이션 항목별 상세설계

1. 홈 메인 화면 제공 (SFR-S-1)

사용자가 홈 메인 화면을 제공받을 수 있다.

▶base.html에 사용자가 다른 기능을 사용하면서도 변하지 않는 상단바와 하단바를 만들어 주기 위해 block tag를 이용해준다. 다른 html파일을 만들어 줄 때 extends 'base.html'을 적어주면 그 html파일에 적용된다

<p>Base.html</p> <pre><!-- 상단바 --> {% block content %} {% endblock %} <!-- 하단바 --></pre>
<p>other.html</p> <pre>{% extends 'base.html' %} {% block content %} <!-- 관련 html태그 --> {% endblock %}</pre>

상단바에는 앱을 사용하면서 자신의 로그인 여부를 계속 알 수 있도록 만들어준다.

```
{% if user.is_authenticated %}

    <p>{{user.last_name}} {{user.first_name}} 님 환영합니다</p>

    <a href="{% url 'cl:signout' %}">로그아웃</a>

{% else %}

    <p>어서오세요! </p>

    <p><a href="{% url 'cl:signin' %}">로그인</a> <a href="{% url 'cl:signup' %}">회원가입</a></p>

{% endif %}
```

년월일: 2019-05-20	페이지: 11
--------------------	------------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

`user.is_authenticated()` 함수는 로그인 여부를 반환해주는 함수이다. `if`문을 이용해서 로그인 되어있을 때는 로그아웃, 로그인이 안 되어있으면 회원가입이나 로그인이 가능하도록 `a`태그를 만들어준다

하단바에는 우리 앱의 이름과 제작자 이름을 적어준다.

2. 회원가입 기능 제공 (SFR-S-2)

사용자가 이 앱에 가입할 수 있도록 한다

▶ `'django.contrib.auth',`

장고에 원래 설치되어 있는 앱을 이용해서 회원 가입 기능을 만들어준다.

▶ `from django.contrib.auth.models import User`

모델은 만들어져 있는 `user` 모델 클래스를 이용하고 `forms.py`로 입력창을 만들어준다.

▶ `widgets = {'password' : forms.PasswordInput() }`

이 때 비밀번호를 설정할 때, 위젯을 이용해 비밀번호를 보호한다.

▶ `views.py`에서는 사용자에게 회원가입폼 객체 생성 및 `html`파일을 전달해주고, 사용자가 채워 넣은 정보가 유효한지 확인해주는 `if`문을 통과하면

`User.objects.create_user`을 통해 유저를 추가해준다.

어서오세요!

[로그인](#) [회원가입](#)

회원 가입

사용자 이름:

비밀번호:

비밀번호 확인:

성:

Address:

150자 이하 문자, 숫자 그리고 @/./+/_만 가능합니다.

년월일: 2019-05-20	페이지: 12
--------------------	------------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

3. 로그인 기능 제공 (SFR-S-3)

사용자가 로그인을 할 수 있다.

```
▶class SigninForm(forms.models.ModelForm):

    class Meta:

        model = User

        fields = ['username', 'password']

        widgets = {'password' : forms.PasswordInput()}
```

Forms.py에 입력폼을 만들고 views.py에서 사용자가 입력한 값을 받는다

```
id = request.POST.get('username')

pw = request.POST.get('password')
```

```
▶u = authenticate(username=id,password=pw)
```

데이터베이스에 입력정보가 일치하면

```
login(request,u)
```

로그인을 해준다.

로그인

사용자 이름: 150자 이하 문자, 숫자 그리고 @/./+/_만 가능합니다.

비밀번호:

4. 로그아웃 기능 제공 (SFR-S-4)

사용자가 로그아웃을 할 수 있다.

▶Views.py에 클라이언트의 User정보를 지워주는 함수를 사용해준다.

```
logout(request)
```

김 영인 님 환영합니다!

[로그아웃](#)

년월일: 2019-05-20	페이지: 13
--------------------	------------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

5. 게시판 글 목록 기능 제공 (SFR-S-5)

그룹 사용자가 글 목록을 볼 수 있다.

```
▶ from django.views.generic.list import ListView
```

ListView(특정 모델클래스의 객체 목록을 다루는 기능이 있는 제네릭 뷰)를 이용해 html에 Post객체를 넘겨준다. ListView에서 paginate_by변수를 이용해 목록에 몇 개의 객체를 출력할지 설정할 수 있다.

```
▶ {% if page_obj.has_previous %}
    {% if page_obj.has_next %}
```

html에서 함수를 이용해 이전 페이지와 다음페이지 버튼을 만들어준다.

```
▶ <table width="90%">
    <tr>
        <th>글번호</th>
        <th>카테고리</th>
        <th>제목</th>
        <th>작성자</th>
        <th>작성일</th>
    </tr>
    {% for obj in objs %}
        <tr>
            <th>{{ obj.id }}</th>
            <th>{{ obj.c }}</th>
            <th><a href="{% url 'blog:detail' obj.id %}">{{ obj.title }}</a></th>
            <th>{{ obj.u }}</th>
            <th>{{ obj.pub_date }}</th>
        </tr>
    {% endfor %}
```

그리고 table태그로 모델클래스 안의 값을 나열해준다

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

이전 페이지		다음 페이지	
글번호	카테고리	제목	작성일
1	오늘의식단	저녁	ssn 2019년 5월 13일 8:54 오전
2	오늘의식단	그리지마요	ssn 2019년 5월 13일 8:56 오전
글쓰기			

