



Read Together

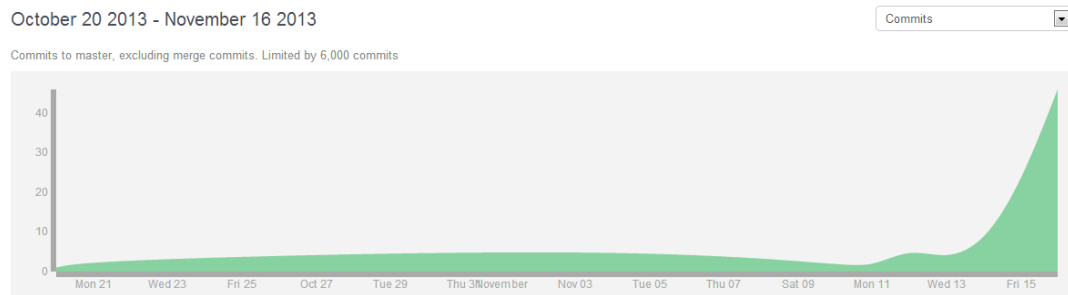
项目管理文档

1. Git9 使用情况报告

1.1 整体提交统计信息

从 10 月 21 日至 11 月 16 日，MarkasRead 小组在 Git9 上共提交 51 次。

1.2 Commit 界面截图



2. IssueTracker 使用情况报告

2.1 整体 issue 统计信息

从 10 月 21 日至 11 月 16 日，MarkasRead 小组共有 10 个 Issue。

2.2 issue 界面截图

Filter by Labels: Any Assignee: Any Milestone: Any	
<input type="checkbox"/> #370 安全性	assigned to 武祥晋 1 安全性 updated about 1 hour ago
<input type="checkbox"/> #340 ugly code....	assigned to 武祥晋 database updated about 17 hours ago
<input type="checkbox"/> #319 衔接汇总贴	assigned to 武祥晋 4 衔接 updated 1 day ago
<input type="checkbox"/> #285 【前端】易用性问题	assigned to 秦同 前端 易用性 updated 4 days ago
<input type="checkbox"/> #233 【git使用】branch?	assigned to 武祥晋 10 git updated 12 days ago
<input type="checkbox"/> #337 搜索结果页无?q=参数时异常	assigned to 武祥晋 1 updated about 16 hours ago
<input type="checkbox"/> #330 程序猿快饿死了，求喂食_(:3」∠)_	assigned to 秦同 5 饿了么 求喂食 updated about 22 hours ago
<input type="checkbox"/> #282 2个database表的同步操作	assigned to 涂珂 6 database updated about 16 hours ago
<input type="checkbox"/> #279 请前端先上传设计好的静态html页面	

3. 项目分工

姓名	主要职责	备注
武祥晋	■ 组长 ■ 前后端衔接	
涂珂	■ 数据库设计与管理	
傅左右	■ 页面设计	
秦同	■ 页面设计	
李响	■ 各种文档	
张思浩	■ 各种文档	

4. 相关文档

4.1 时间进度安排

2013.10.21-2013.11.3

进行初次讨论，确定了工作环境配置及各人的阶段性任务。

2013.11.4-2013.11.9

进行了第二次正式讨论，确定了原型系统内的具体实现细节。

2013.11.10-2013.11.16

进行了第三至第 n 次的会议，实现了各个模块间的衔接。

4.2 第一阶段的总结

LOGO:

还有 LOGO 没有设计！！

管理系统内部：

有消息系统？提醒（公文）系统？（比如 1 个星期前我禁用了某个人的权限，设定 1 星期之后发消息给管理员审批，发现那货不知悔改，继续禁用）<-（突然想到一个可能的实现细节问题，如果我设置自动 1 周后将权限设置为 true，同时那货一周之内要毕业了，结果就出现那货的权限 1 周内因为毕业设为 false，又因为系统没有对改变加优先级什么的，所以 1 周后因为处罚撤销被自动改成 true，显然是不对的）

书籍查询：

好像有一个书籍统一检索的方法.....

粗粗的搜了一下...叫什么...

