南 京 理 工 大 学

软件设计报告

|  |  |  |  |
| --- | --- | --- | --- |
| **学 号** | 914106840104 | **姓 名：** | 马倩 |
| **学院(系)** | 计算机科学与工程学院 | | |
| **班 级** | 9141062301 | | |
| **题 目** | 设计同时支持PC端及手机移动端访问的Web网站 | | |
| **指导教师** | 马 勇 | | |
| **课程** | 人机交互技术 | | |

2016年 1月

目 录

**[摘 要 1](#_Toc1836)**

[1.需求分析 2](#_Toc17869)

[1.1背景 2](#_Toc8622)

[1.2软件特点 2](#_Toc3026)

[1.3后续完善 2](#_Toc20994)

[2.概要设计 2](#_Toc14287)

[2.1网站及安卓app初始框架 2](#_Toc19138)

[2.2基本原理概述 3](#_Toc15498)

[3.详细设计 4](#_Toc16443)

[3.1流程图 4](#_Toc24374)

[3.1.1 LEvernote网页 4](#_Toc9359)

[3.1.2 LEvernote app 5](#_Toc19547)

[3.2设计思路与具体实现 6](#_Toc4445)

[3.2.1 LEvernote网页 6](#_Toc24914)

[3.2.2 LEvernote app 9](#_Toc22518)

[4.系统运行 11](#_Toc13876)

[4.1重要源代码 11](#_Toc13288)

[4.1.1产生验证码 11](#_Toc5295)

[4.1.2 登录 14](#_Toc15138)

[4.2实现结果截图 16](#_Toc10787)

[4.2.1主界面 16](#_Toc16387)

[4.2.2登录 17](#_Toc30698)

[4.2.3 登录后主界面 18](#_Toc15630)

[4.2.4注册 22](#_Toc23239)

[4.2.5登出 25](#_Toc24349)

[4.2.6找回密码 26](#_Toc2756)

[4.3界面分析 27](#_Toc10823)

[5.问题解决 27](#_Toc16136)

[5.1解决问题 27](#_Toc22228)

[5.1.1 EditText值比较 27](#_Toc15733)

[5.1.2背景图片 28](#_Toc12020)

[5.1.3MVC 28](#_Toc7173)

[5.2编程环境 28](#_Toc15126)

[5.3总结 28](#_Toc27203)

摘 要

本次人机交互大作业的课题是设计并实现同时支持PC端以及手机移动端访问的web网站，同时只要求实现登录相关的功能，内部具体功能可以不用实现，本人设计的是记录task的软件，名称为LEvernote，面向所有有需要的用户，此次实训程序设计主要分为两大部分简单GUI设计和程序功能设计。通过GUI设计，实现图形用户界面，方便用户操作。使用Java语言以及php语言编写。

通过设计和实现一个具有基本登录功能的简单记事本，提高我们对相关编程语言设计的掌握能力，理论联系实际进一步提高软件开发技术从而培养我们分析、解决问题的能力。

**关键字：**MVC xampp 活动activity PC端与手机移动端 LEvernote

软件设计报告

# 1.需求分析

## 1.1背景

随着计算机科学日渐成熟，任务记事本它已进入人类社会的各个领域并发挥着越来越重要的作用。仅从日常应用方面来说，一个文本编辑器只需一些简单实用的功能就够了。本程序设计就是依照这样一种使用需要设计了一个简单的记事本程序，面向所有有需要记录事项的用户。

## 1.2软件特点

软件名称为LEvernote，受启发于自己最近一直在使用的一种在线笔记本——印象笔记，该软件给我的生活带来了太多的便利，LEvernote致力于做成便利贴形式的小巧的记事本，用户登陆后方可进行添加与删除task。软件简单易懂，用户使用起来很容易，用户在完成任务的时候，不会有其他操作的干扰。网页端开发环境为atom编辑器、xampp配置数据库及搭建服务器、laravel框架，Android app开发环境为Android Studio集成环境。

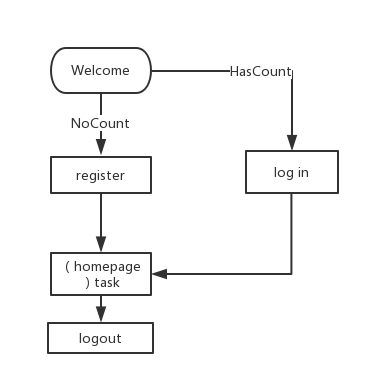
## 1.3后续完善

目前只完成了LEvernote的登录注册功能，其中网页端与数据库连接验证，而手机移动端只在后台验证，未与数据库连接，后续将善LEvernote的功能菜单，新增增加图片、音乐等功能，增强用户体验。