6. 게시판 글 상세보기 기능 제공 (SFR-S-6)

그룹 사용자가 목록의 글을 읽을 수 있다.

```
▶ from django.views.generic.detail import DetailView
```

DetailView(데이터베이스에 특정 모델클래스에서 한 개의 객체를 추출하는 제네릭 뷰)를 views.py에 작성해준다.

```
▶ <a href="{% url 'blog:detail' obj.id %}">{{ obj.title }}</a>
```

목록부분의 html에서 a태그를 이용해 해당 객체의 id를

```
path('<int:pk>/', Detail.as_view(), name='detail')
```

pk변수로 views.py의 DetailView로 전해주고 html파일로 뿌려준다.

```
▶ <a download href="{f.f.url}">{{ f.file.name }}</a>

```

전해진 Post객체의 정보 중 파일은 a태그를 이용해 다운로드를 할 수 있도록 해주고 이미지는 img태그를 이용해 띄워준다.

[오늘의식단] 저녁

ssn / 2019년 5월 13일 8:54 오전

나는 오늘 다이어트를 위해 굶을 것이다!



[메인페이지로 돌아가기](#)

년월일: 2019-05-20	페이지: 15
--------------------	------------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

7. 게시판 글 작성 기능 제공 (SFR-S-7)

그룹 사용자가 카테고리를 선택해 글과 그림을 올릴 수 있다

```
▶ from django.views.generic.edit import FormView
```

FormView(폼클래스를 기반으로 사용자에게 입력공간을 제공하고 사용자가 입력한 데이터르 바탕으로 객체를 생성해주는 뷰)를 이용해 views.py에 작성해준다,

```
▶ files = forms.FileField(label='첨부파일', required=False)
```

```
images = forms.ImageField(label='이미지', required=False)
```

forms.py에 FormView 클래스와 연동할 폼클래스를 만들어준다. FileField, ImageField를 이용해서 첨부파일과 이미지파일을 올릴 수 있도록 해준다.

(모델클래스에 파일과 이미지 모델클래스를 만들어 주었다.)

이 때, 이미지파일 이용을 위해 pip install Pillow을 해줘야한다

```
▶ def form_valid(self, form):
    p = form.save(commit=False)
    p.u = self.request.user
    p.save()
```

폼의 사용자 입력이 유효한 경우, 저장한다.

글쓰기

C:
오늘의식단 ▼

Title:
저녁

Content:
나는 오늘 다이어트를 위해 굶을 것이다!

첨부파일:
파일 선택
201600621_...기소개서.docx

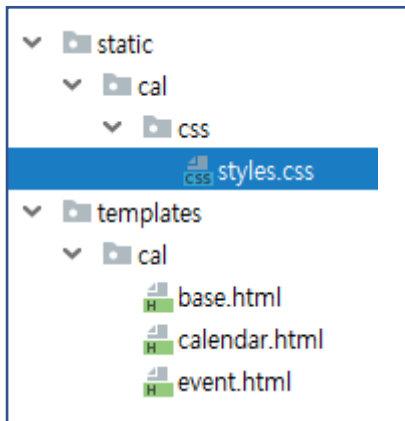
이미지:
파일 선택
20171002_164746.png

글쓰기

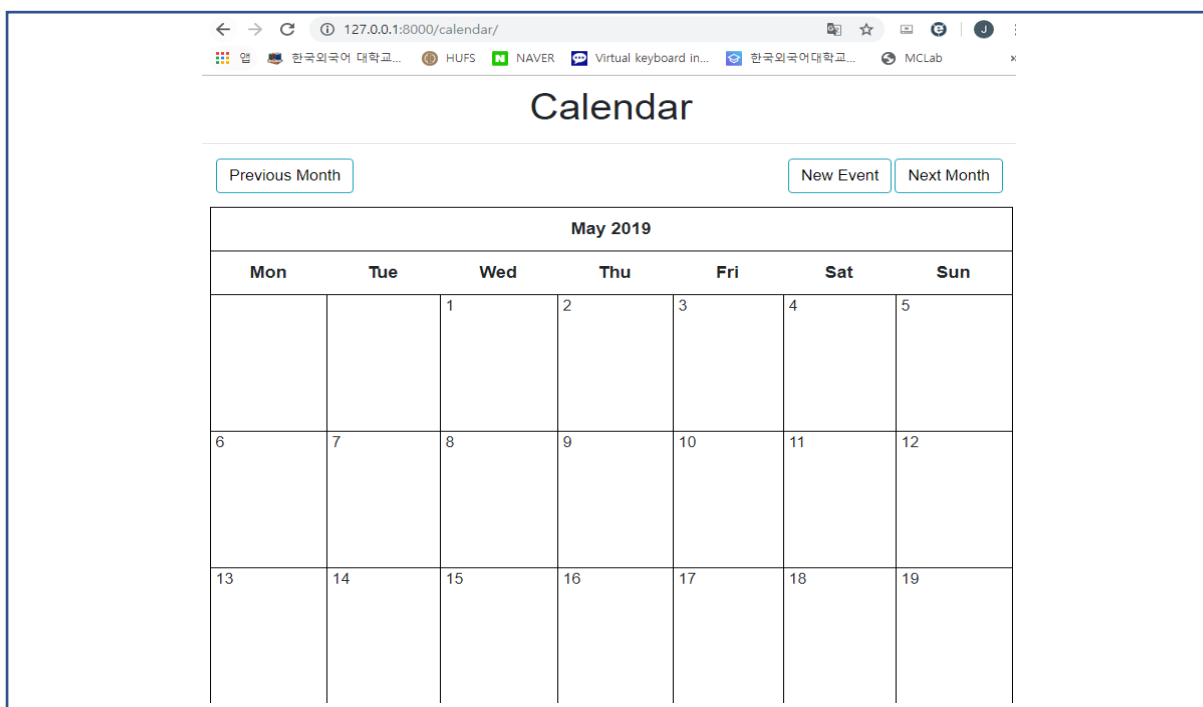
[메인페이지로 돌아가기](#)

