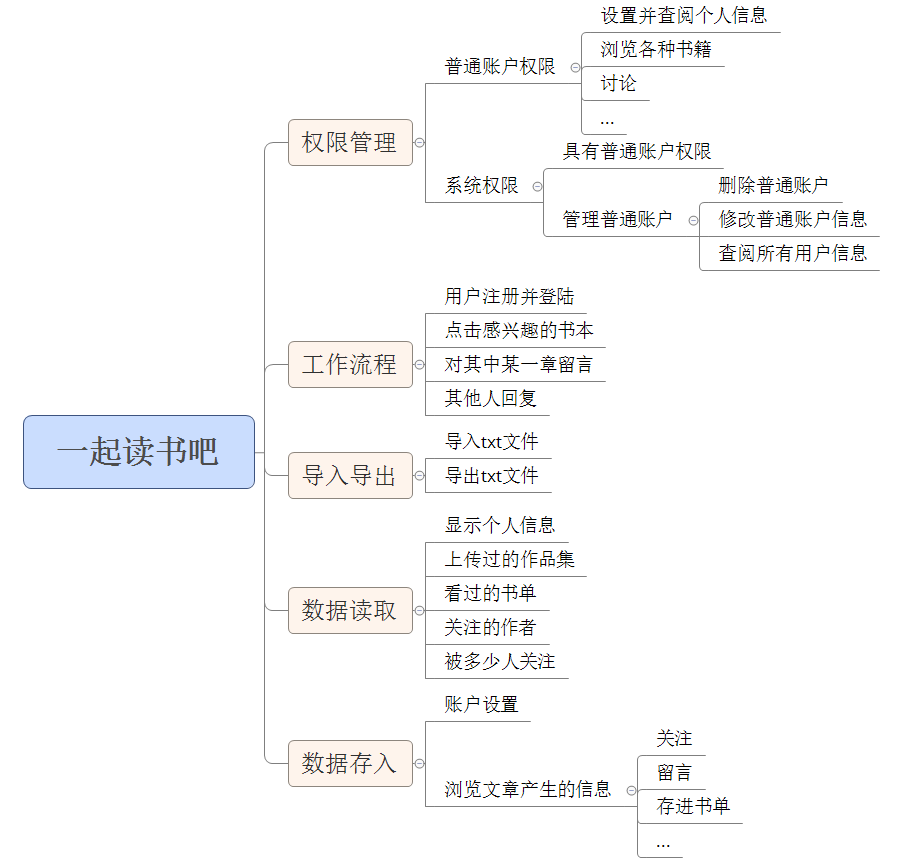
## 项目需求分析

**系统名称：**一起读书吧

**数据录入：**通过用户注册收集用户数据，至于书本的数据暂时不用真实内容。或许可以尝试解析pdf或者txt文件**。**

* **系统功能模块：**

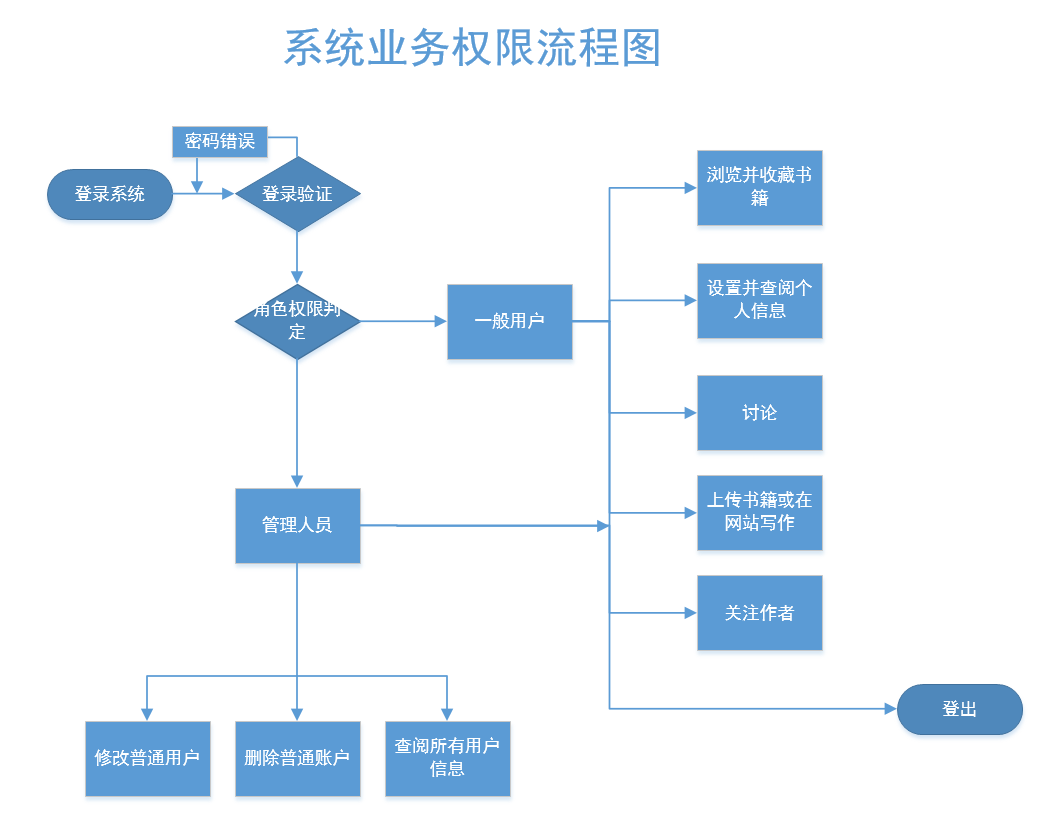
系统由五个大的功能模块组成，每个大功能模块由若干个小的功能模块组成，如上图所示，能基本满足一般用户的需求。