SFX 是什么意思？

SFX 含义为内容相关的、敏感的链接服务，即图书馆为读者提供的除文献检索外的其它扩展服务，可以根据不同的记录内容提供不同的服务。例：对期刊（图书）来说，有电子版的可直接链接到相应的期刊（图书）、卷、期或全文，没有电子版的则可选择馆藏目录查询、文献传递、读者咨询、搜索引擎等扩展服务。它的最大特点是总能根据不同的文献情况提供不同的有针对性的服务。

参考链接：<http://www.lib.ncu.edu.tw/book/n39/39-2a.pdf>

这是检索语法的一个参考：

<http://innopac.lib.tsinghua.edu.cn>

（比如搜索关键字段，搜索语法，搜索范围）

杂项（不一定实现~只是乱想中(ノ▽ノ)）：

图书评级、图书借阅次数榜（评选最勤奋的读者？（我去...突然发现原来图书馆还能这么好玩）233）、在搜索出书籍的同时在网页右侧（像豆瓣一样）显示别人的评论啊评分啊购买价格什么的...

用户的帮助系统...

（还要指点一下用户如何更加正确的使用，第一目标是“保证网页设计的用户能够正常的使用”，有一些说明可以指导用户“更加正确而不出错的使用”）

书籍订购：

另外订购你们打算怎么管理？需求文档中写了么。好像对订购这种事情完全没提？是能够导出为订单由工作人员直接发给出版社？还是直接从网上订购？这些好像就没有说。

技术指标

技术指标好像完全没提能够支持多少用户多少本书.....

注册：

学号/工号（临时人员应该没有，但有临时来校的目的，如开什么会），用户名/昵称/真实姓名三者是否都有（建议只有用户名和真实姓名，更简单的可以用邮箱取代用户名），注意邮箱已存在、学号/工号已存在的情况。注册后是否要管理员审核。

登录：

验证码、忘记密码

管理系统中的公文系统

管理员之间可以分配任务啊...转发任务啊...向上级管理员申请什么的啊...

可能的逻辑失误

对了还有一个逻辑...就是借书的用户预约了他借的那本书怎么办...不能让他预约这本书...

有趣的小功能

诶...突然想到有趣的一点...如果可以用图书馆内的 wifi 对读者进行定位...然后在手机馆内平面图标记出读者想要的书籍的位置... (that's interesting! ~(\cong $\nabla \cong$)/~我就顺手一说~)

保存想读的书

啊! 对了! 我们还可以提供书架功能...就是有时候想看一本书...但是都被借走或者被预约了...但是我还想把它记下来...(相照应的在我们图书馆有一个功能叫做“保存检索式”...)

另外的参考样例，他山之石

@秦同 关于这个界面你看如何? 纽约公共图书馆...好像国外有几个数据库的检索界面都跟这个也差不多

[http://nypl.bibliocommons.com/search?t=smart&q=圣诞
&commit=Search&searchOpt=catalogue](http://nypl.bibliocommons.com/search?t=smart&q=圣诞&commit=Search&searchOpt=catalogue)

用户体验

啊...我们是不是漏了考虑图书的封面(图书属性)，搜索的时候可以辅助显示，这样读者就能一眼看出是不是想要的书，部分书有就行了

4.3 2013-10-30 分配任务 会议记录

概述:

由于缺乏总体了解，我们要先动起来，有点体会了以后才能写出特别细节的设计文档。因此先简单根据网页的页面做如下规划，作为最初的指导。以下基本单位为页面，每个页面包括其承载的功能和部分页面设计。不包括用户不可见的页面，暂不考虑与系统整体关系不大的枝节功能（如用户行为记录）。

根据这一版本的规划，男主、系草了解一下设计文档的结构与写法，作好准备并先写与细节无关的部分。右右、萌用图片或静态 HTML 设计有关界面，至少有主页的 HTML 版。神霸了解 Django 与 MySQL 联用，主要是 Object-Relation-Mapping，若是太烦也可以考虑纯 Python 直接操作 MySQL 的关系数据表。昨天的讨论好像没有把要存什么讨论得很细，不过我会在下面先给一个 toy model。我先把 Apache+Django 的环境做好，完成网站运行最基本的框架，使得大家可以联动起来，以便进行接下来的开发。