# 2.概要设计

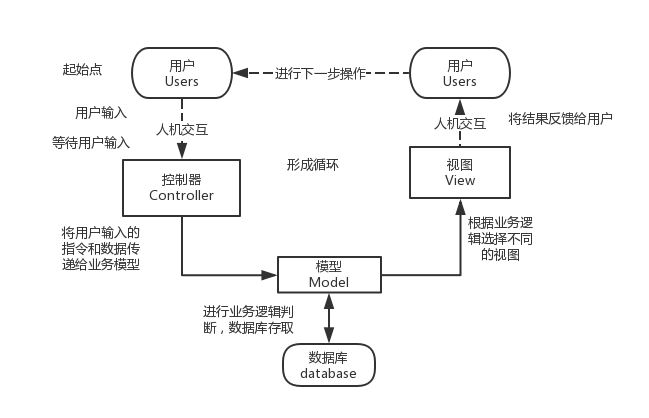
## 2.1网站及安卓app初始框架

软件初始指定要实现的功能为登录、注册，以及登录页面与注册页面，如下图所示：



## 2.2基本原理概述

网站及安卓app的架构均遵循MVC模式，



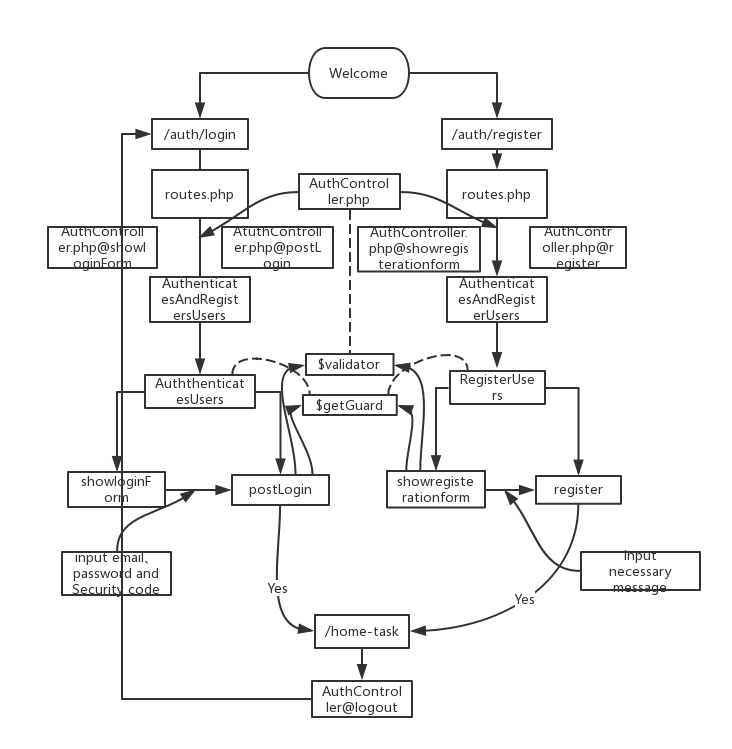
如上图所示，MVC，及模型、视图、控制器，模型表示企业数据和业务规则。在MVC的三个部件中，模型拥有最多的处理任务。例如它可能用象EJBs和ColdFusion Components这样的构件对象来处理数据库。被模型返回的数据是中立的，就是说模型与数据格式无关，这样一个模型能为多个视图提供数据。由于应用于模型的代码只需写一次就可以被多个视图重用，所以减少了代码的重复性。视图是用户看到并与之交互的界面。对老式的Web应用程序来说，视图就是由HTML元素组成的界面，在新式的Web应用程序中，HTML依旧在视图中扮演着重要的角色，但一些新的技术已层出不穷，它们包括Macromedia Flash和象XHTML，XML/XSL，WML等一些标识语言和Web services.如何处理应用程序的界面变得越来越有挑战性。MVC一个大的好处是它能为你的应用程序处理很多不同的视图。在视图中其实没有真正的处理发生，不管这些数据是联机存储的还是一个雇员列表，作为视图来讲，它只是作为一种输出数据并允许用户操纵的方式。控制器接受用户的输入并调用模型和视图去完成用户的需求。所以当单击Web页面中的超链接和发送HTML表单时，控制器本身不输出任何东西和做任何处理。它只是接收请求并决定调用哪个模型构件去处理请求，然后用确定用哪个视图来显示模型处理返回的数据。

# 3详细设计

## 3.1流程图

### 3.1.1 LEvernote网页

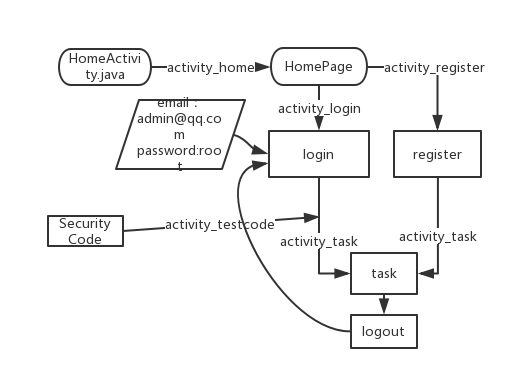
网页的基本框架如下图所示：



网页端的模型体现在task.php文件以及user.php文件中，视图由views目录下的.blade.php文件构成，组成呈现给用户的交互界面，其中app.blade.php为导航，是组成其他视图的必备；控制器位于controllers目录下，路由通过接收到的请求选择不同控制器中的方法并将结果反馈给用户，以达到人机交互的目的。用户登录前需要注册一个账户。

### 3.1.2 LEvernote app

Android系统是一个Activity 的 onCreate() 方法调用开始启动程序。一个回调方法 - 启动一个活动，以及其它回调方法，如销毁一个活动。基本框架如下图所示：

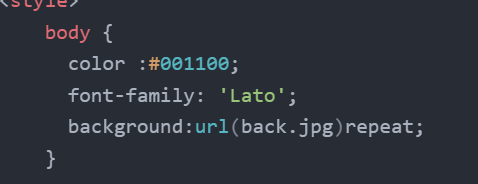


程序从HomeActivity的onCreate（）函数函数调用开始进入函数，模型为model文件夹下的数据库文件，在数据库中创建了user和task表，视图为layout文件夹下的.xml布局文件，home为欢迎界面，activity文件则为控制器类型，表示一个单一屏幕上的用户界面。目前登录的用户名为admin@qq.com，密码为root。

