软件设计实践报告

单 位： 信息工程学院

班 级： 15网络工程2

学 号： 2015551634

姓 名： 曾碧霄

任课教师： 叶松涛

湘 潭 大 学

2018年6月

软件设计实践

**项目名称：魔卡**

湘潭大学信息工程学院 王康，曾碧霄

1. 项目概述
2. 概述

随着城市快节奏的发展，项目多、重高效的工作、生活状态促使越来越多人倾向于选择计划性、自律性的生活方式。人们通过自制计划表、勾选任务栏等形式来记录自己一天的工作、生活足迹。然而手工制作表项、填涂选框总是费时且麻烦的，而且有时也会试着把每天的计划偷工减料，更重要的是每天一张纸的开销也不利于环保。“魔卡”应用旨在抛弃传统的记手账模式，简洁易上手的操作界面让用户轻松打卡，即刻生成当天的打卡报表，用户可以随时查看自己当天所完成的任务和相应历史完成次数。当日所有任务按用户自定义计划完成时，“魔卡”会提示用户今日任务已全部完成，激发用户完成任务的成就感，如果用户在某天还没有打卡，“魔卡”也会严格的发出提醒。极具特色的是，“魔卡”十分“严苛地”维护历史数据，用户一旦新建某项任务，该任务就必须每天都要完成，这将准确服务于容易偷工减料、半途而废的用户，帮助其在严格的任务管理中实现自我约束。

1. 目的和用途

“魔卡”希望能让用户爱上有计划、有总结、环保的打卡方式。用户可以用“魔卡”来自定义当天计划，当用户需要打卡时，只需要轻松一键便可勾选任务完成一次打卡；用户可以用“魔卡”来查看当前计划完成进度，进度达到百分之百时，用户将得到“今日任务已全部完成”的提示；用户可以用“魔卡”来实现自我约束，每个任务一旦被创建，用户将确保自己每天都完成一次，“魔卡”更像是一位严格的教练，监督着用户完成自己最初的规划。

1. 需求说明
   1. 业务需求
      1. 新建任务

用户对当天需要完成的任务有增加需要时，可以创建一个新的任务。注意从此以后的每一天，用户都需要完成这项任务的打卡，不可中途放弃，否则视为当天任务没有完成。

* + 1. 任务打卡

用户可以通过一键勾选的简单操作完成对某项任务的单次打卡。

* + 1. 今日报表

用户通过查看今日报表可以了解到当天的任务完成情况。具体来说，当某项任务今天已完成时，可以看到该任务的历史已完成次数。当所有任务都达到计划要求时，判定今日任务已全部完成。另外，如果今日还没有开始打卡，用户将得到待打卡的提醒。