这一部分 **Deadline** 为本周日晚，然后讨论下一步的任务。尤其要把数据明确化。

页面汇总：

首页、公告列表页、公告内容页、榜单页、搜索结果页、图书信息页、用户注册页、用户个人中心、用户信息修改、图书管理员个人中心、借出页、归还页、上架信息录入页、图书信息修改页、批量下架页、用户管理员个人中心、**root** 中心。

首页：

LOGO、搜索框、最新公告、榜单、登录入口

搜索框暂时不考虑各种选项，如书名、作者名、图书编码，只有一个框。

榜单可以只显示主榜单，也可以多榜单切换。具体有哪些榜单待确定。

登陆入口不区分用户/管理，在登录时根据用户名区分跳转。

公告列表页、公告内容页

哪个管理员负责发布公告？！

列表页显示公告题目列表，时间排序，可翻页。

公告分为长期的（如用户守则）和临时的（如开闭馆公告）。

内容页显示一条公告的具体内容。

榜单页

罗列各榜单的情况。

搜索结果页

搜索结果对应的图书列表，应该提供除题目外的简单信息如作者、出版社、版次。**Filter** 和各种排序先不考虑，默认“神奇排序”。

图书信息页

以 **ISBN** 为标准确定是否有独立的图书信息页，页面显示与图书信息相关的详细信息，以及这本书每一个 **copy** 的情况。

用户评论使用 **AJAX** 动态加载，无独立页面。

用户注册页、用户信息修改页

按照需求文档设计一个表单即可。

用户个人中心

显示简要个人信息、最近借阅记录、评论记录。

借阅记录保证未还全部显示，可以取消预约、赔偿，更老的 **AJAX** 动态加载。

评论记录太多也 **AJAX** 还是独立页面？

图书管理员借出页、归还页、上架页、图书信息修改页、下架页

都是表单，上架、下架支持批量。注意操作单位是一个 **copy** 而不是一个 **ISBN** 对应的书。

图书信息维护包含可删除对本书的评论。

注：**copy** 删除不影响 **ISBN** 对应的图书信息页，赔偿行为由用户操作。

图书管理员个人中心

昨天忘了讨论这里有什么，难道图书管理员一登录直接跳到借出页之类的具体功能页面？

也许可以有上下架记录、图书信息修改的 **log** 什么的.....

用户管理员个人中心

登录后显示各种待“处理”用户的列表，如等待提权、等待注册审批、黑名单（手动洗白）

用 **AJAX** 实现一个搜索/**filter** 框，便于找到想操作的用户，以实现手动拖黑、删除用户等主动管理。

ROOT 中心

增删管理员、清理数据库等。

我们可以先不实现细节，留一个空页面。

数据模型

以上很多功能要存储的数据还都不够细，所以可以先考虑如下内容：

用户信息：用户名、密码、邮箱，（参见需求文档和文档君）

管理员信息：不独立只是对用户加一个域表示？0 是 **root**，1 图书 2 用户，10 普通用户

图书信息：参见需求文档和文档君

图书 **copy** 信息：独立还是合并入图书信息呢？

评论

借阅行为：参见需求文档

提降权申请、待审批注册的用户（总之就是用户管理员要面对的待“处理”用户列表）

榜单的缓存：（虽然榜单可以根据借阅行为计算，但是定时计算然后缓存会好一点）

4.4 内部设计约定

按照以下顺序开发：图书查询、用户信息、借阅行为、公告、评论、后台管理、榜单等。这阶段至少做到图书查询、用户信息、借阅行为，当然公告也就是顺手的事。

主页、导航

导航

主页、公告、榜单、搜索框、(登录+注册)/(个人主页+登出)

主页

{% url 'rt:index' %}（下文提到主页时还会写出 url，大家可以顺便帮忙检查下是否一致）

公告

{% url 'rt:info' %}

榜单

{% url 'rt:rank' %}

搜索框

以下页面不显示：主页、搜索结果页建议将公告页的搜索框去掉，换成导航条的小搜索框。你是怎么想的？不显示是因为页面上有更大的，或是为了风格统一设计简单都显示？@右右@萌（我感觉还是统一的好，要不然搜索条一会有，一会没有好奇怪的。。。）跳转地址和参数名见“主页”->“搜索框”

(登录+注册)/(个人主页+登出)

细节参见用户部分，比较复杂这里先不写。

主页

{% url 'rt:index' %}

LOGO、搜索框、主榜单、短时公告、常驻公告

搜索框

两个按钮应该怎么设计？都是 submit 带不同参数还是怎么的？能不能先弹一个“功能开发中”.....

