

# 益在中大一软件设计文档

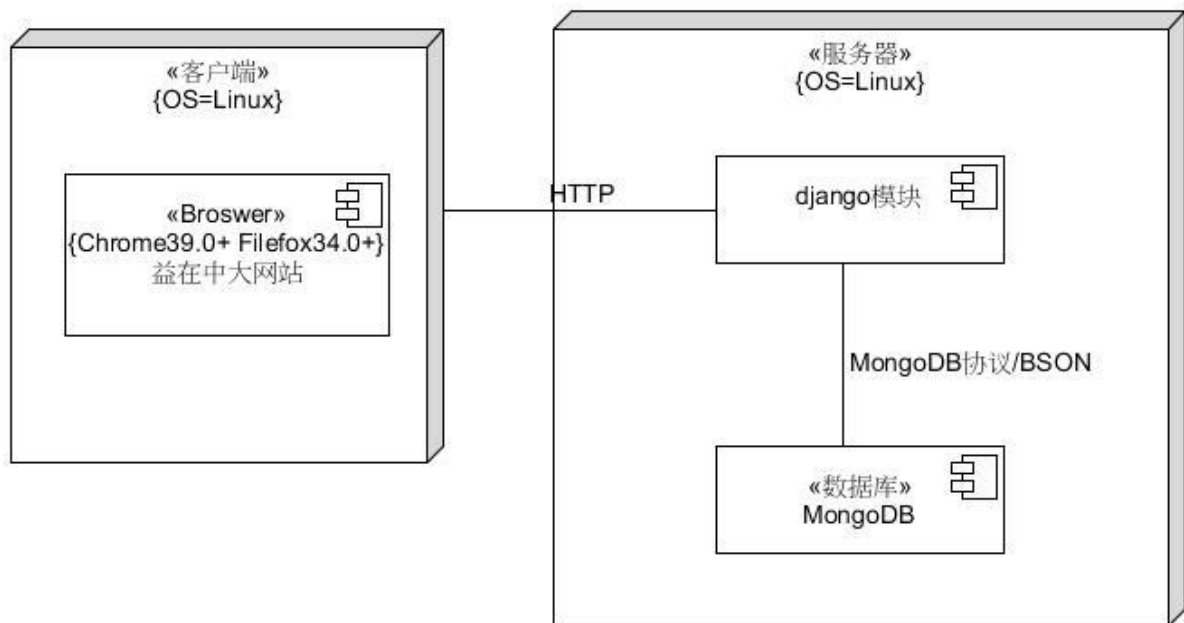
## 1. 技术选型

项目	Web app
1 终端支持	PC、Pad、Phone
1.1 开发语言框架	HTML5、CSS3、Javascript
1.2 响应式布局框架	Bootstrap
2 服务端支持	
2.1 语言	Python
2.2 web 框架	Django
2.3 数据库	Mongodb
2.4 ORM 框架	Mongoengine
3 开发平台与工具	
3.1 IDE	pyCharm
3.3 源代码管理	Github

- 后台选用 **django** 框架，它使用了 **MTV** 设计模式——由 **MVC** 模式衍生而来，**Model**，**Template**，以及 **View** 三者分别明确的定义了数据模型、视图、以及控制器三类代码，在实际编程中的体现就是这三部分代码的分离，有利于我们对项目业务逻辑和项目实际进度的掌控。
- 数据库使用 **MongoDB**，并使用 **Mongoengine** 对象关系存储框架简化对数据模型的操作
- 把应用做成 **web** 应用可以跨平台，前端选用 **bootstrap** 框架，它是一种响应式的布局前端框架，能够让我们的 **web** 应用在不同的手机或 **pad** 上都有较好的展现。