* + 1. 个人中心

用户的所有注册信息都可以在个人中心找到。包括账户、邮箱、注册日等个人信息。

* 1. 非功能性需求
     1. 时间特性要求

每天的报表在凌晨将定时重置，具体来说，每项任务的勾选框将出现在界面中，用以服务用户新的一天打卡。

* + 1. 灵活性要求

用户当天有增加新任务的要求时，可以允许用户新建当前未被创建的任务，如果该任务是已被创建过的，用户将得到新建无效的提示。

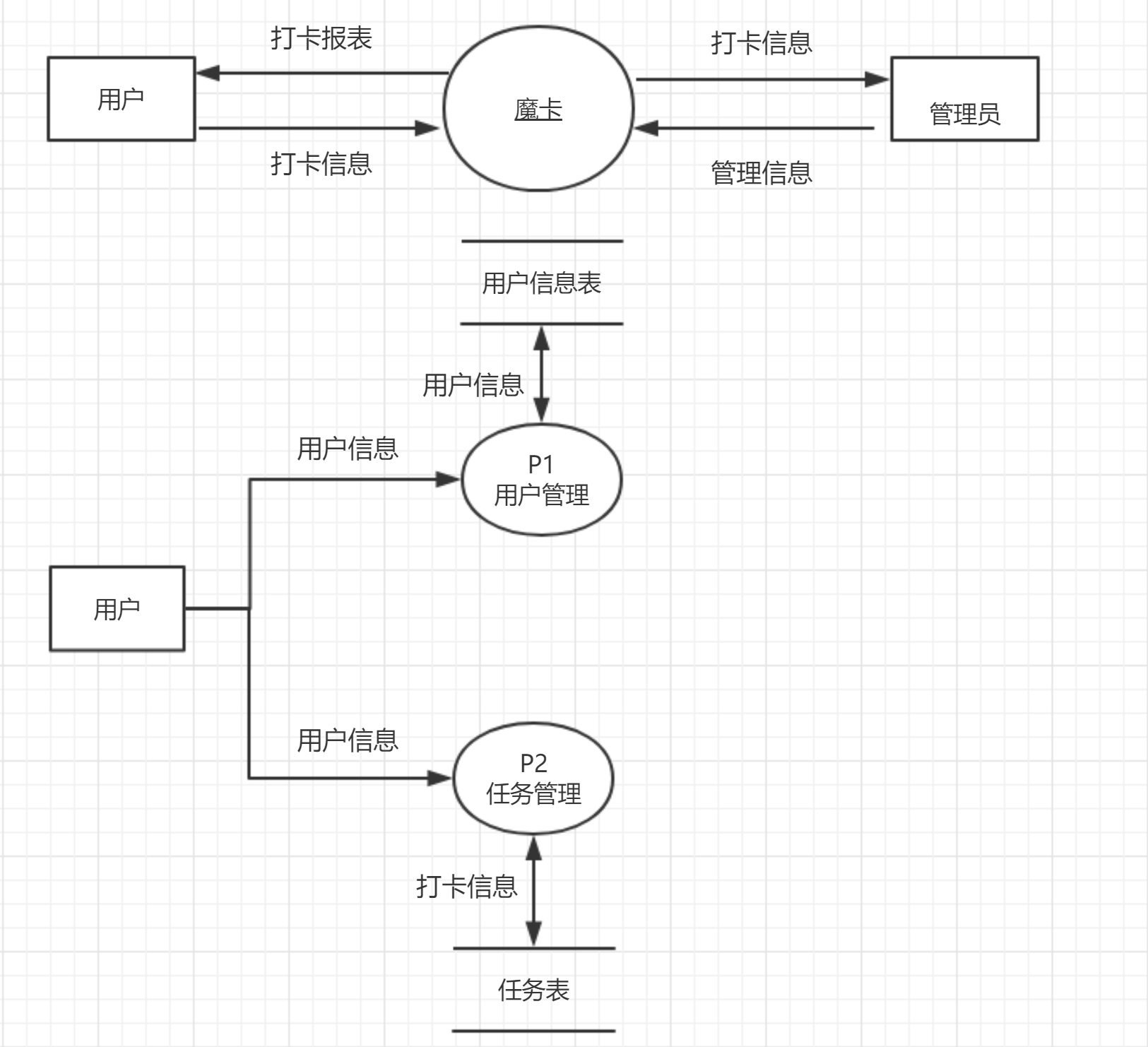
* + 1. 数据管理能力要求

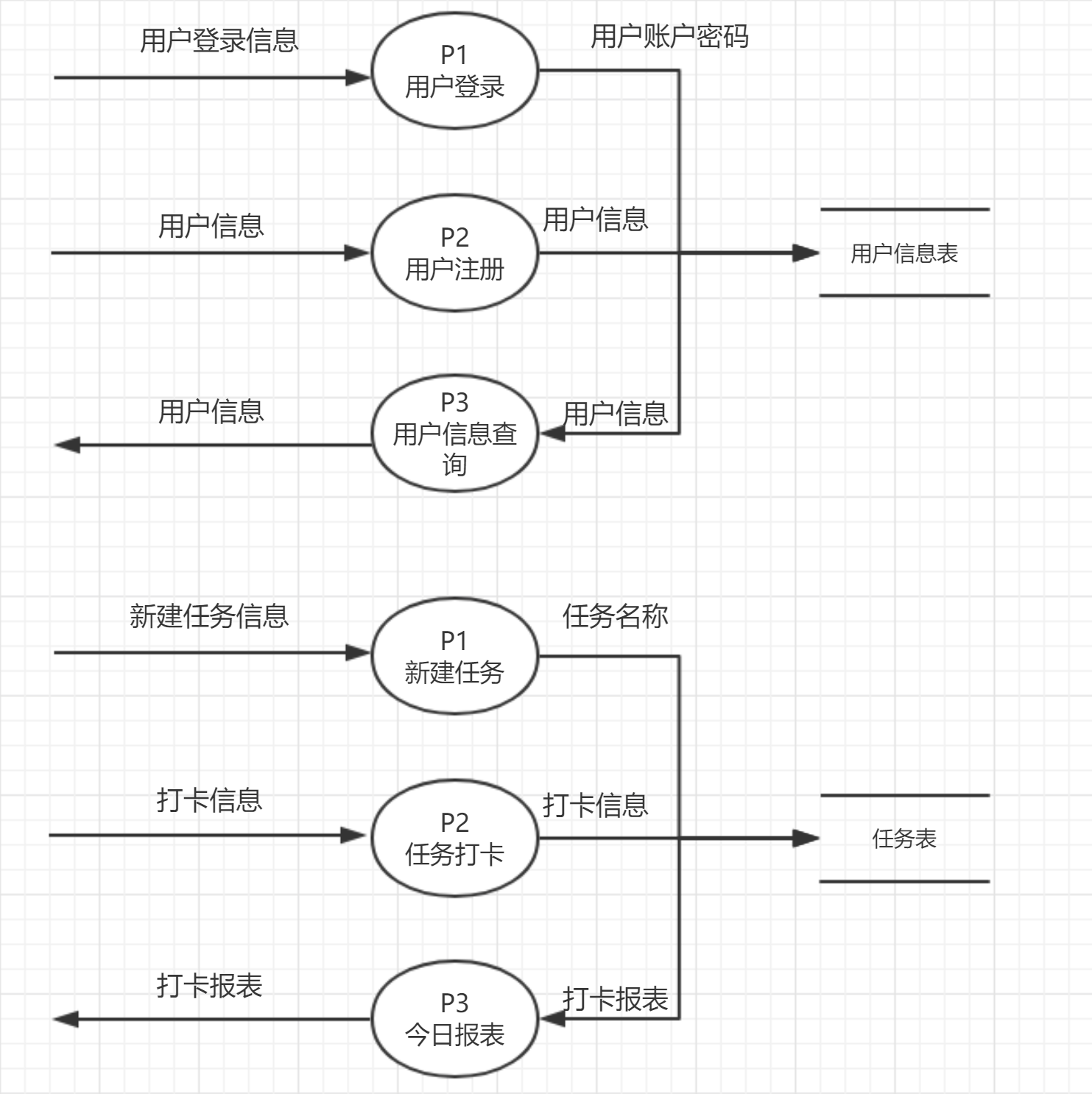
数据库需要维护每项任务已完成的历史次数，实现每次打卡后的次数累加。当有新任务被创建时，一个新的表项也要对应生成。