8. 캘린더 기능 제공 (SFR-S-8)

1. Calendar 디렉토리



2. Calendar 화면



▶ Utils.py에 Calendar 클래스 정의

```
class Calendar(HTMLCalendar):
    def __init__(self, year=None, month=None):
        self.year = year
        self.month = month
        super(Calendar, self).__init__()
```


상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

► Utils.py에 정의된 calendar 클래스 views.py에서 사용

```
class CalendarView(generic.ListView):
    model = Event
    template_name = 'cal/calendar.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        d = get_date(self.request.GET.get('month', None))
        cal = Calendar(d.year, d.month)
        html_cal = cal.formatmonth(withyear=True)
        context['calendar'] = mark_safe(html_cal)
        context['prev_month'] = prev_month(d)
        context['next_month'] = next_month(d)
        return context
```

9. 캘린더 전, 후 달 이동 기능 제공 (SFR-S-9)

► Previous Month / Next Month

```
def prev_month(d):
    first = d.replace(day=1)
    prev_month = first - timedelta(days=1)
    month = 'month=' + str(prev_month.year) + '-' + str(prev_month.month)
    return month

def next_month(d):
    days_in_month = calendar.monthrange(d.year, d.month)[1]
    last = d.replace(day=days_in_month)
    next_month = last + timedelta(days=1)
    month = 'month=' + str(next_month.year) + '-' + str(next_month.month)
    return month
```

Calendar			
Previous Month			
April 2019			
Mon	Tue	Wed	Thu
1	2	3	4
8	9	10	11
...

년월일:	페이지:
2019-05-20	18

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

10. 캘린더 이벤트 추가 기능 제공 (SFR-S-10)

▶ Model.py에 Event Class 기능 정의

```
class Event(models.Model):
    title = models.CharField(max_length=200)
    description = models.TextField()
    start_time = models.DateTimeField()
    end_time = models.DateTimeField()
```

Event

Title:

오늘의 기록!

Description:

140kg (-0.5kg)
출석!

Start time:

2019-06-04 오전 02:01

End time:

2019-06-04 오전 02:02

Submit

4		5
<ul style="list-style-type: none"> 오늘의 기록! 		

Title:

오늘의 기록!

140kg (-0.5kg)
출석!

년월일: 2019-05-20	페이지: 19
--------------------	------------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

event에는 제목, 내용 시간 등을 적을 수 있다.

우리 어플리케이션은 몸무게, 출석체크 여부와 각자 기록할 수 있는것을 기록한다.

11. 회원 그룹 만들기 기능 제공 (SFR-S-11)

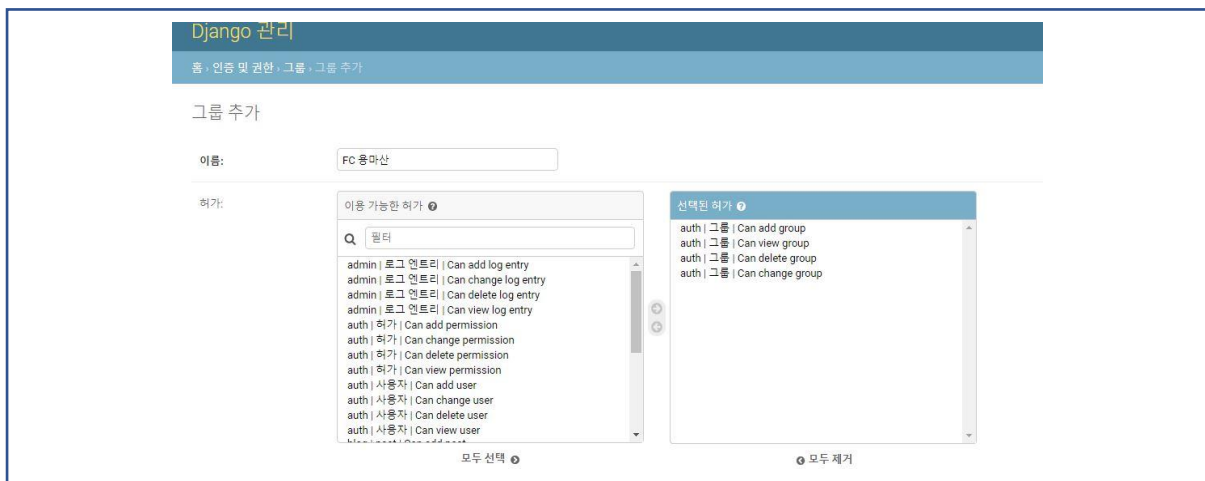
▶ 현재 그룹 생성, 그룹내 회원 추가 admin superuser만 가능.

추후 django.contrib.auth.models.Group 으로 유저가 그룹을 개설할 수 있게 만들 예정.