## 2. 架构设计

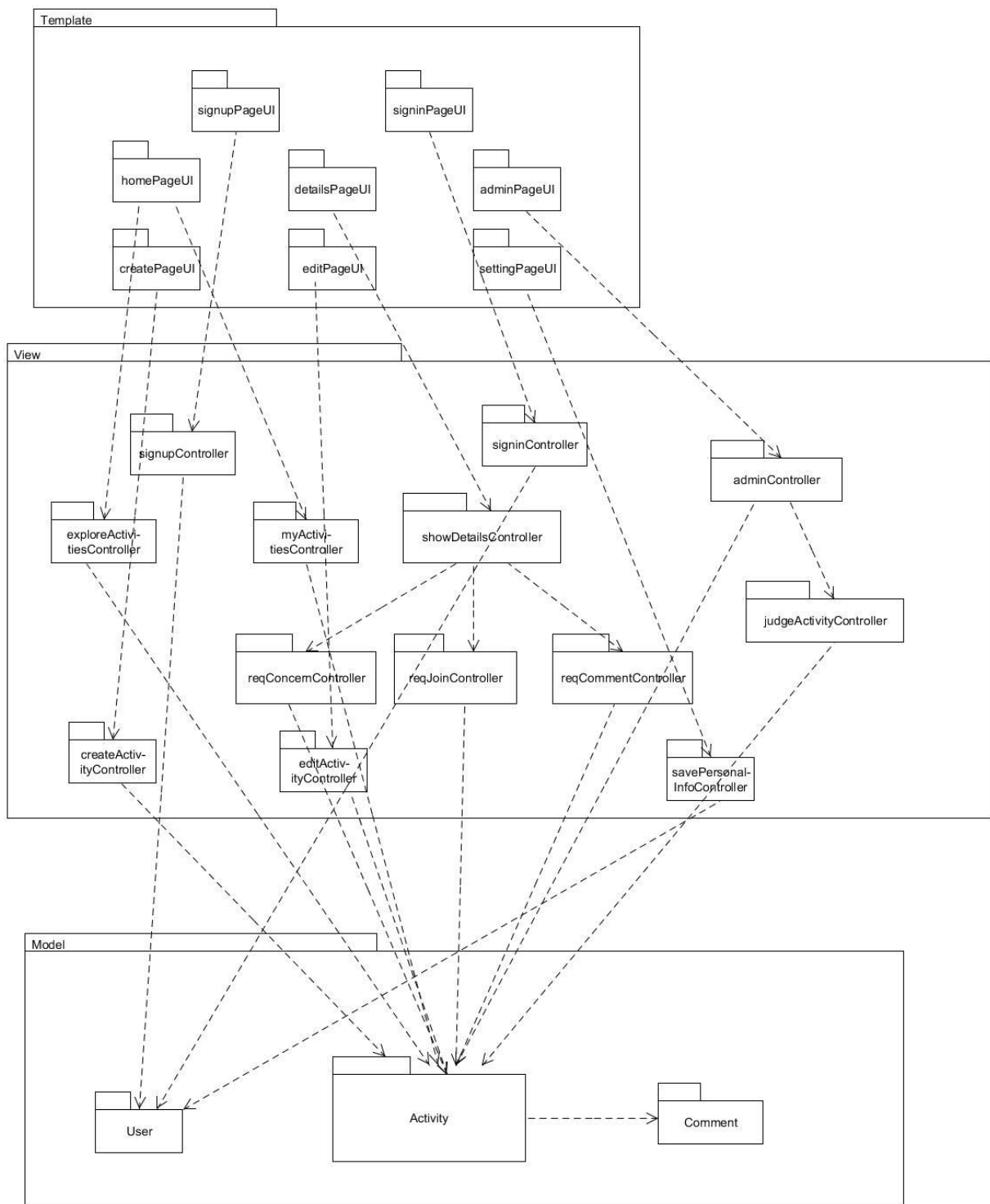
和一般的网站一样，益在中大是 **B/S** 类的应用，它的部署架构如下：



客户端是用户的浏览器，用户通过浏览器访问益在中大网站，发送一系列请求给服务器上 django 模块，django 模块能够访问 MongoDB 数据库，三者分别是表现层，业务逻辑层，和数据持久层。当然 django 模块中包含许多组件，在第三部分——模块划分中会进行详细的说明。