1. 环境要求

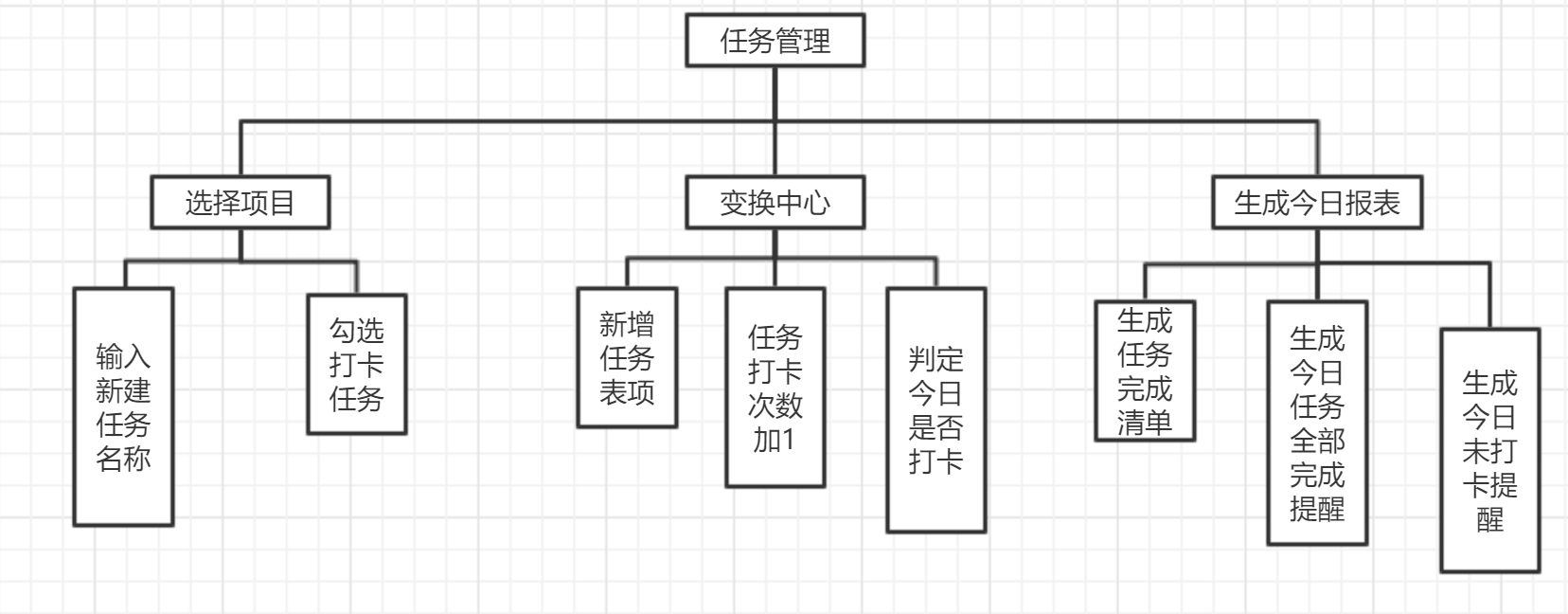
Win10或Linux php5.6+ Apache mysql

1. 系统分析与设计
2. 系统分析
   1. 数据流图





* 1. IPO图



* 1. 数据字典

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 描述 | 定义 | 有关数据流 |
| 用户 | 需要打卡、创建任务的用户 |  | 用户信息、打卡信息 |
| 管理员 | 管理用户信息、打卡信息的工作人员 |  | 用户信息、打卡信息 |
| 用户信息表 | 存储用户的账号密码等注册信息 | 表项=id+name+email+password+创建时间+最近更新时间 | 用户信息 |
| 任务表 | 存储所有用户创建过的所有任务信息 | 表项=id+project+stime+etime+打卡次数+最近打卡时间 | 打卡信息、打卡报表 |
| 今日报表 | 展示用户已完成的任务清单 | 负责展示该用户所有任务的已完成次数 | 打卡信息、打卡报表 |
| 打卡报表 | 表示系统传出的报表数据 | 任务完成次数、今日未打卡的提醒、今日任务全部完成的提示 | 打卡报表 |