Django의 admin 창



그룹 추가



년월일:	페이지:
2019-05-20	20

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

그룹 생성 목록

12. 그룹 회원 추가 기능 제공 (SFR-S-12)

사용자 이름

임의로 사용자 추가

년월일:	페이지:
2019-05-20	21

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

13. 운동 모임 개설 기능 제공 (SFR-S-13)

▶ Model.py에 모임 투표 정의

```
class Question(models.Model):
    name = models.CharField(max_length=200)
    attend = models.IntegerField(default=0)
    absence = models.IntegerField(default=0)
    pub_date = models.DateField()

    def __str__(self):
        return self.name
```

▶ 운동 목록 페이지

```
return render(request, 'vote/index.html',{'a': a})
```

개설된 운동 모임 목록

- [운동개설](#) 생성일 : 2019년 5월 13일 [수정](#) [삭제](#)

▶ 운동 개설 페이지

```
return render(request, 'vote/detail.html',{'q':q, 'c_list':c_list})
```

운동개설

- ☐ 출석 [수정삭제](#)
- ☐ 결석 [수정삭제](#)

투표

▶ 운동 결과 페이지

```
return render(request, 'vote/result.html',{'q': q })
```

운동개설 의 투표 결과

1. 출석 - 1 표
2. 결석 - 0 표

[메인으로 돌아가기](#)

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

지금은 운동 모임 개설자가 운동 항목을 직접 입력해야 하지만 앞으로는 데이터베이스에 운동 카테고리 등록할 예정

날씨 정보와 연동해 운동 개설 페이지 상단에 실내, 실외 운동 추천할 예정

14. GPS/날씨예보 api를 활용 의사결정 지원 (SFR-S-14)

운동 개설 시 입력한 위치에 따라 GPS api를 활용한 지도 기능 제공. 또한 위치에 해당하는 날씨예보를 제공하여 사용자가 운동 장소 선정, 운동 종목 선정 시 참고할 수 있도록 지원.

날씨 정보는 강수확률과 강수형태를 제공함.

```
{'POP': 70, 'PTY': 1, 'REH': 75, 'SKY': 4, 'T3H': 22, 'TMX': 23.0, 'UUU': -6.4, 'VEC': 105, 'VVV': 1.8, 'WSD': 6.8}
```

- 강수형태(PTY) 코드 : 없음(0), 비(1), 비/눈(2), 눈(3)

여기서 비/눈은 비와 눈이 섞여 오는 것을 의미 (진눈개비)

15. 칼로리 계산을 통한 운동 평가 기능 (SFR-S-15)

사용자가 운동 실시 후 수행한 운동 종목 및 시간을 입력하면 칼로리 소비량 정보 제공.

이때, 몸무게에 따라 소모 칼로리량이 다름. DB에 입력되어있는 데이터에 따라 계산되어 표시.

운동종목	몸무게(kg)	kcal/30min	시간(30min)	총Kcal
걷기	68.6	136	30	136
	56.5	112	60	224
등산	68.6	286	30	286
	56.5	235	60	470
수영	68.6	321		0
	56.5	265		0
요가	68.6	89	30	89
	56.5	74	30	74
줄넘기	68.6	357		0
	56.5	294		0
자전거	68.6	286	30	286
	56.5	235	60	470
달리기	68.6	250	30	250
	56.5	206	60	412
스쿼트	68.6	250	30	250
	56.5	205	30	205

년월일: 2019-05-20	페이지: 23
--------------------	------------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