## 3.2设计思路与具体实现

### 3.2.1 LEvernote网页

1. 背景图片添加：



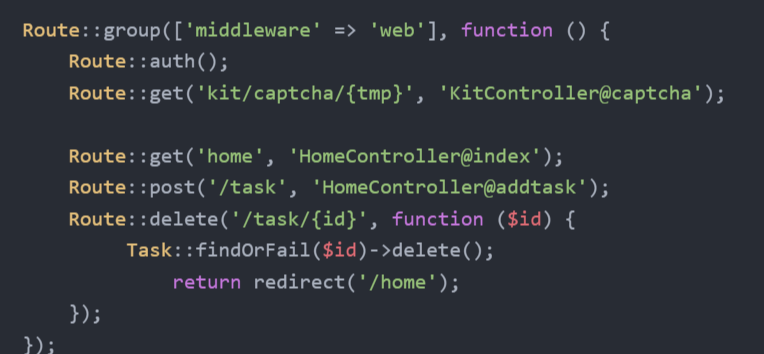
在body标签中加入background属性，值为背景图片地址，如果背景图片是存放在项目根目录下，地址只需填图片名字即可。

1. 验证码的添加：

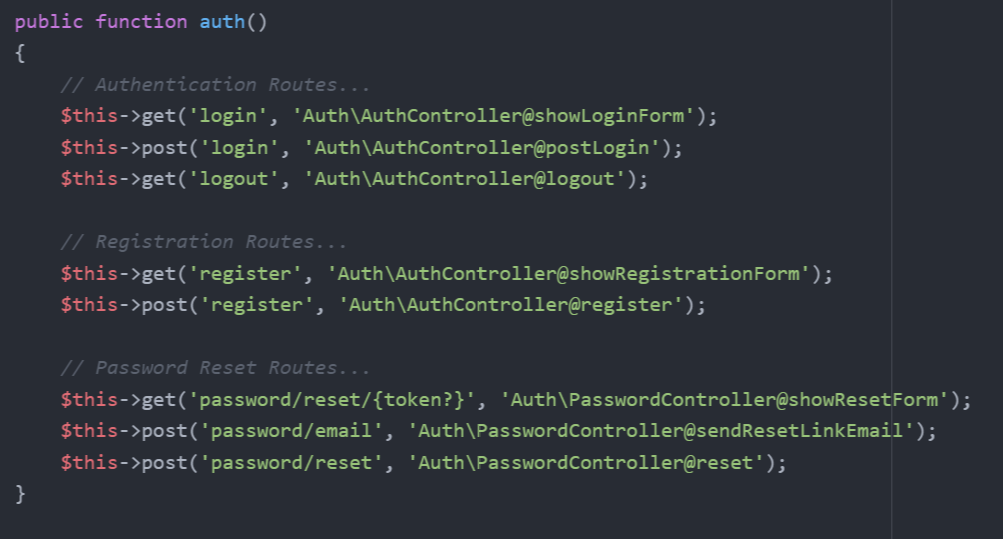


验证码的使用是在postLogin方法中加入if-else判断验证码输入是否正确，错误需要从新填写，该方法位于/vendor/laravel/framework/src/Illuminate\Foundation\Auth。

1. 路由：



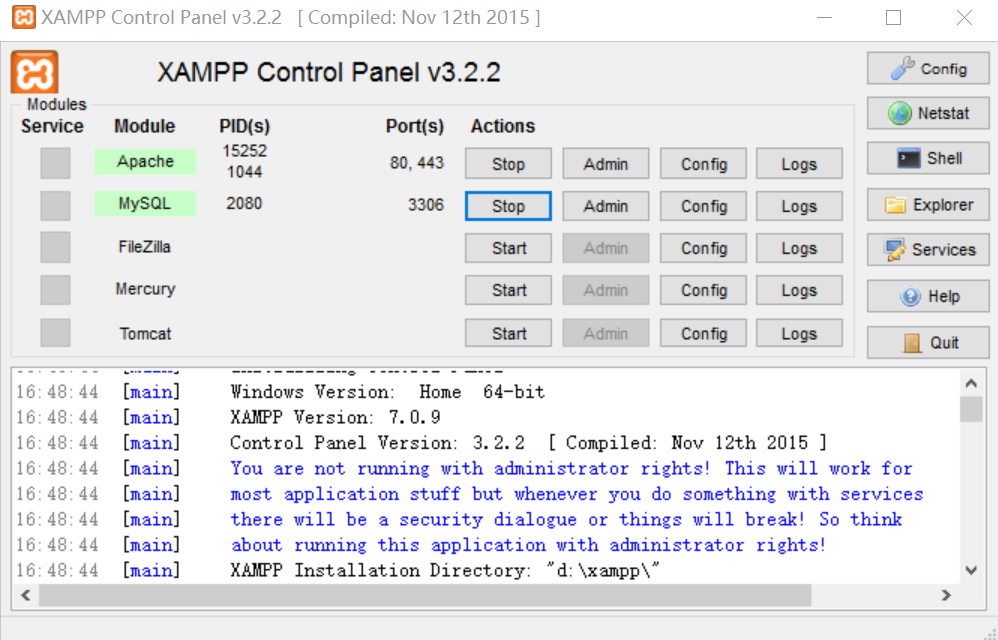
1. auth（）方法：



位于/vendor/laravel/framework/src/Illuminate\Routing目录下，调用该方法后路由中得到这些组件，并根据请求类型调用相应方法。

1. 服务器与数据库

xampp是php开发环境，是完全免费且易于安装的Apache发行版，其中包含MariaDB、PHP和Perl，其中数据库登录用户名和密码均是laravel，数据库名称为laravel，如下为主界面，大作业文件夹laravel.sql下为导出的PC端的数据库文件：



