【朝华Blossom】

主要功能设计

- 目标：设计一个同学录网站
  - 主要作用：记录高中生活，毕业后留念回忆
  - 使用对象：学生、老师，部分功能不对老师开放
- 功能：自由发挥
- 需要帮助：帮忙想有什么功能，大类即可（即加粗标题），细节也可完善

ps：需要给功能想一个文学性的名字（一切自由发挥）

需要翻译！

1. 区分老师和学生的字段
2. 班级职位：`role`

【（暂代）基本信息】【英文：】同学老师基本信息

1. 可以加一个特殊网页，方便使用者对同学们进行特殊备注，关键词记录

【与君酌Toast】相互之间的看法

1. 同学和老师（同属一个基类）可以对相互发表公开看法
2. 同样可以发表私密的文章（哈哈表白专用，放心我会做加密系统的）
3. 可以发表就是那种只能给别人看但是评价对象看不到的那种
4. **想法** 设置屏蔽人

【拾花录】【英文：】活动记录

> 1. 项目名称：
>   - 生命书 Lifebook
>   - 回忆灯 Memorylight
>   - 惜别路 Rememberlane
>   - 时印 Timeprints
>   - 故事城 Talescape
>   - 约言 Promisebook
>   - 未来志 Futurelog
>   - 青春记 Youthmemo
>   - 朋友词 Friendverse
>   - 同窗程 Classjourney
> 2. 功能名称及英文翻译：
>   - 基本信息
>   Essentials
>   Identity
>   Profile
>   Details
>   Introduction
>   Facts

> - Traits
> - Features
> - Characteristics
> - 与君酌
>   - Impressions
>   - Opinions
>   - Musings
>   - Comments
>   - Reflections
>   - Feedback
>   - Exchanges
>   - Perceptions
>   - Conversations
>   - Impress
> - 拾花录
>   - Harvest
>   - Journal
>   - Memories
>   - Chronicles
>   - Diary
>   - Reminiscence
>   - Record
>   - Log
>   - Memoir
>   - Garden

建议：

1. 加一个匿名系统【待细说】

要点

1. 一人一账号一身份，非本校学生不得访问
2. 一个人可以对应多个班级

主要问题及解决方案

与主站共用一套用户系统的问题

用户数据如何互通

1. 子站后台均调用主站`account`应用，保证`django`模型结构相同
2. 应用内包括的模型使用单独的数据库，保证数据共享
3. 关于用户资料及设置等的修改进入单独子站`account`，保证修改界面相同，且只有一份，方便修改升级
4. 对于子站（包括主站）登录及注册操作均跳转到`account`进行

存在仍需解决的技术问题

由于域名之间的`localStorage`不互通，没办法让`authToken`在域名间流通，即登录状态无法同步，这个问题同样存在于`app`端

目前查到打算使用的处理办法：SSO

> **通过页面重定向的方式**
> 最后一种介绍的方式，是通过父应用和子应用来回重定向中进行通信，实现信息的安全传递。 父应用提供一个GET方式的登录接口，用户通过子应用重定向连接的方式访问这个接口，如果用户还没有登录，则返回一个的登录页面，用户输入账号密码进行登录。如果用户已经登录了，则生成加密的Token，并且重定向到子应用提供的验证Token的接口，通过解密和校验之后，子应用登录当前用户。
> ————————————————————
> 版权声明：本文为CSDN博主「一只橙子0」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。
> 原文链接：https://blog.csdn.net/qq_41595452/article/details/122066984

`sso`服务暂定挂载在`account.sakuyark.com/login`

拟跳转方案[1]

在本跳转方案下仍存在问题

1. 重定向必须从子域跳到`sso`服务再回到子域，过程繁琐
2. 在什么时候检查局部会话：用户在子域点击登录时，一般只会出现子域跳转到用户管理系统的情况，但是用户管理系统是`sso`终端，及全局会话

5.

主站基本服务内容

由于主站搭建了一些基本服务措施，无法在各项目内互通

1. 用户系统（上有方案，此处尝试统一解决）
2. 应用发布与更新套装
3. 公告系统
4. 图片处理

> 用于解决多图片上传问题
> 解决思路：用一个模型单独保存图片，用多对多连接，可以设置保存时长

5. 个推系统

可能可行的解决方案

1. 搭建本地web服务

改变计划，连网站一起拆掉重做（前端不用动）

模型构建与关联

Model Person（暂定）

字段设计

- `name` 姓名：`CharField`
- `gender` 性别：`CharField`，`TextChoices(男，女)`
- `class_（暂定）` 班级：`M2M(Class)`，定义
- `membership` 中间表：`O2M(ClassMembership)`，接收
- `role` 身份：`CharField`，`TextChoices(老师，学生)`
- `phone` 手机号：`PhoneNumberField`，一人对应一个手机号，暂定
  - 未使用过改字段，需要测试
- `email` 邮箱：`EmailField`，一人对应一个邮箱号
- `?` 物种：`CharField`

定义

```python
class GenderChoices(models.TextChoices):
    male = "MALE", "男"
    female = "FEMALE", "女"


class RoleChoices(models.TextChoices):
    student = "STUDENT", "学生"
    teacher = "TEACHER", "老师"


class Person(models.Model):
    name = models.CharField("姓名", max_length=256)
    gender = models.CharField("性别", choices=GenderChoices.choices)
    class_ = models.ManyToManyField("Class", related_name="person")
    role = models.CharField("角色", choices=RoleChoices.choices)
    phone = PhoneNumberField(verbose_name="手机号码", region=86)
    email = models.EmailField(verbose_name="邮箱")
```

Model Class（暂定）

字段设计

- `id` 班级编号：`CharField(primary=True)`，采用格式`{建班年份}{班级}`，`{班级}`的描述待定
- `created` 建班年份：`PositiveIntegerField`
- `name` 班级名称：`CharField`
- `type_` 班级类型：`CharField`，`TextChoice(行政班级，走班班级)`
- `student（暂定）` 学生：`M2M(Person)`，接收
- `teacher（暂定）` 老师：`M2M(Person)`，接收
- `membership` 中间表：`O2M(ClassMembership)`，接收

定义

```
class ClassTypeChoices(models.TextChoices):
    administrative = "ADMINISTRATIVE", "行政班级"
    walking = "WALKING", "走班班级"


class Class(models.Model):
    id = models.CharField("班级编号", primary_key=True, max_length="256")
    created = models.PositiveIntegerField("建班年份")
    name = models.CharField("班级名称", max_length="256")
    type_ = models.CharField("班级类型", choices=ClassTypeChoices.choices)
```

Model ClassMembership

字段设计

- `id` 编号：`UUID(uuid4)`
- `class_` 班级：`ForeignKey(related_name="membership")`
- `person` 人：`ForeignKey(related_name="membership")`
- `position` 职位：`CharField`
- `number` 学号：`PositiveIntegerField(null=True)`

定义

```
class ClassMembership(models.Model):
    """
    Class-Person中间表
    """

    class Meta:
        constraints = [
            models.UniqueConstraint(fields=['class_', 'person'], name='class_membership')
        ]

    id = models.UUIDField(primary_key=True, default=uuid.uuid4, editable=False, max_length=64)
    class_ = models.ForeignKey("Class", related_name="membership", on_delete=models.CASCADE)
    person = models.ForeignKey("Person", related_name="membership", on_delete=models.CASCADE)
    position = models.CharField("职位", max_length=1024)
    number = models.PositiveIntegerField("学号", null=True)
```

1. ↵