请用 GET 方法发到{% url 'rt:search' %}，搜索字符串 name 为 q。（暂定，Django 的 URL 设计哲学和 Form 处理机制我还没看）

主榜单（详见榜单页说明，此版本暂不涉及）

变量名 rank，是一个 list，请使用 for..empty 语句显示“当前没有榜单信息”之类的。

短时公告（详见公告说明）具体参见[数据库文档](#)的 **Info** 部分

变量名 news，是一个 list，每一个 item 有以下域：title(str), date(datetime), id(int) 请使用{% url 'rt:info_detail' item.id %}链接到公告的具体页面

常驻公告（详见公告说明）

变量名 guide，与 news 接口一致。

图书查询

搜索结果页

{% url 'rt:search' %}

Django 的 Form 处理我还没看，但是保证以下功能：

变量 q(str)用于在搜索框内显示当前查询字符串。

变量 result(list)，每一个 item 包含以下域：具体参见[数据库文档](#)的 **Book** 部分
simple_name(str), author(str), simple_version(str), id(int)

请使用{% url 'rt:book' item.id %}链接到图书细节页

至于更丰富的图书信息再慢慢扩展吧。

本阶段可以不考虑分页。

图书细节页

{% url 'rt:book' book.id %}

变量 `book` 包含以下域：具体参见[数据库文档](#)的 `Book` 部分

- `id(int)`
- `simple_name`, `author`, `simple_version(str)`

变量 `copy(list)` 的每一个 `item` 包含以下域：具体参见[数据库文档](#)的 `BookCopy` 部分

- `status(int)`, 用于判断是否显示借阅或预约按钮
- `id(int)`, 用于生成借阅('rt:borrow')或预约('rt:queue')链接。

用户

登录

点击导航条出现浮动窗口，登录后跳转到个人主页，失败跳转到主页。

POST 到{% url 'rt:login' %}, `name` 分别为 `username` 和 `password`。

需要登录的界面如果未登录，统一跳转到主页。怎么提示登录？

注册

点击导航条出现浮动窗口，注册成功跳转到个人主页，失败跳转到主页。

POST 到{% url 'rt:register' %}, `name` 以及具体有哪些域参照@神霸。

注册所需内容：

- 用户名
- 密码
- 邮箱
- 姓名(昵称)

登出

直接链接到{% url 'rt:logout' %}然后跳转到主页？这样实现最简单.....

个人主页/个人中心/个人信息页

{% url 'rt:user' %}

变量 `user` 包含以下域：

- `username(str)` 用户名
- `email(str)` 邮箱

变量 `profile` 包含以下域：

- `name(str)` 真实姓名
- `get_group_name(str)` 分组（默认为 `NormalUser`， 应该为 `NormalUser`, `AdvancedUser`, `Blacklist`, `Admin` 之一， 可申请提权）

变量 `action(list)` 的每个 item 包含以下域：

用户信息修改

借阅、预约

预约的链接 `{% url 'rt:queue' copy.id %}` 是用 AJAX 吗？否则我该显示什么或者跳转到哪里.....如果是 AJAX，我就返回 `{'user':uid, 'copy':cid, 'status': 'OK'}` 或者 `{'error': 'Invalid user. / Invalid copy. / Copy not available.'}`

公告

公告列表页

`{% url 'rt:info' %}`

变量名 `news` 和 `guide`，是 list，每一个 item 有以下域：具体参见[数据库文档](#)的 Info 部分

`title(str)`, `date(datetime)`, `id(int)`, `content(str)`

请使用 `{% url 'rt:info_detail' item.id %}` 链接到公告的具体页面

常驻公告的 `pub` 表示最近一次修改时间，临时公告一般不会修改相当于发布时间

公告细节页

`{% url 'rt:info_detail' item.id %}`

突然想到现在的设计里，公告内容是直接在列表页就能看见的！那么还要细节页吗？还是“常驻公告”有独立页面（因为用户守则一般比较长）而临时通知没有？

4.5 数据库部分细节

这部分描述了数据库实现细节，有什么的地方或需要的接口函数写在 **Issue** 里吧

目前分组与权限：（参见需求文档）

```
group_list = ['NormalUser', 'AdvancedUser', 'Blacklist', 'Admin'] # guest
permission_list = ['can_search', 'can_comment', 'can_manage']
permission_num_list = ['borrowing_num', 'borrowing_coefficient', 'queue_book_num']
```