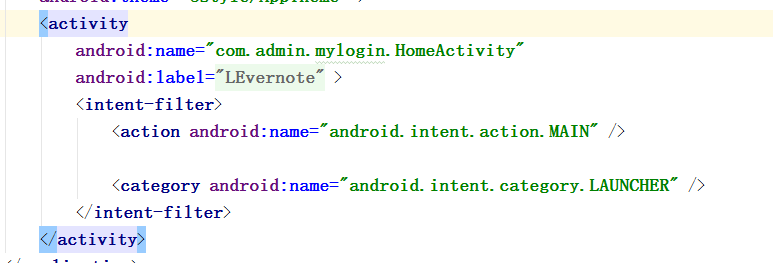
数据库如下，其中有4个表，migrations为数据库自带，用来记录数据库迁移信息的，由127.0.0.1/phpmyadmin访问得到，数据库登录账户名及密码均为laravel，需要在项目的.env文件中配置相关数据信息，才能利用命令行进行数据库迁移操作。

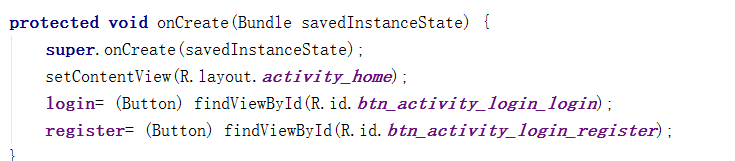




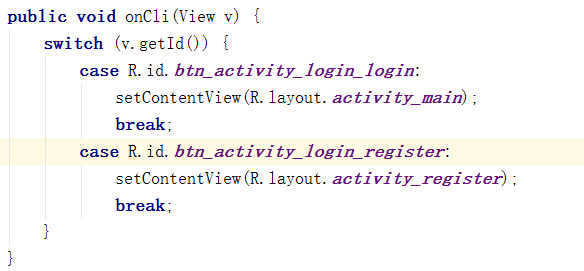
### 3.2.2 LEvernote app

1. 程序起始活动的设置：

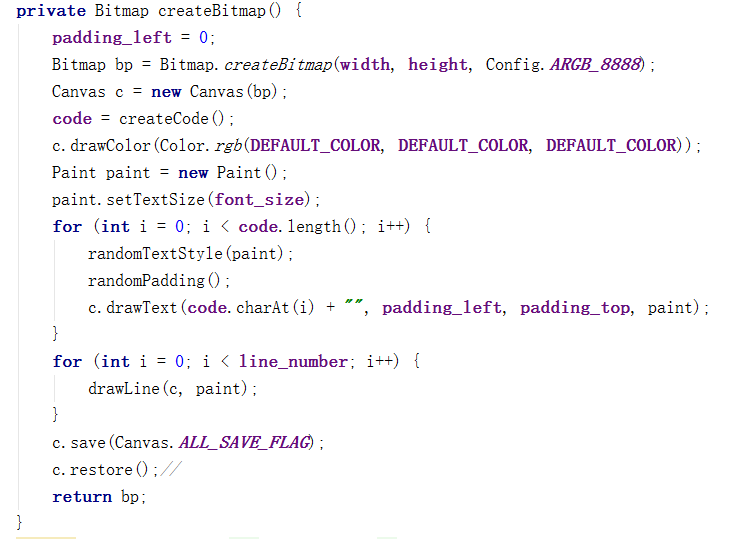




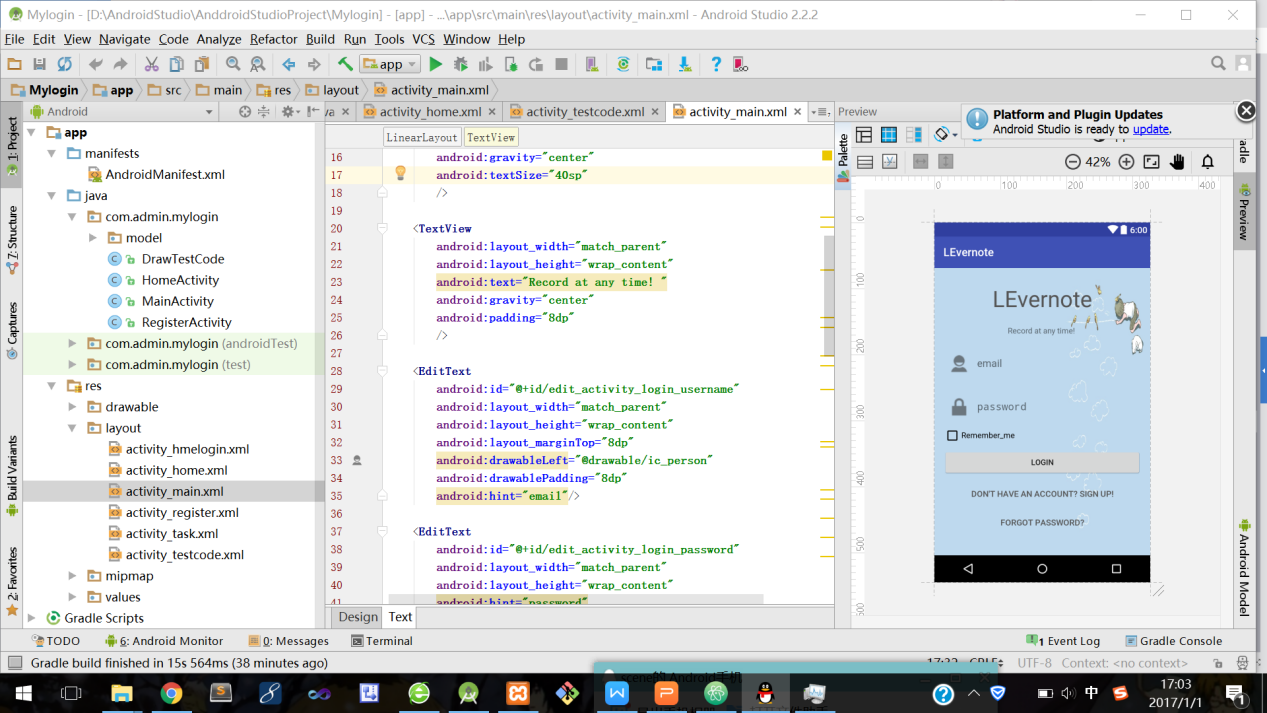
1. 判断按钮ID，选择要跳转的页面：



1. 实现图片验证码：



1. 开发环境：



1. 背景图片添加

只需要将图片放在drawable目录下，并在每个页面的.xml文件中加入如下代码：

53LC4_2HU}F$Q(@~_C[U_]1

即可实现以back为名的图片作为背景图。

# 4.系统运行

## 4.1重要源代码

### 4.1.1产生验证码

//手机移动端，DrawTestCode.java

**package** com.admin.mylogin;  
**import** java.util.Random;  
**import** android.graphics.Bitmap;  
**import** android.graphics.Bitmap.Config;  
**import** android.graphics.Canvas;  
**import** android.graphics.Color;  
**import** android.graphics.Paint;  
**public class** DrawTestCode {  
 **private static final char**[] ***CHARS*** = { **'0'**, **'1'**, **'2'**, **'3'**, **'4'**, **'5'**, **'6'**,  
 **'7'**, **'8'**, **'9'**, **'a'**, **'b'**, **'c'**, **'d'**, **'e'**, **'f'**, **'g'**, **'h'**, **'i'**, **'j'**,  
 **'k'**, **'l'**, **'m'**, **'n'**, **'o'**, **'p'**, **'q'**, **'r'**, **'s'**, **'t'**, **'u'**, **'v'**, **'w'**,  
 **'x'**, **'y'**, **'z'**, **'A'**, **'B'**, **'C'**, **'D'**, **'E'**, **'F'**, **'G'**, **'H'**, **'I'**, **'J'**,  
 **'K'**, **'L'**, **'M'**, **'N'**, **'O'**, **'P'**, **'Q'**, **'R'**, **'S'**, **'T'**, **'U'**, **'V'**, **'W'**,  
 **'X'**, **'Y'**, **'Z'** };  
 **private static** DrawTestCode *bpUtil*;  
 **private** DrawTestCode() {  
 };  
 **public static** DrawTestCode getInstance() {  
 **if** (*bpUtil* == **null**)  
 *bpUtil* = **new** DrawTestCode();  
 **return** *bpUtil*;  
 }  
 **private static final int *DEFAULT\_CODE\_LENGTH*** = 4;*// 验证码的长度 这里是4位* **private static final int *DEFAULT\_FONT\_SIZE*** = 60;*// 字体大小* **private static final int *DEFAULT\_LINE\_NUMBER*** = 3;*// 多少条干扰线* **private static final int *BASE\_PADDING\_LEFT*** = 20; *// 左边距* **private static final int *RANGE\_PADDING\_LEFT*** = 35;*// 左边距范围值* **private static final int *BASE\_PADDING\_TOP*** = 42;*// 上边距* **private static final int *RANGE\_PADDING\_TOP*** = 15;*// 上边距范围值* **private static final int *DEFAULT\_WIDTH*** = 200;*// 默认宽度.图片的总宽* **private static final int *DEFAULT\_HEIGHT*** = 70;*// 默认高度.图片的总高* **private final int DEFAULT\_COLOR** = 0xdf;*// 默认背景颜色值* **private int width** = ***DEFAULT\_WIDTH***;  
 **private int height** = ***DEFAULT\_HEIGHT***;  
 **private int base\_padding\_left** = ***BASE\_PADDING\_LEFT***;  
 **private int range\_padding\_left** = ***RANGE\_PADDING\_LEFT***;  
 **private int base\_padding\_top** = ***BASE\_PADDING\_TOP***;  
 **private int range\_padding\_top** = ***RANGE\_PADDING\_TOP***;  
 **private int codeLength** = ***DEFAULT\_CODE\_LENGTH***;  
 **private int line\_number** = ***DEFAULT\_LINE\_NUMBER***;  
 **private int font\_size** = ***DEFAULT\_FONT\_SIZE***;  
 **private** String **code**;*// 验证码* **private int padding\_left**, **padding\_top**;  
 **private** Random **random** = **new** Random();  
 **private** Bitmap createBitmap() {  
 **padding\_left** = 0;  
 Bitmap bp = Bitmap.*createBitmap*(**width**, **height**, Config.***ARGB\_8888***);  
 Canvas c = **new** Canvas(bp);  
 **code** = createCode();  
 c.drawColor(Color.*rgb*(**DEFAULT\_COLOR**, **DEFAULT\_COLOR**, **DEFAULT\_COLOR**));  
 Paint paint = **new** Paint();  
 paint.setTextSize(**font\_size**);  
 **for** (**int** i = 0; i < **code**.length(); i++) {  
 randomTextStyle(paint);  
 randomPadding();  
 c.drawText(**code**.charAt(i) + **""**, **padding\_left**, **padding\_top**, paint);  
 }  
 **for** (**int** i = 0; i < **line\_number**; i++) {  
 drawLine(c, paint);  
 }  
 c.save(Canvas.***ALL\_SAVE\_FLAG***);  
 c.restore();*//* **return** bp;  
 }  
 **public** String getCode() {  
 **return code**;  
 }  
 **public** Bitmap getBitmap() {  
 **return** createBitmap();  
 }  
 **private** String createCode() {  
 StringBuilder buffer = **new** StringBuilder();  
 **for** (**int** i = 0; i < **codeLength**; i++) {  
 buffer.append(***CHARS***[**random**.nextInt(***CHARS***.**length**)]);  
 }  
 **return** buffer.toString();  
 }  
 **private void** drawLine(Canvas canvas, Paint paint) {  
 **int** color = randomColor();  
 **int** startX = **random**.nextInt(**width**);  
 **int** startY = **random**.nextInt(**height**);  
 **int** stopX = **random**.nextInt(**width**);  
 **int** stopY = **random**.nextInt(**height**);  
 paint.setStrokeWidth(1);  
 paint.setColor(color);  
 canvas.drawLine(startX, startY, stopX, stopY, paint);  
 }  
 **private int** randomColor() {  
 **return** randomColor(1);  
 }  
 **private int** randomColor(**int** rate) {  
 **int** red = **random**.nextInt(256) / rate;  
 **int** green = **random**.nextInt(256) / rate;  
 **int** blue = **random**.nextInt(256) / rate;  
 **return** Color.*rgb*(red, green, blue);  
 }  
 **private void** randomTextStyle(Paint paint) {  
 **int** color = randomColor();  
 paint.setColor(color);  
 paint.setFakeBoldText(**random**.nextBoolean()); *// true为粗体，false为非粗体* **float** skewX = **random**.nextInt(11) / 10;  
 skewX = **random**.nextBoolean() ? skewX : -skewX;  
 paint.setTextSkewX(skewX); *// float类型参数，负数表示右斜，整数左斜* }  
 **private void** randomPadding() {  
 **padding\_left** += **base\_padding\_left** + **random**.nextInt(**range\_padding\_left**);  
 **padding\_top** = **base\_padding\_top** + **random**.nextInt(**range\_padding\_top**);  
 }  
}