1. 系统设计
   1. 设计人-机交互界面

* 初始化



* 登录



* 注册



* 未打卡提示



* 创建任务



* 打卡



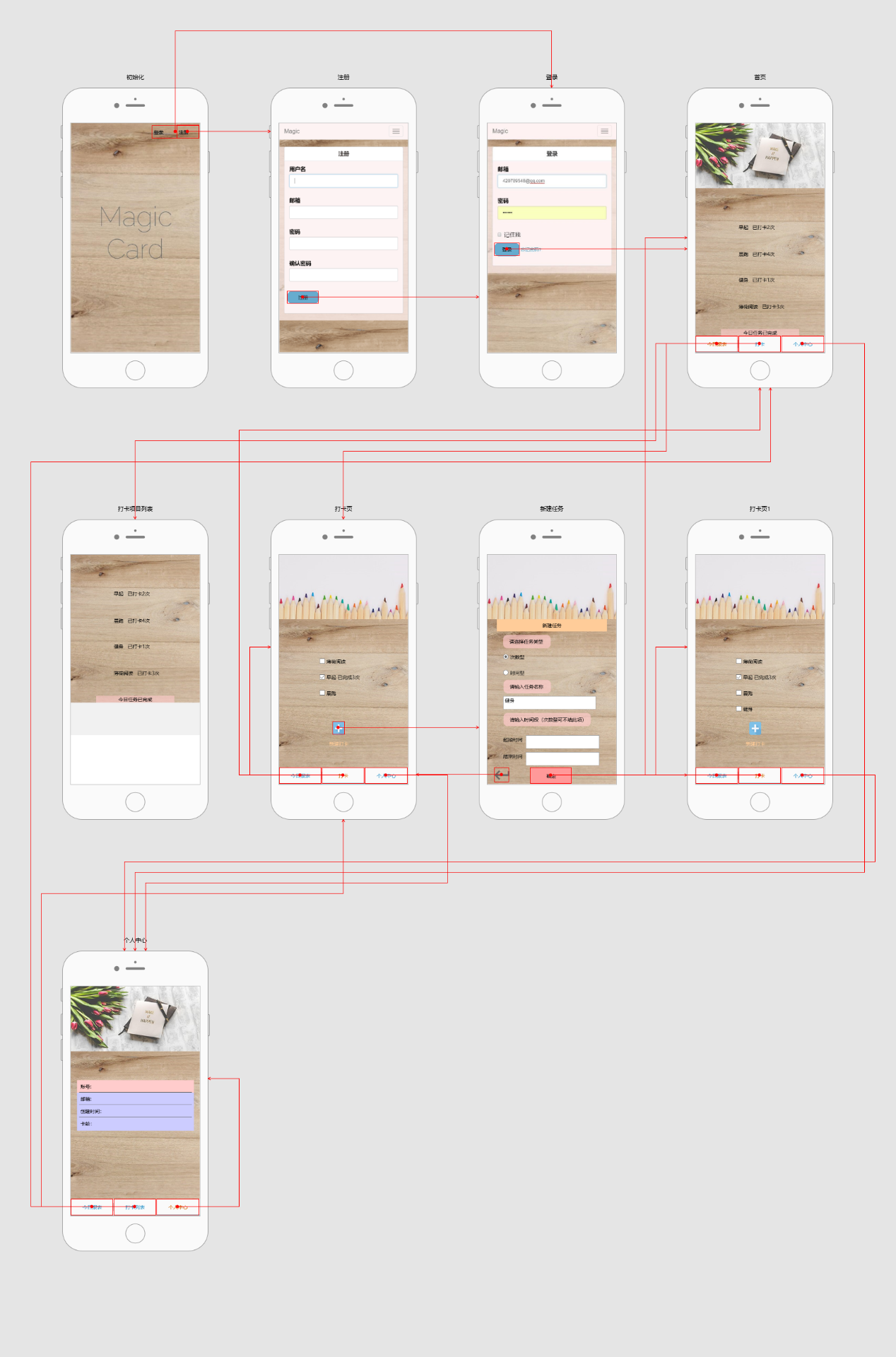
* 今日报表



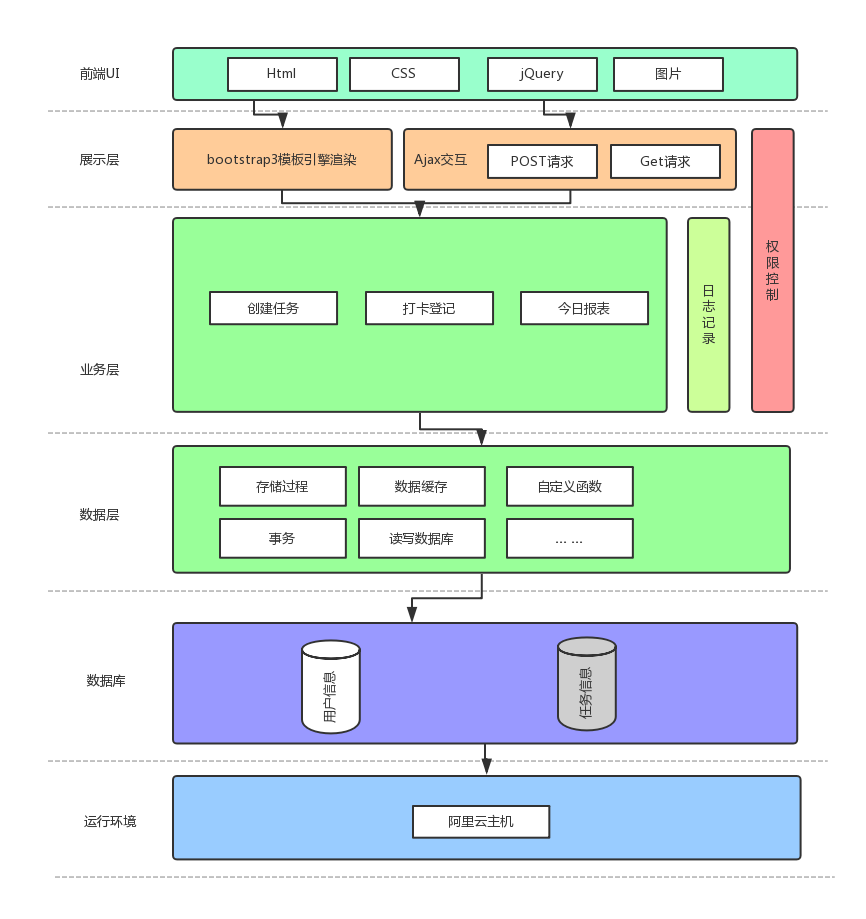
* 个人中心



页面流程图：



* 1. 系统结构图



1. 模块/函数设计

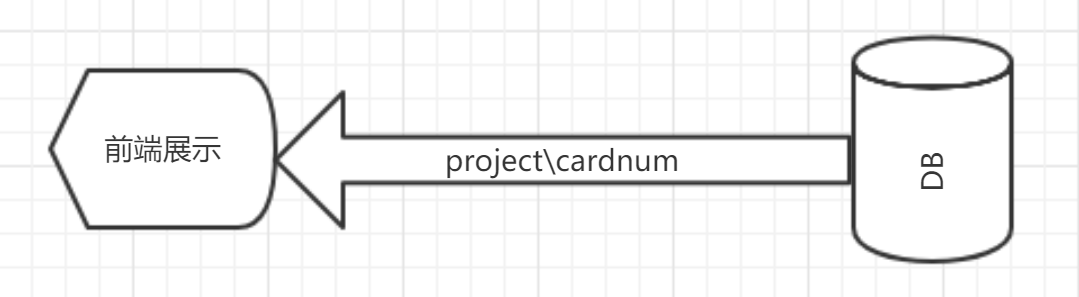
* home.blade主页模块

功能：判定用户今日任务是否完成，若完成则接收数据库返回的所有任务的历史次数。

数据结构：marktime 表示最近一次打卡时间

project 表示打卡任务名称

cardnum 表示任务被打卡的历史次数



* create.blade创建任务模块

功能：创建该用户历史从未创建过的任务，将任务表单传递给数据库。

数据结构：project表示被创建任务名称

type 表示被创建任务类型,其中1表示次数型任务，2表示时间型任务

stime 时间型任务的起始时间

etime时间型任务的结束时间

botton类型的“创建”按钮



* submidmark.blade返回打卡信息模块