### 3. 模块划分

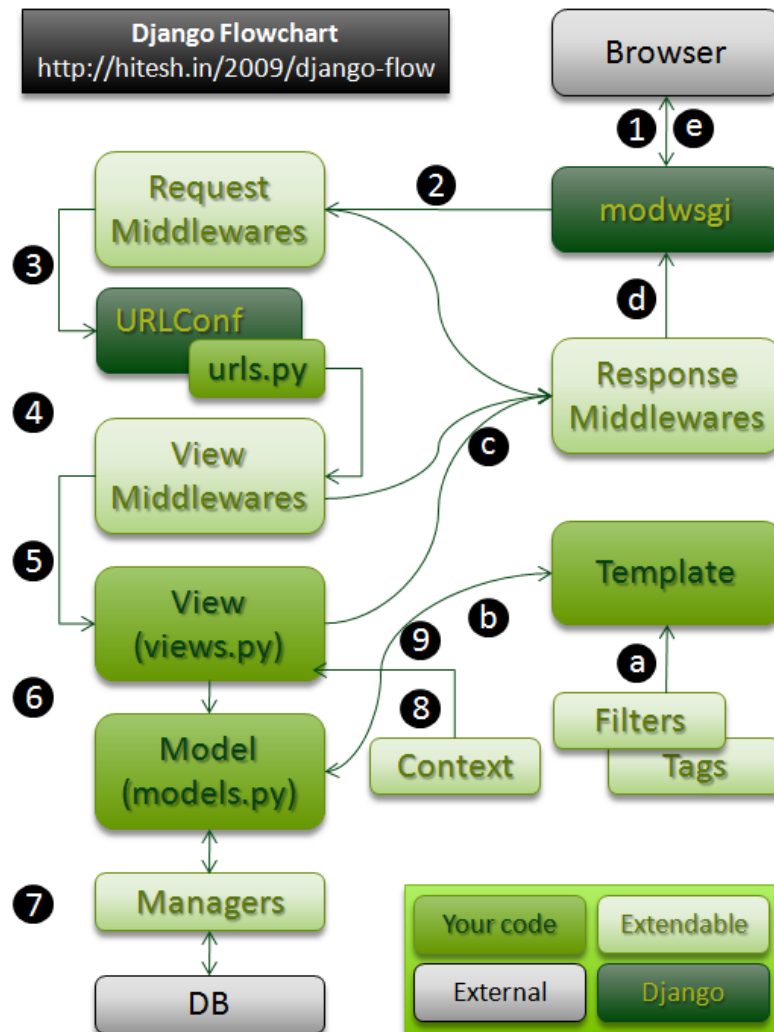
按照 MVT 模式我们将系统分为三大模块，如下：



## 4. 使用的软件设计技术

### 4.1. 使用 MVT 设计模式——Django 设计模式

下面这幅图片很好地展示了 Django 处理请求并作出响应的过程：



详细流程如下：

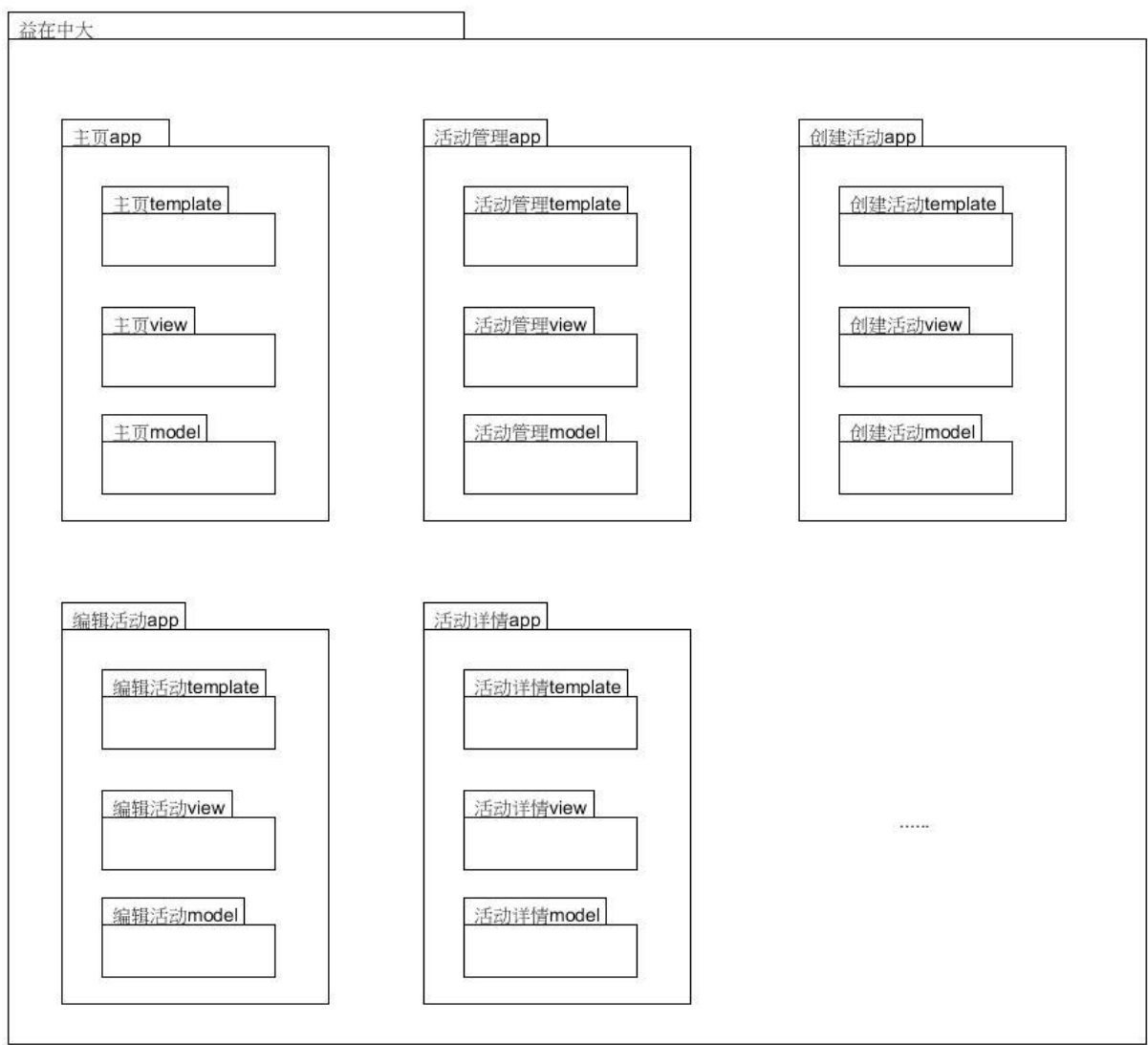
1. 用户通过浏览器请求一个页面
2. 请求到达 Request Middlewares，中间件对 request 做一些预处理或者直接 response 请求
3. URLConf 通过 urls.py 文件和请求的 URL 找到相应的 View
4. View Middlewares 被访问，它同样可以对 request 做一些处理或者直接返回 response
5. 调用 View 中的函数
6. View 中的方法可以选择性的通过 Models 访问底层的数据
7. 所有的 Model-to-DB 的交互都是通过 manager 完成的
8. 如果需要，Views 可以使用一个特殊的 Context
9. Context 被传给 Template 用来生成页面
- a. Template 使用 Filters 和 Tags 去渲染输出
- b. 输出被返回到 View
- c. HttpResponse 被发送到 Response Middlewares
- d. 任何 Response Middlewares 都可以丰富 response 或者返回一个完全不同的 response