4. 데이터베이스 설계

4.1 데이터베이스 필요 소프트웨어

구분	항목	비고
소프트웨어	SQLite3	

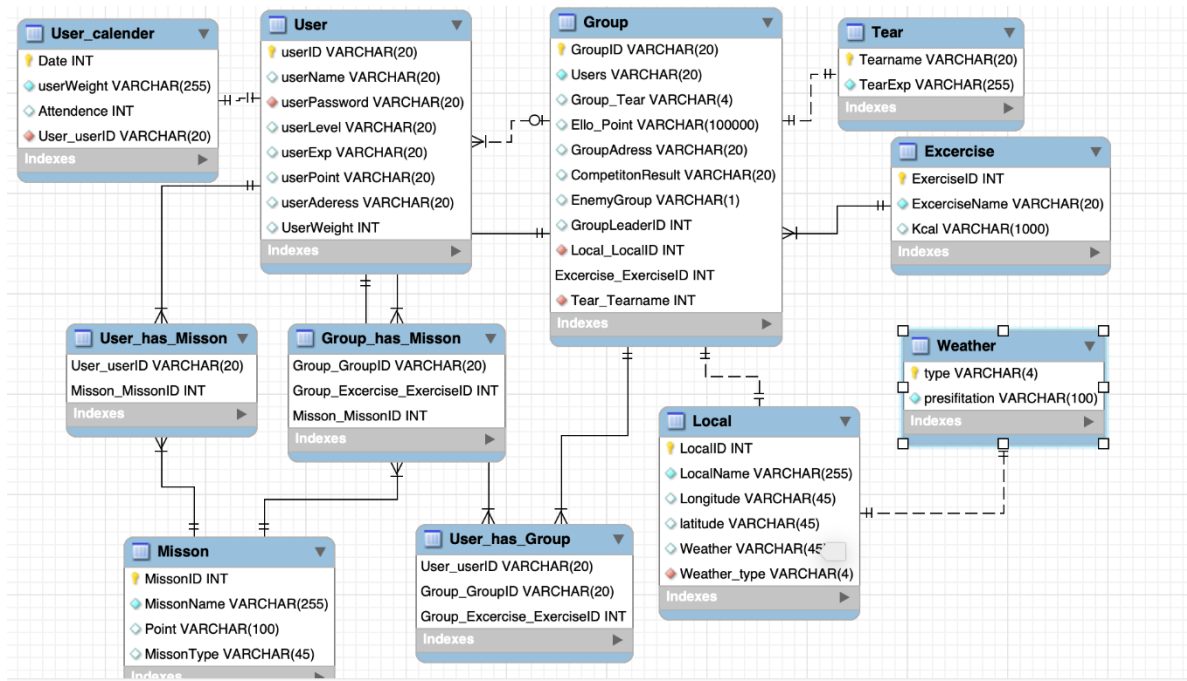
4.2 데이터베이스 상세설계 목록

상세유형	요구사항 명	상세설계 ID	비고
데이터베이스 상세설계 내역	사용자 회원가입 시 이름, ID, PW, 주소를 저장	DB-001	
	사용자 데이터 관리	DB-002	
	그룹 데이터 관리	DB-003	
	운동 데이터 관리	DB-004	
	미션 데이터 관리	DB-005	
	지역 데이터 관리	DB-006	
	날씨 데이터 관리	DB-007	
	캘린더 데이터 관리	DB-008	
	로그인 시도 시 정보의 일치 여부를 확인	DB-009	

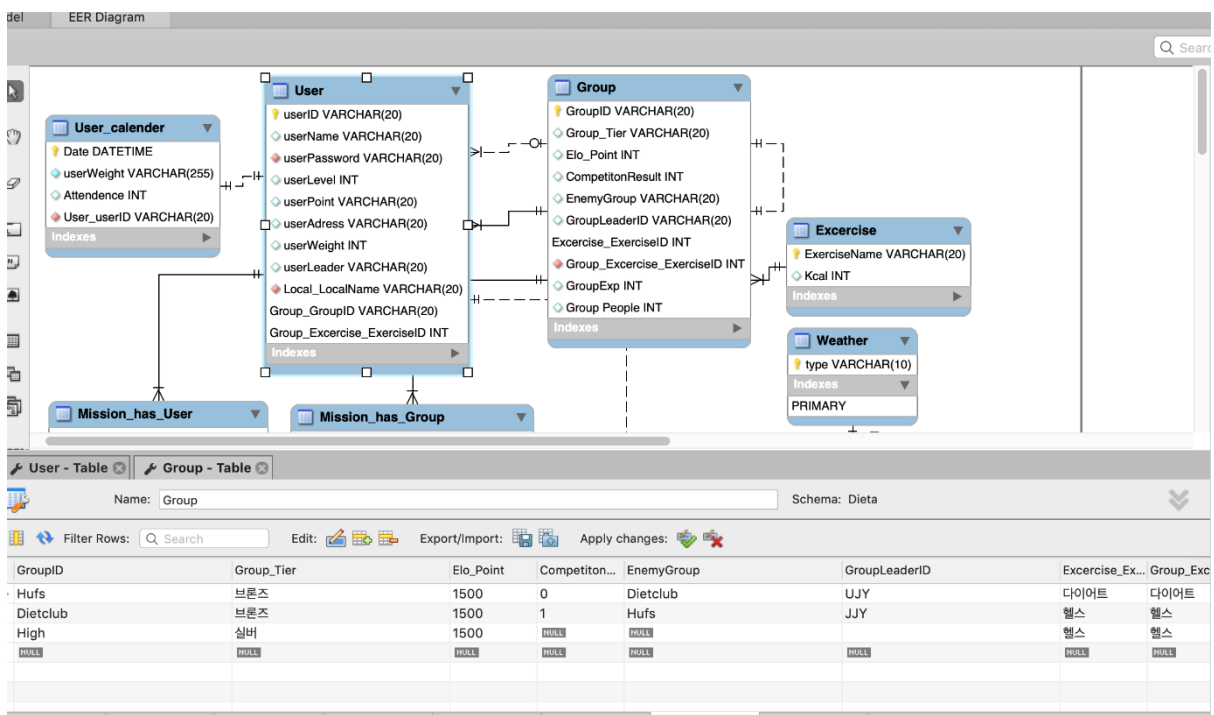
상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