使用的时候通过调用函数即可，例如：

**vc\_image** = (ImageView) findViewById(R.id.***vc\_image***);  
**vc\_image**.setImageBitmap(DrawTestCode.*getInstance*().getBitmap());  
**vc\_code** = (EditText) findViewById(R.id.***vc\_code***);  
**getCode** = DrawTestCode.*getInstance*().getCode(); *// 获取显示的验证码***update** = (Button) findViewById(R.id.***update***); //刷新验证码  
**update**.setOnClickListener(**this**);

### 4.1.2 登录

PC端，截取部分核心代码：

public function postLogin(Request $request)

{

$userInput = $request->get('captcha');

if (Session::get('milkcaptcha') == $userInput) {

$this->validate($request, [

$this->loginUsername() => 'required', 'password' => 'required',

]);

$throttles = $this->isUsingThrottlesLoginsTrait();

if ($throttles && $this->hasTooManyLoginAttempts($request)) {

return $this->sendLockoutResponse($request);

}

$credentials = $this->getCredentials($request);

if (Auth::attempt($credentials, $request->has('remember'))) {

return $this->handleUserWasAuthenticated($request, $throttles);

}

if ($throttles) {

$this->incrementLoginAttempts($request);

}

return redirect($this->loginPath())

->withInput($request->only($this->loginUsername(), 'remember'))

->withErrors([

$this->loginUsername() => $this->getFailedLoginMessage(),

]);

} else { //用户输入验证码错误

echo '您输入验证码错误，请重新输入！';

return view('auth.login');}

//return $this->login($request);

}

4.1.3 HomeController控制器

<?php

namespace App\Http\Controllers;

use App\Http\Requests;

use Illuminate\Http\Request;

use App\Task;

use Validator;

use App\Http\Controllers\Controller;

class HomeController extends Controller

{

/\*\*

\* Create a new controller instance.

\*

\* @return void

\*/

public function \_\_construct()

{

$this->middleware('auth');

}

/\*\*

\* Show the application dashboard.

\*

\* @return \Illuminate\Http\Response

\*/

public function index()

{

return view('tasks', [

'tasks' => Task::orderBy('created\_at', 'asc')->get()

]);

//return view('tasks');

}

public function addtask(Request $request) //实现addtask功能

{

$validator = Validator::make($request->all(), [

'name' => 'required|max:255',

]);

if ($validator->fails()) {

return redirect('/home')

->withInput()

->withErrors($validator);

}

$task = new Task;

$task->name = $request->name;

$task->save();