e. Response 返回到浏览器，呈现给用户

基于 Django 框架，我们需要编程实现的主要是 url 配置文件、template、model 以及充当 controller 的 view 类文件。

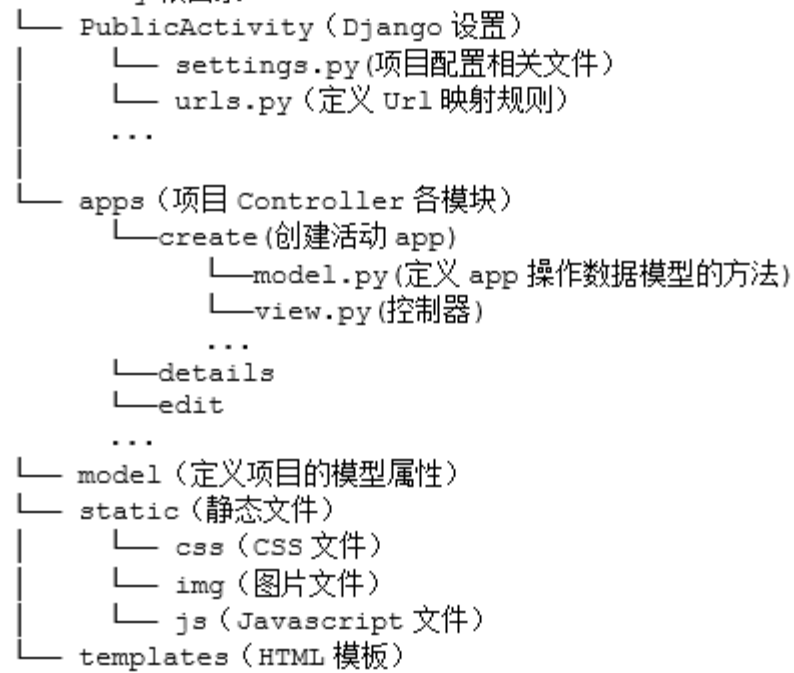
我们的系统的基础就是 MVT 三个模块，每个模块中都有与特定的 app 相对应的部分，在项目目录结构上以 app 为基础进行。App 是我们网站中完成某项业务的基本单元，比如用户注册登录注销 app、添加活动 app、查看活动详细信息 app 等。

当我们想要向系统中添加新的用例时可以直接创建一个新的 app，系统会自动为该 app 生成 MVT 模式需要的三类文件，我们只需要分别修改它们，并在 url 配置文件中增加相应的关联即可完成新功能的添加。