4.3 데이터베이스 모델링

(가) 데이터베이스 초기 디자인

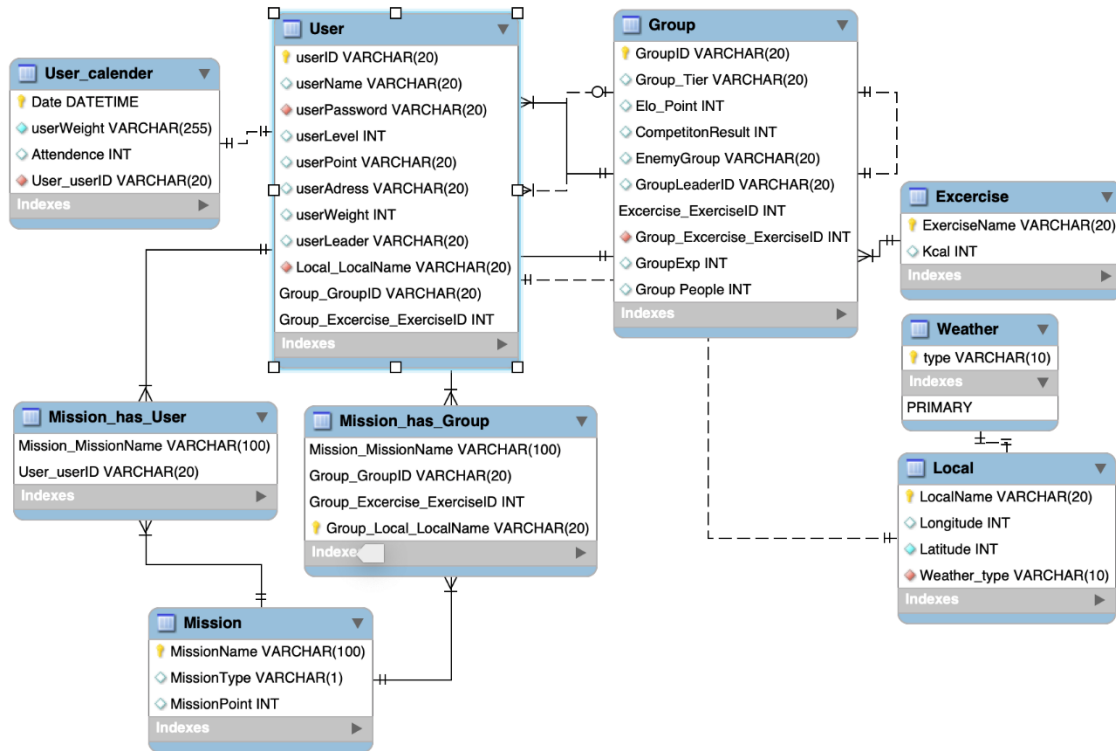


(나) 정규화



상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

(다) 데이터베이스 수정 디자인(수정가능성있음)



년월일: 2019-05-20	페이지: 26
--------------------	------------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

4) 데이터베이스 항목별 상세설계

1. 사용자 회원가입 시 이름, ID, PW, 주소를 저장 (DB-001)

서버에서 요청하는 사용자의 ID 존재 여부 확인 후 같은 ID가 없을 시에 사용자는 회원 가입을 할 수 있다.

- ▶ server에서 받은 데이터와 DB의 USER 테이블의 ID를 비교한다.
- ▶ DB USER 테이블에 동일한 ID가 있을 경우 존재한다는 메시지를 server에 전송한다.
- ▶ DB USER 테이블에 동일한 ID가 없을 경우 존재하지 않는다는 메시지를 server에 전송한다.
- ▶ 사용자가 회원가입을 완료 했을 경우 USER 테이블에 새로운 tuple을 생성하여 데이터를 저장한다.

```
SQL > INSERT INTO User VALUES ("user_ID", "userName", "userPassword", "userAddress"...)

```

2. 사용자 데이터 관리 (DB-002)

데이터베이스에서는 사용자의 data 정보를 관리한다.

- ▶ 사용자의 data 정보 관리를 위해 User 테이블을 만든다.
- ▶ 테이블의 COLUMN은 사용자 ID, PW, 이름, 주소, 레벨, 경험치, 그룹, 몸무게, 리더 여부 등이 들어간다.
- ▶ 리더 여부를 알려주는 isLeader의 경우 INT값을 받아 user가 그룹 리더이면 1로, 그렇지 않으면 0으로 처리한다.

SQL >

```
CREATE TABLE User

```

```
(
```

년월일: 2019-05-20	페이지: 27
--------------------	------------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

```

userID VARCHAR(20) NOT NULL,

userPassword VARCHAR(20) NOT NULL,

userName VARCHAR(20) NOT NULL,

userAddress VARCHAR(20) NOT NULL,

userLevel INT NOT NULL,

userExp INT NOT NULL,

GroupID VARCHAR(20),

isLeader INT NOT NULL,

userWeight DOUBLE,

PRIMARY KEY (userID),

FOREIGN KEY (GroupID) REFERENCES Group(GroupID),

FOREIGN KEY (userAddress) REFERENCES Local(LocalName)

);

```

3. 그룹 데이터 관리 (DB-003)

데이터베이스에서는 그룹의 data 정보를 관리한다.

- ▶ 그룹 data 정보 관리를 위해 Group 테이블을 만든다.
- ▶ 테이블의 COLUMN은 그룹 ID, 그룹 티어, 엘로 포인트, 그룹 주소, 경쟁 결과, 경쟁 그룹, 그룹리더 아이디, 운동, 그룹 경험치 등이 들어간다.
- ▶ 경쟁 결과를 받는 CompetitionResult의 경우 INT 자료형으로 승리 시 1, 패배 시 0으로 처리한다.

SQL >

```
CREATE TABLE Group
```

년월일: 2019-05-20	페이지: 28
--------------------	------------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

```
(
  GroupID VARCHAR(20) NOT NULL,
  GroupTier VARCHAR(20),
  eloPoint INT NOT NULL,
  GroupAddress VARCHAR(20) NOT NULL,
  CompetitionResult INT,
  EnemyGroup VARCHAR(20),
  GroupLeaderID VARCHAR(20) NOT NULL,
  GroupExercise VARCHAR(20),
  GroupExp INT NOT NULL,
  PRIMARY KEY (GroupID),
  FOREIGN KEY (GroupAddress) REFERENCES Local(LocalName),
  FOREIGN KEY (EnemyGroup) REFERENCES Group(GroupID),
  FOREIGN KEY (GroupExercise) REFERENCES Exercise(ExerciseName)
);
```

4. 운동 데이터 관리 (DB-004)

데이터베이스에서는 운동 data 정보를 관리한다.