* **用户与权限管理：**

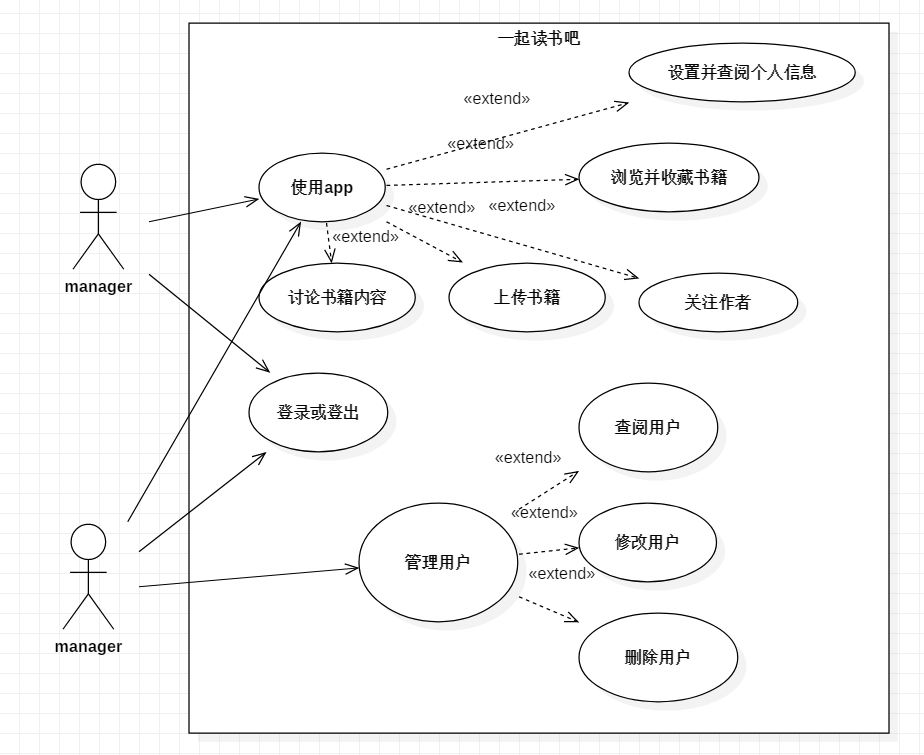
**用户角色：**一般用户，管理用户

**权限级别：**管理用户比一般用户多了几个功能：删除，修改，查阅用户信息

**数据可视性：**普通用户只能看见自己的信息，而管理用户可以看见所有用户信息。



* **用例设计：**



* **模块设计：**

**后台：**

Entity模块：使用ORM（Hibernate/MyBatis）时需要建立和数据库的联系。

Dao模块：主要是做数据持久层的工作，主要与数据库进行交互，DAO层首先会创建DAO接口，

DaoImpl模块：实现Dao接口的各个工作。

Controller模块：直接和前端交互的接口，接收和发送数据

Service模块：实际上是接口模块，定义每一个服务

ServiceImpl模块: 实现Service接口的具体服务

Log模块：用来调试程序的，在终端打印出程序运行的过程，可以很快找到出错的位置。