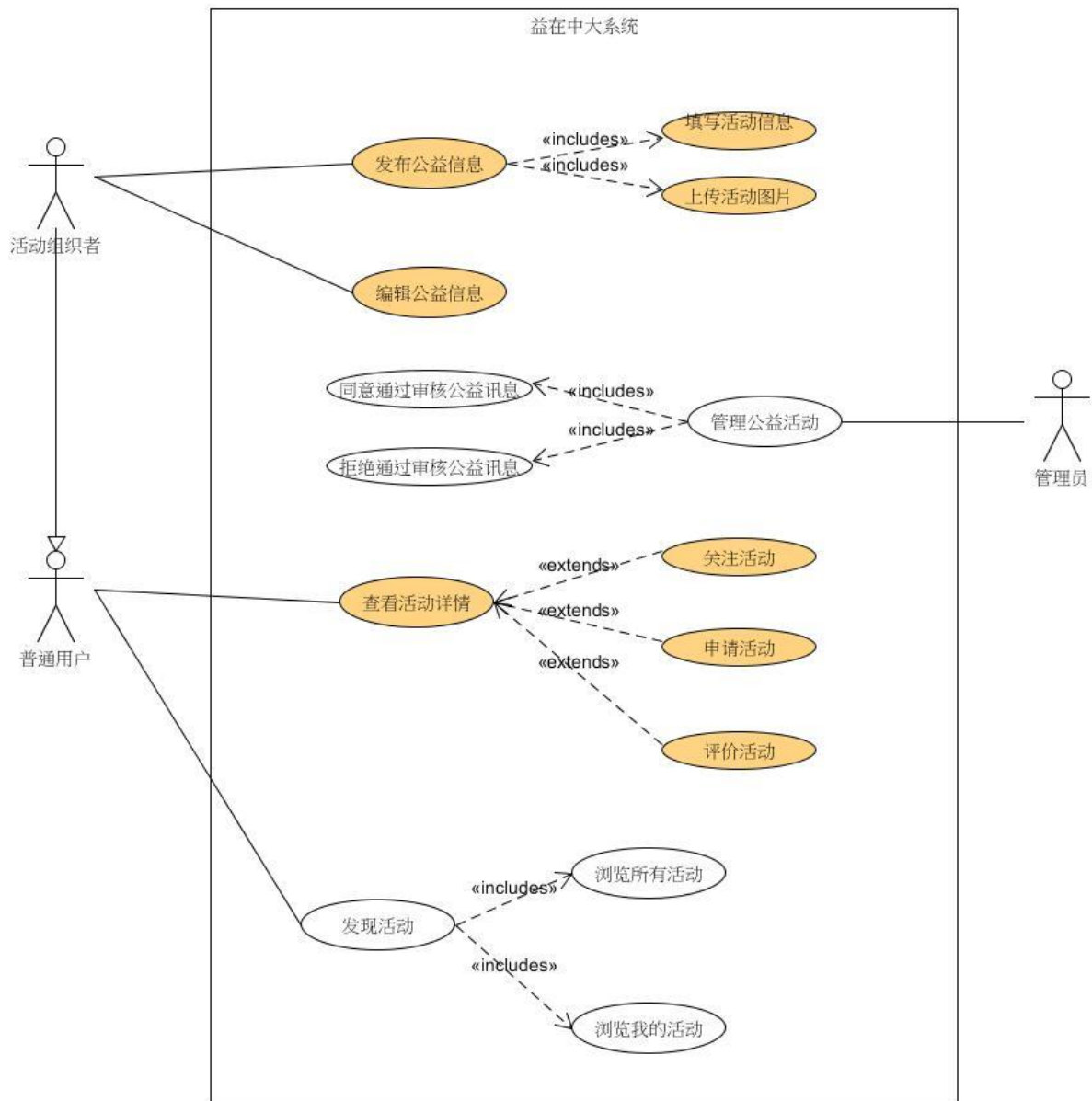
以上模块的划分从项目的目录结构上可以反应一二：

## PublicActivity 根目录

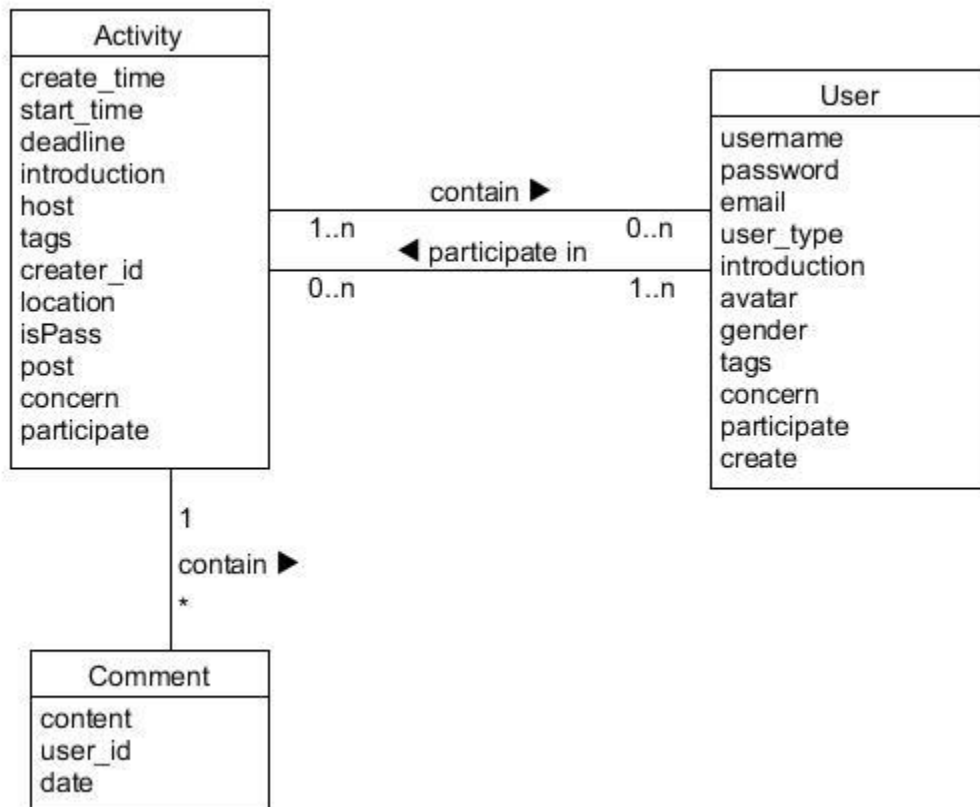


## 5. 关键模型

### 5.1. 用例图



## 5.2. 领域模型



## 5.3. 架构图