- ▶ 운동 data 정보 관리를 위해 Exercise 테이블을 만든다.
- ▶ 테이블의 COLUMN은 운동 이름, 소모 칼로리 등이 들어간다.

SQL >

```
CREATE TABLE Exercise
```

```
(
  ExerciseName VARCHAR(20) NOT NULL,
```

년월일: 2019-05-20	페이지: 29
--------------------	------------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

Kcal INT,

PRIMARY KEY (ExerciseName)

);

5. 미션 데이터 관리 (DB-004)

데이터베이스에서는 미션 data 정보를 관리한다.

- ▶ 미션 data 정보 관리를 위해 Mission 테이블을 만든다.
- ▶ 테이블의 COLUMN은 미션 이름, 미션 타입(그룹/개인), 미션 점수 등이 들어간다.
- ▶ MissionType은 그룹 미션이면 G로, 개인 미션이면 P로 처리한다.

SQL >

CREATE TABLE Mission

(

MissionName VARCHAR(20) NOT NULL,

MissionType VARCHAR(1) NOT NULL,

MissionPoint INT,

PRIMARY KEY (MissionName)

);

6. 지역 데이터 관리 (DB-006)

데이터베이스에서는 지역 data 정보를 관리한다.

- ▶ 지역 data 정보 관리를 위해 Local 테이블을 만든다.
- ▶ 테이블의 COLUMN은 지역 이름, 위도, 경도, 날씨 등이 들어간다.

년월일: 2019-05-20	페이지: 30
--------------------	------------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

SQL >

CREATE TABLE Local

(

LocalName VARCHAR(20) NOT NULL,

Longitude INT NOT NULL,

Latitude INT NOT NULL,

WeatherType VARCHAR(10),

PRIMARY KEY (LocalName)

);

7. 날씨 데이터 관리 (DB-007)

데이터베이스에서는 날씨 data 정보를 관리한다.

▶ 날씨 data 정보 관리를 위해 Weather 테이블을 만든다.

▶ 테이블의 COLUMN은 등이 들어간다.

SQL >

CREATE TABLE USER

(

VARCHAR(20) NOT NULL,

PRIMARY KEY ()

FOREIGN KEY () REFERENCES ()

);

년월일: 2019-05-20	페이지: 31
--------------------	------------

상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

8. 캘린더 데이터 관리 (DB-008)

데이터베이스에서는 캘린더의 data 정보를 관리한다.

- ▶ 캘린더의 data 정보 관리를 위해 User_calender 테이블을 만든다.
- ▶ 테이블의 COLUMN은 Date, userWeight, Attendance, userID 등이 들어간다.
- ▶ Attendance의 경우 int 값을 받아 0이면 결석, 1이면 출석으로 처리한다.

SQL >

```
CREATE TABLE UserCalender
```

```
(
```

```
    Date DATETIME NOT NULL,
```

```
    userWeight VARCHAR INT,
```

```
    Attendance INT,
```

```
    userID,
```

```
    PRIMARY KEY (Date),
```

```
    FOREIGN KEY (userID) REFERENCES User(userID)
```

```
);
```

9. 로그인 시도 시 정보의 일치 여부를 확인 (DB-009)

server에서 요청하는 사용자의 ID와 PW 데이터를 받아 일치 여부를 판단한다.

- ▶ 사용자가 Interface Application에서 로그인을 시도하여 server로 request를 전송한다.
- ▶ DB의 User 테이블에서 ID를 검색한다.
- 1 ID가 있을 경우: 해당 ID의 패스워드를 검색하여 일치할 경우 성공 메시지를 server에 보낸다.

server에서 요청하는 사용자의 ID와 PW 데이터를 받아 일치 여부를 판단한다.

- ▶ 사용자가 Interface Application에서 로그인을 시도하여 server로 request를 전송한다.
- ▶ DB의 User 테이블에서 ID를 검색한다.

년월일: 2019-05-20	페이지: 32
--------------------	------------



상세설계서: 그룹 운동 커뮤니티 어플리케이션(DIETA)