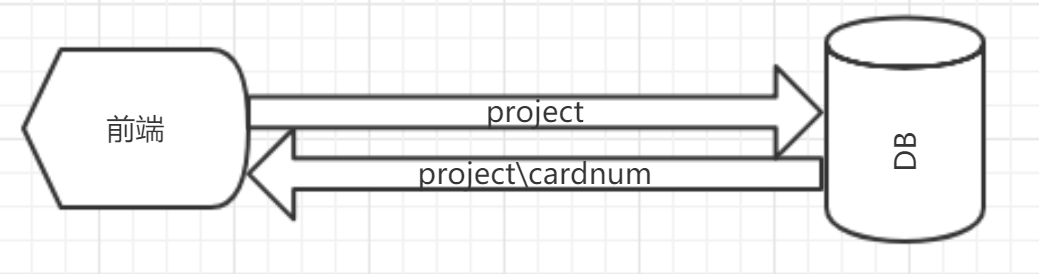
功能：将用户打卡的若干任务返回给数据库，接收数据库返回的已完成任务信息。

数据结构：project 表示待打卡或已被打卡的任务

cardnum 表示今日已完成的任务的历史完成次数

marktime 表示任务最近一次的打卡时间

button类型的“打卡”按钮



* user.blade个人中心模块

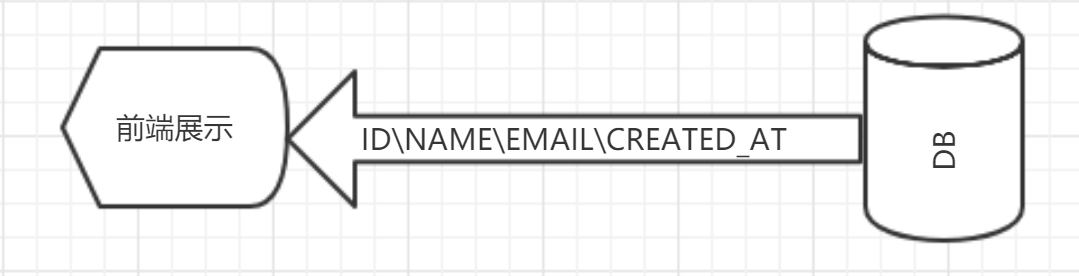
功能：接收数据库返回的个人注册信息。

数据结构：ID 表示该用户的编号

NAME 表示该用户的昵称

EMAIL 表示该用户的注册邮箱

CREATED\_AT表示用户的注册时间



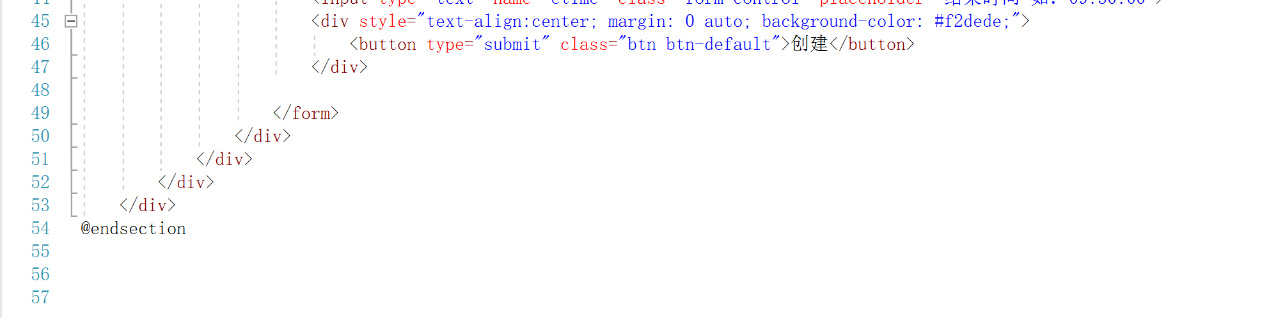
1. 测试方案
2. 4.1 功能测试
3. 测试用例：新建用户、利用账号登录，新建任务打卡
4. 测试工具：手动测试
5. 4.2 性能测试
6. 测试用例：新建大量任务
7. 测试工具：手动复制
8. 4.3 负载测试
9. 由于还未成功上传至服务器，所以负载均衡还未能测试
10. 4.4 容量测试
11. 个人数据库只有20M，容量较小
12. 4.5 安装测试
13. 直接安装apk包成功
14. 系统实现
15. 关键模块/函数的实现