return redirect('/home'); //重定向

}

}

## 4.2实现结果截图

### 4.2.1主界面

1)PC端

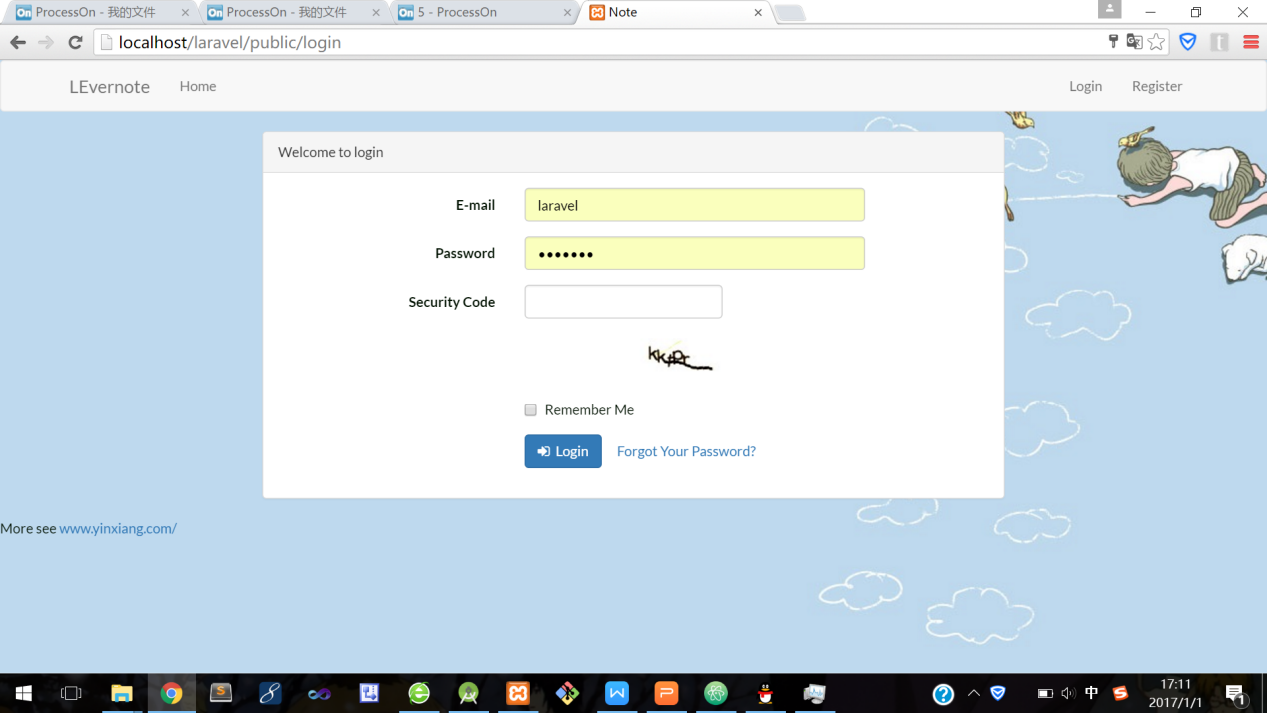


2)手机移动端：



### 4.2.2登录

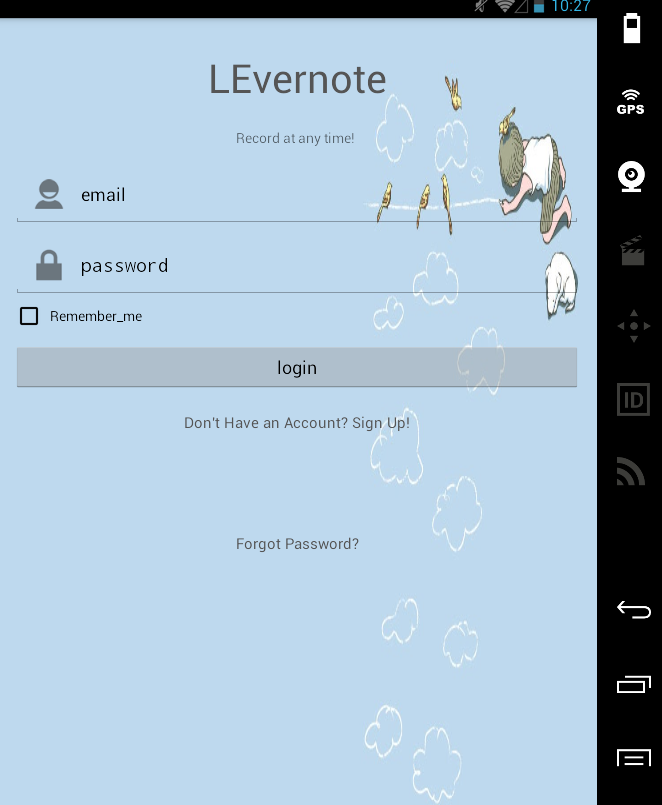
1. PC端



其中