1 ID가 있을 경우: 해당 ID의 패스워드를 검색하여 일치할 경우 성공 메시지를 server에 보낸다.

2 ID가 없을 경우: 에러 메시지를 Interface Application에 비동기 방식으로 보낸다.

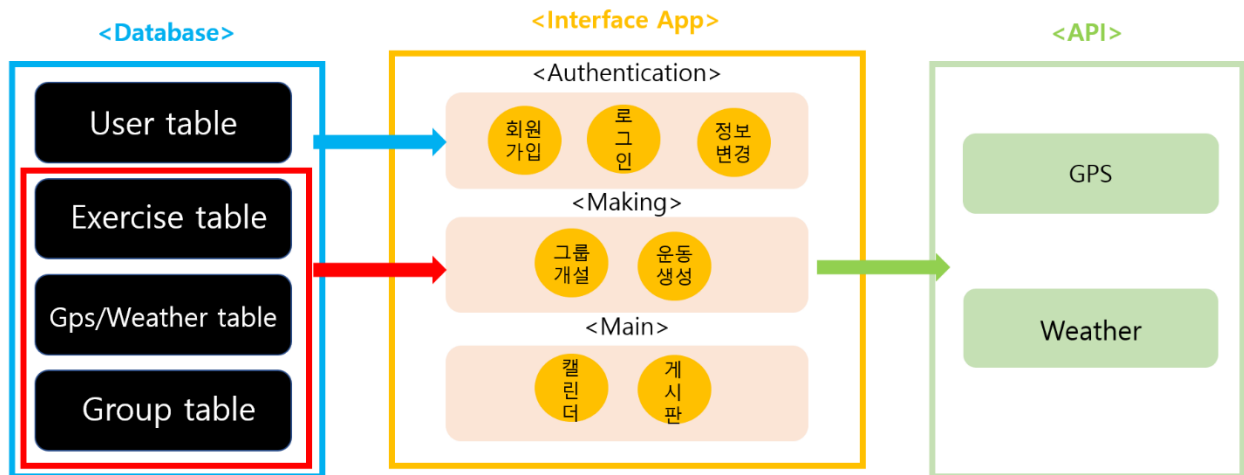
SELECT userID, userPassword

FROM User

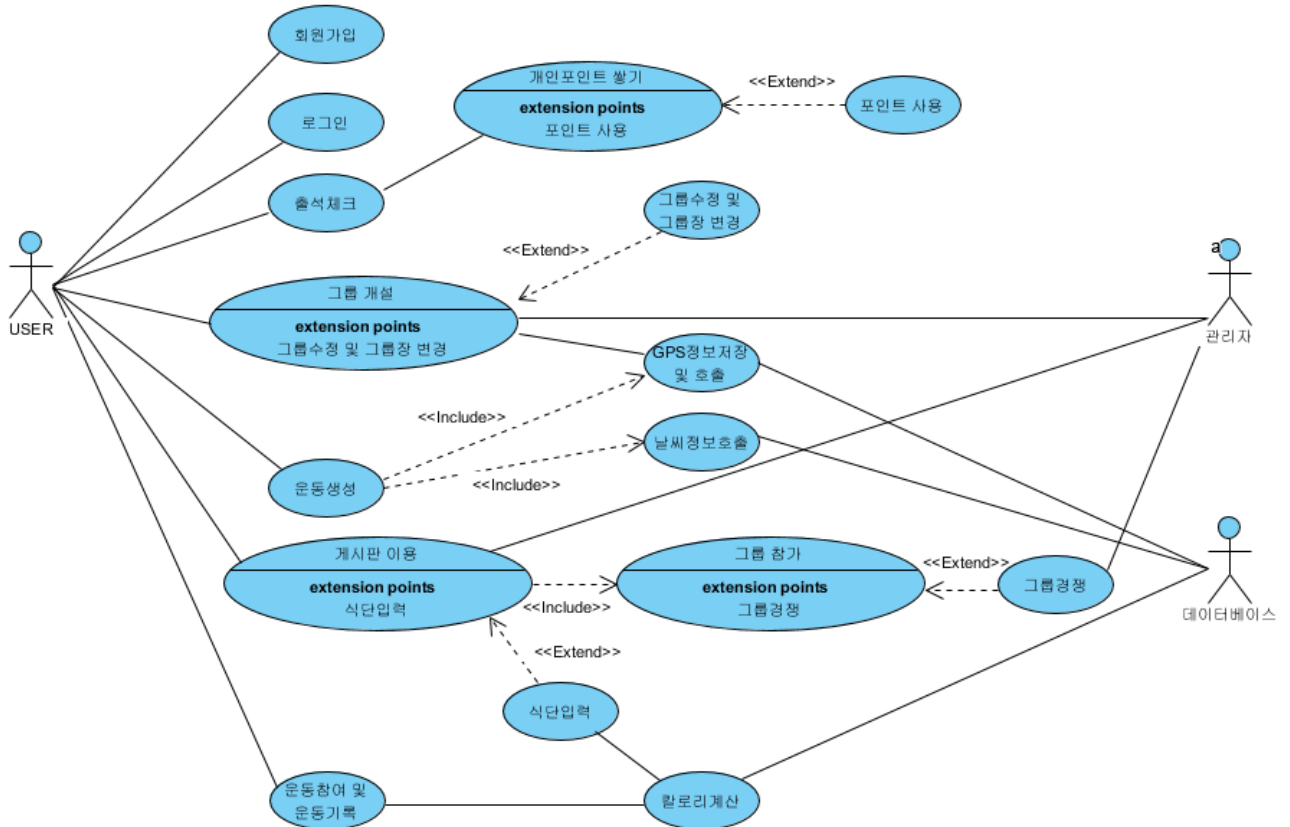
WHERE userID = 'input data' and userPassword = 'input data';

5. 상세설계 산출물

5.1 시스템 종합 설계도



5.2 Use case diagram



- 사용자가 www.dieta.com으로 Web을 통해 접속하면 Dieta가 지원하는 기능들을 사용할 수 있다.

- 회원가입 후 로그인을 하면 출석체크를 하게 된다. 이때 개인 포인트가 적립되며 추후에 제휴된 제품을 구매할 때 포인트를 쓸 수 있다.

- User가 그룹 생성을 요청하면 관리자가 그룹을 생성한다.

(**추후 django.contrib.auth.models.Group를 활용해 User가 그룹을 개설할 수 있게 만들고 주소 입력 시 GPS주소를 저장할 예정)

- User가 운동 목록을 만들면 회원들이 참여 투표를 한다. 그리고 참여 현황을 알 수 있게 된다.

(**추후 날씨 정보와 연동해 사용자가 카테고리에 지정된 운동을 선택 시 도움을 줄 예정)

- 사용자는 게시판을 들어가서 커뮤니티 활동을 할 수 있다.

- 사용자는 별도의 개설된 링크를 통해 운동에 참여할 수 있고, 운동 기록 시 소모된 칼로리가 계산된다.

6. 담당업무