* home.blade主页模块



* create.blade创建任务模块

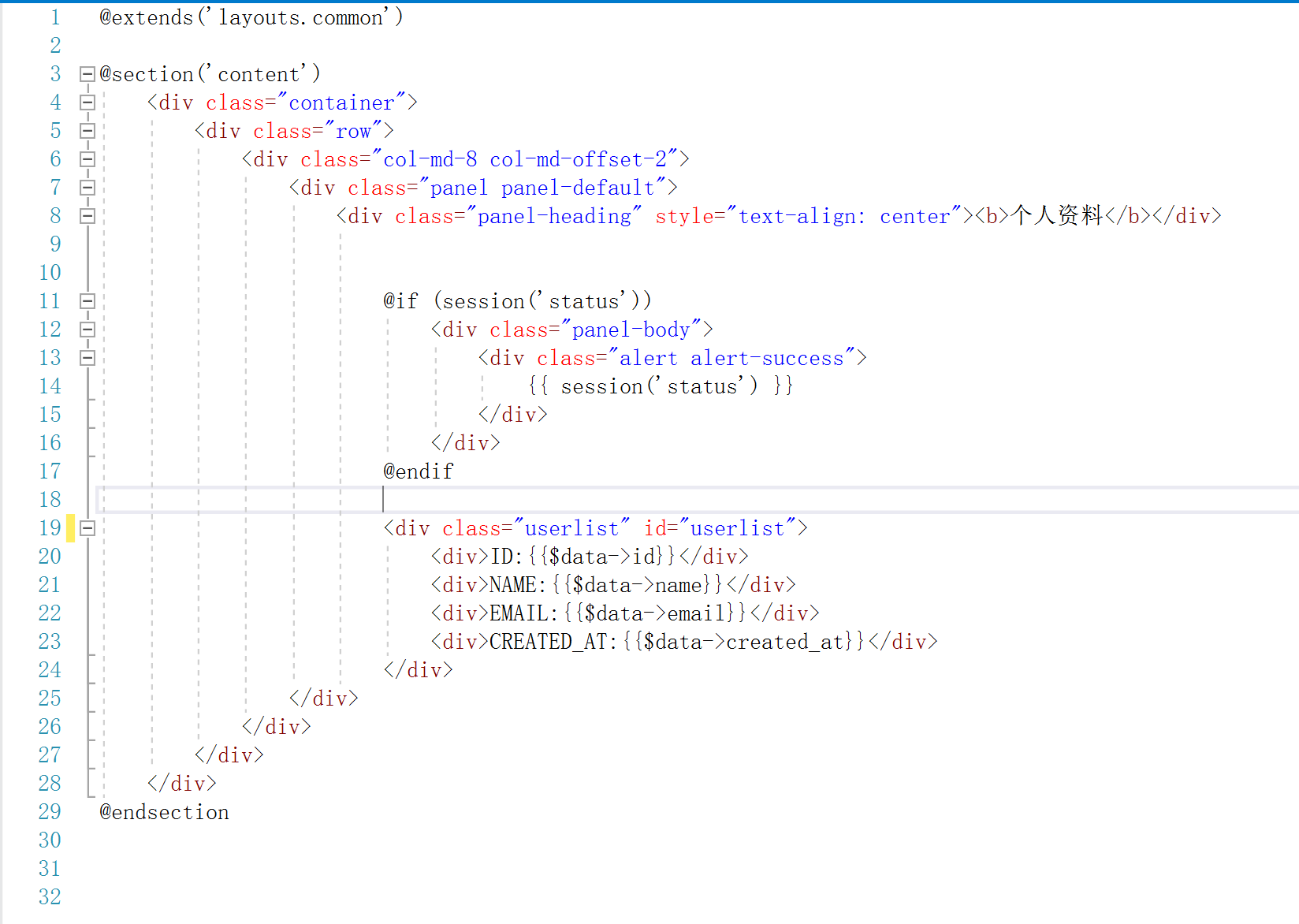




* submidmark.blade返回打卡信息模块



* user.blade个人中心模块



1. 系统安装与使用

系统安装：

项目经由Hbuilder打包成APK安装包，Android手机只需直接下载安装安装包即可。或者用装有PHP环境的PC，在项目根目录输入命令：php -S 127.0.0.1：：8080 -t public，通过浏览器访问即可。