其中 `'can_manage'` 为权限

`'can_midify_book'`, `'can_change_perm'`, `'can_generate_tempuser'`, `'can_manage_blacklist'`, `'can_delete_user'` 的集合

MYUSER

域

- user, OneToOneField 一对一表关系到 django 自带的 User 模型。这个模型包含：
 - username 用户名
 - password 密码（加密后）
 - email 邮箱
 - groups 分组('NormalUser', 'AdvancedUser', 'Blacklist', 'Admin' 之一，默认为 NormalUser)
 - permissions 权限(由分组决定其权限，目前没有为每个用户单独设置权限)，目前只有三种权限('can_search', 'can_comment', 'can_manage')
- name, 真实姓名（昵称）

接口函数

- register(username, password, email, name, group='NormalUser'), 注册，**必须保证传入数据合法 #282**
- set_group(group) 更改分组,用于拉黑，提权等
- get_group_name() 得到当前分组名
- get_group() 得到所有分组类的数组(应该用不到)
- has_perm(string) 判断是否有某种权限，传入 string，返回 bool 值
- get_perm(string) 返回数字权限的对应值

BOOK

域

duartion 借书期限，为 0， 7， 14

- name_cn 中文名，若为中文书、中译本或影印本则直接为书名，若对于外文原本则赋空值。
- author 作者名：中英文名称皆支持。多个，“,”分割
- press 中英文名称皆支持。
- pub_year 出版年份：代表公元纪年 4 位数字。
- revision 版次：一个阿拉伯数字。
- ISBN 根据国际标准，每本书都有一个对应的 ISBN 码。
- name_origin
- translator 多个，“,”分割
- pub_year_origin
- revision_origin
- type 分类：对于图书的类型定义，如文学，数学，计算机等。

接口函数

- `simple_name()` 返回简单书名
- `simple_version()` 返回简单版本信息
- `search(string) staticmethod`, 返回搜索结果的 `queryset`

BOOKCOPY

域

- `book` `ForeignKey` 连接其对应的 `book` 类
- `location` 馆藏位置

接口函数

`get_status` 返回一个 `dict` 类型的值 `status`。`status['text']` 为 `'borrowing'`、`'arranging'`、`'disappear'`、`'on shelf'`，分别对应出借中、（归还后）整理中、已丢失、在架上。当 `status['text']` 为 `'borrowing'` 时，另有 `status['expire']`、`status['reborrow_count']`、`status['queue']` 分别表示到期时间、已续借次数、当前预约人数。注意，到期时间可能已经超过 `today()`，即应为“已超期”。

INFO

域

- `title` 标题
- `content` 内容
- `date` 时间（自动存入创立时的时间）
- `species` 种类，整数域，存入 `species_choice=((0, 'news'),(1, 'guide'))`

接口函数

- `local_time()` 返回本地时间，一个类
- `get_all(sp=None)` 返回 `sp` 为某种类的所有 `Info`（`sp` 传入 `news` 或 `guide`），默认返回所有
- `get_species_display 状态(str)`，即上一条的对应关系

BORROWING

域

- `status` 状态(需传入 `int`,如下) `status_choice = ((0, 'borrowing'),(1, 'reborrowing 1'),(2, 'reborrowing 2'),(3, 'arranging'),(4, 'queue'), (5, 'disappear'))`
- `datetime` 发生时间
- `book_copy` 对应的书的副本
- `myuser` 对应的用户

- `is_active` bool 值，记录是否已经失效
接口函数
- `borrow(myuser, book_copy)` staticmethod, 用户 `myuser` 去借 `book_copy`
- `reborrow(myuser, book_copy)` staticmethod 用户续借
- `return_book(myuser, book_copy)` staticmethod 用户归还
- `queue_next(book_copy)` staticmethod 管理员把归还的书给最先预约的人
- `readify(book_copy)` staticmethod 管理员把归还的书上架
- `disappear(myuser, book_copy)` staticmethod 用户将图书丢失