**前端：**

Navigate模块：在app底端的导航栏

State模块：简单的显示该页面处于哪个页面，位于app的顶部

Container模块：主功能区，里面又划分为多个小模块

Comment模块：评论模块，很重要的部分

Message模块：显示他人对自己书中章节的评论，和他人对自己的回复。不提供用户之间的socket，毕竟这可能需要设计到消息中间件，目前还没有涉及到这方面。

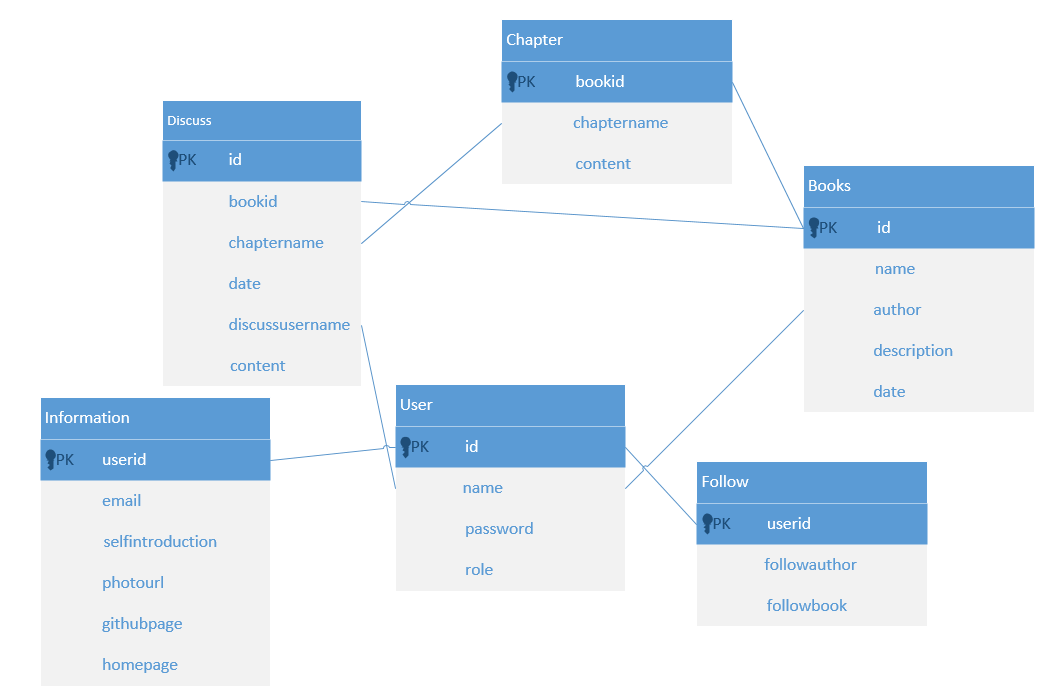
Books模块：显示从数据库中get的书名，作者，时间等信息，做一个列表往下铺。

Self\_information模块：显示个人信息，里面的数据可以从设置进入修改

Login模块：负责登录逻辑

前端的原型发布在axure云上，还是比较简陋的界面，没有配色和icon纯文字。见谅

* **数据库设计：**

****

上图是设计的原稿，下图是实际建立的库中的结构图，更加的详细和明确。建立了和Hibernate的联系之后：