使用：

2.1 进入主页面



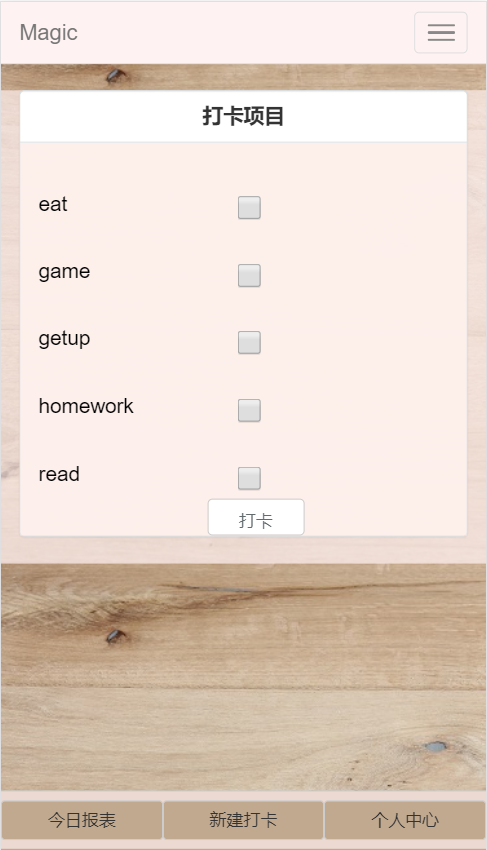
2.2 如果有账号则直接登录



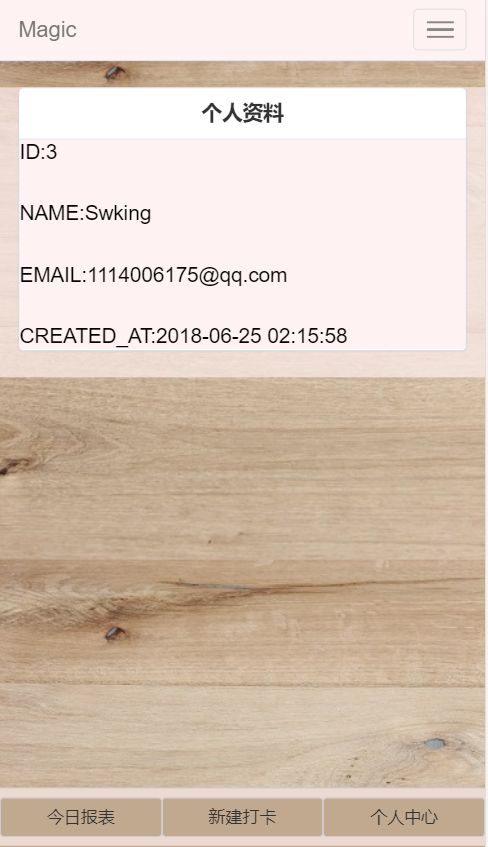
2.3 如果没有账号则注册



2.4 登录后进入今日报表，如果没有任务则需要新建任务



2.5个人中心



1. 测试分析
2. 测试与调试

测试之前需要准备php的开发环境，可以用warmserver或者phpstudy或者Laravel自带的服务器（命令行：php -S 127.0.0.1:8080 -t public）浏览器访问即可。

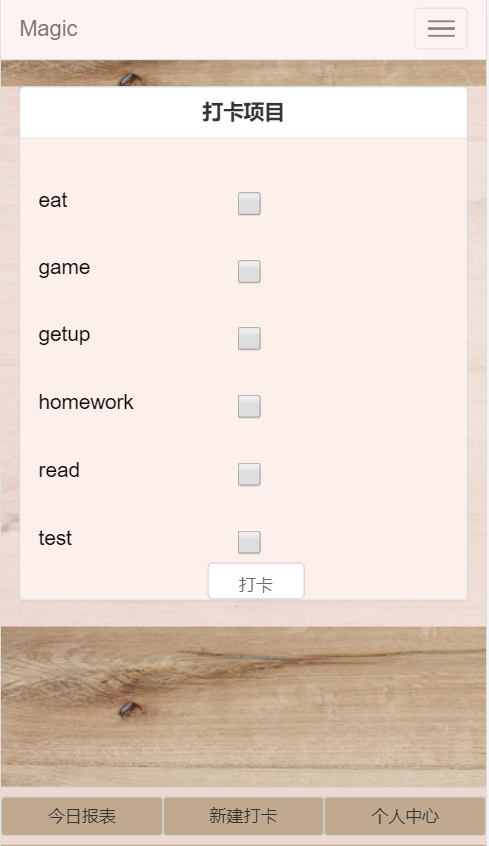
通过对系统功能的验证，人工体验软件功能进行测试

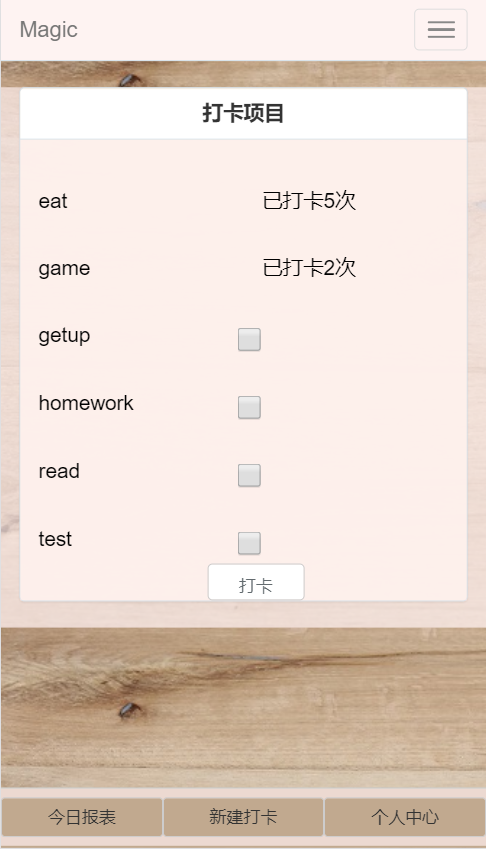
1.1 新建任务测试





1.2 打卡测试







1. 测试结果分析

新建任务时，任务分为两种类型，对应是否填写时间段，其实后端有方法判断任务类型，抉择时间段是否录入数据库，可以避免用户输入错误的情况。

打卡时可以一次完成多个任务或单个任务，当打卡后，对应打卡的任务会显示打卡多少次来提示用户今日已经打卡，如果还有项目未完成，则提示用户需要打卡，否则则提示今天任务完成。后端设计了打卡时间标志，使得任务列表每天更新一次。

1. 结论

新建任务测试和打卡测试基本实现项目的需求。用户可以根据自身需求设计自己的打卡任务，创建后用户则需要每天坚持打卡，否则不记录当天的打卡信息，这样有利于激励用户积极打卡完成任务，从而养成习惯。

1. 设计体会
2. 设计过程中的问题及其解决方案

问题：项目刚开始时，想要实现的功能很多，不知从何下手，也不确定各项功能的实现是否可行。

解决方案：从宏观到局部，比如先设计顶层数据流图，再逐个细化各类加工；从简单到复杂，比如先将勾选某项任务的信息传递到数据库这个功能实现，再去扩展一次性勾选多个任务的情况。

问题：急于实现前端到数据库的信息传送和接收，未与设计数据库的队友充分交流。

解决方案：耐心等待数据库的结构设计确定以后，再进行前端的信息收集和发送处理。

问题：得知每天需要做一次任务重置，即凌晨以后所有任务的勾选框重新显示，而不再显示已完成次数，如何实现当天任务重置？

解决方案：在home.blade里面设置一个marktime用以记录最后一次打卡的时间，将marktime与当天凌晨的时间对比，如果小于凌晨则默认为今天还未打卡，向数据库请求待打卡任务名称并显示勾选框。

1. 设计过程中的错误及其原因分析

错误：首页设定的是打卡界面，用户登录后见到的第一个界面就是打卡。但“今日未打卡”的功能提醒是出现在今日报表界面的，无法满足“收到提醒”->“开始打卡”这一顺序要求。

原因分析：既然“今日未打卡”的提醒出现在今日报表界面，那么应设置首页为今日报表的界面，这样才能够先收到提醒再开始打卡。

错误：交互界面设计的时候，新建任务界面点击确定以后原任务清单并没有发生变化。

原因分析：新建任务点击确定以后的页面链接设置出了问题，应该跳转至新的打卡页，即增加了任务栏的新打卡页。

错误：注册以后直接进入首页。

原因分析:注册成功的链接设置应跳转至登录界面，登录以后才可进入首页。

1. 体会和收获

以小组合作的形式完成一个app项目时，我很赞同顺序分工、适当并行的模式：前端设计基本框架和模块，后端建立数据库、E-R模型，前端再向数据库传递数据和接收数据库返回的数据。这样的分工使得组内成员的分歧会比较小，前端充分考虑后端给予的技术可行性意见，后端配合前端的模块设计。还有一点感想是：我一直秉信，如果能力不够，就要谦虚的向队友讨教，并做些自己力所能及的事，与人合作时要多思考自己能做什么，而不是冷眼旁观，否则会影响到队友的效率，自己的能力也不会得到任何提升。我很开心在大学生活结束之前终于能与自己赞赏的、相信的队友一起完成项目，这是一次真正意义上的合作（当然队友比较辛苦），喻之圆满。

1. 附录(配置清单)
2. 在PC端使用项目的配置
   1. 安装PHP+Apache+Mysql
   2. 使用命令行进入项目根目录
   3. 命令行输入php -S 127.0.0.1:8080 -t public，浏览器访问127.0.0.1:8080即可（浏览器大小调制手机大小）。
3. 项目的数据库信息配置

DB\_CONNECTION=mysql

DB\_HOST=bdm135157202.my3w.com

DB\_PORT=3306

DB\_DATABASE=bdm135157202\_db

DB\_USERNAME=bdm135157202

DB\_PASSWORD=\*\*\*\*\*\*\*\*\*\*\*

DB\_PREFIX=mc\_

1. 参考文献

Bootstrap3 中文文档  http://v3.bootcss.com/components/

1. 基本信息
2. 项目小组成员

曾碧霄 王康

1. 各成员分担的实验内容

曾碧霄：完成前端版块（个人中心，新建打卡，今日报表）的设计，并负责设计收集新建打卡与今日报表的表单数据传至后端，同时设计接口接收后端反馈的数据信息实时反馈给用户。

王康：完成后端数据处理部分。通过对前端传来的数据进行分析处理，将前端的新建打卡数据通过分析后插入到数据库的项目表中，将今日报表的打卡情况的前端请求信息从数据库取出发送给前端并实时更新数据库的打卡数据。

1. 设备编号

windous ID：00342-30262-00002-AAOEM

1. 实验时间（每次都要记录）
   1. 2018年06月12日 小组成员共同确定项目方向
   2. 2018年06月14日 小组成员共同对项目的需求进行分析
   3. 2018年06月15日 小组成员进行分工（曾碧霄：前端部分 王康：后端部分）
   4. 2018年06月16日-06月19日 完成软件的顶层数据流图、分层数据流图以及IPO图、系统结构图的设计（针对前端版块）
   5. 2018年06月21日 通过对比后端数据库表设计前端收集信息表的格式、完成前后端数据库对接
   6. 2018年06月22日 收集用户填写的信息数据发送给后端
   7. 2018年06月27日 完成对新建打卡和今日报表的表单设计，并收集用户数据发送给后端处理，同时接收后端返回的数据，将详细情况展示给用户界面
   8. 2018年06月28日 小组成员共同对系统进行测试
2. 报告完成日期

2018年06月30日