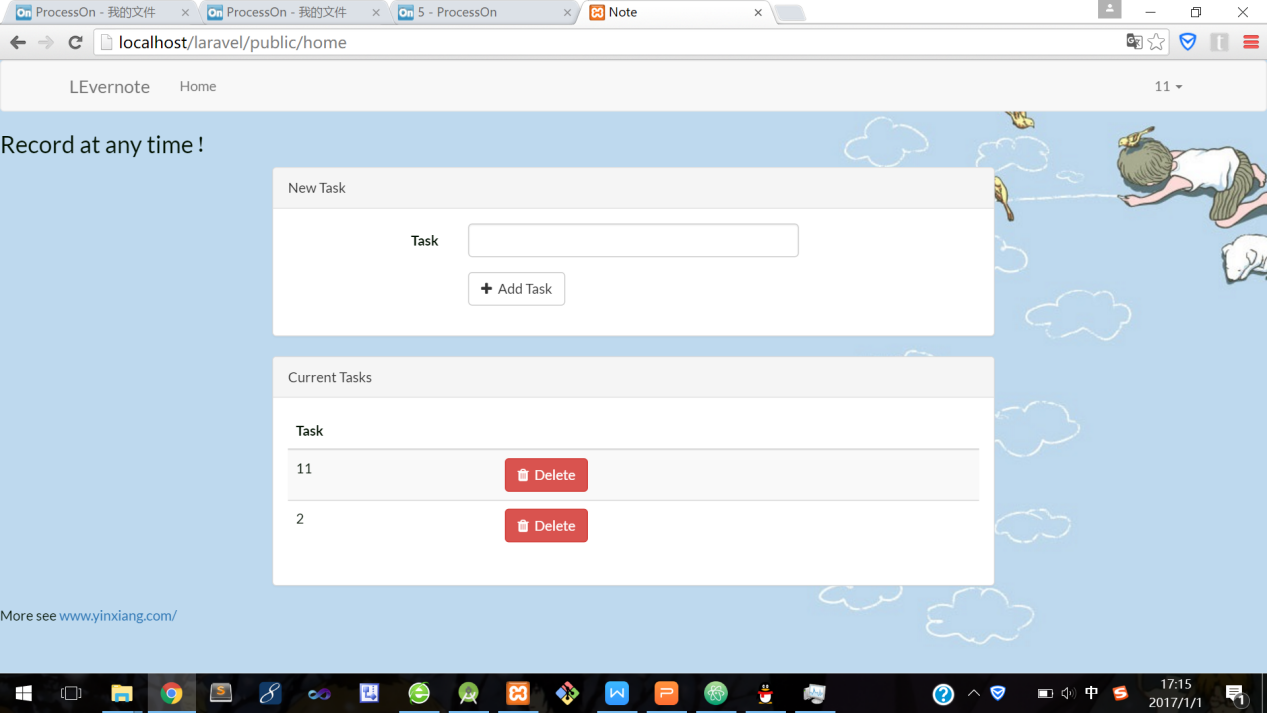
为印象笔记官网链接。

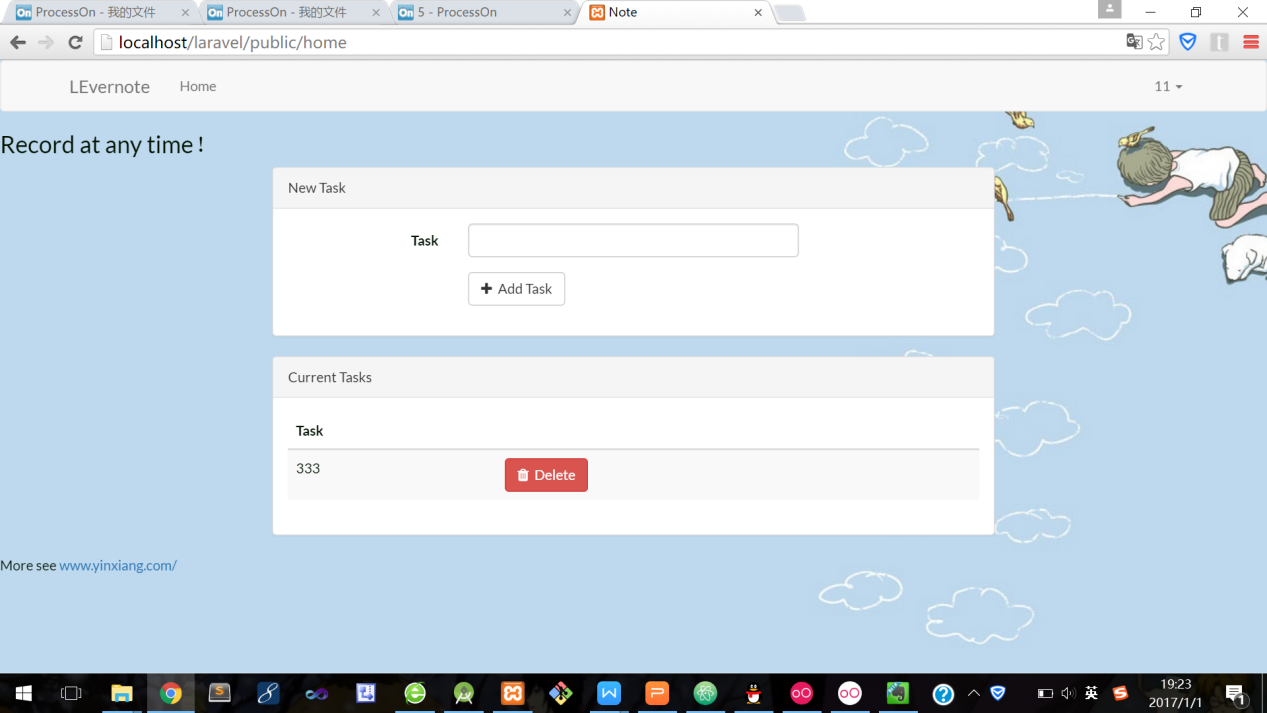
1. 手机移动端



### 4.2.3 登录后主界面

1. PC端





1. 手机移动端

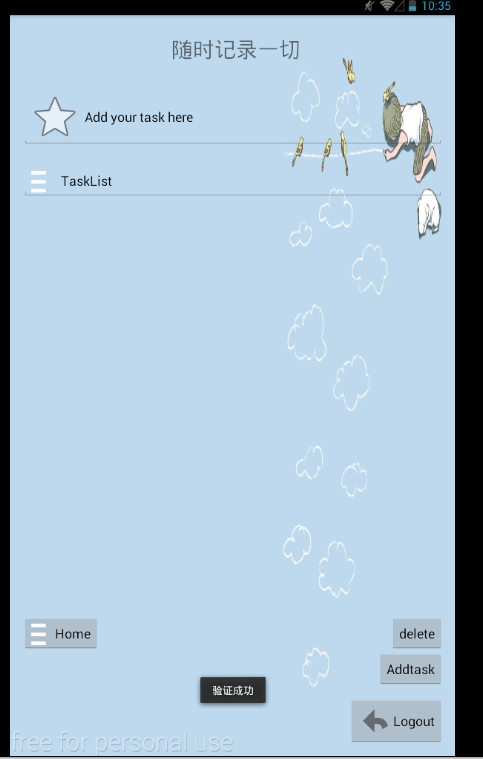
输入账号admin@qq.com，密码root，就会弹出验证码窗口，如下：





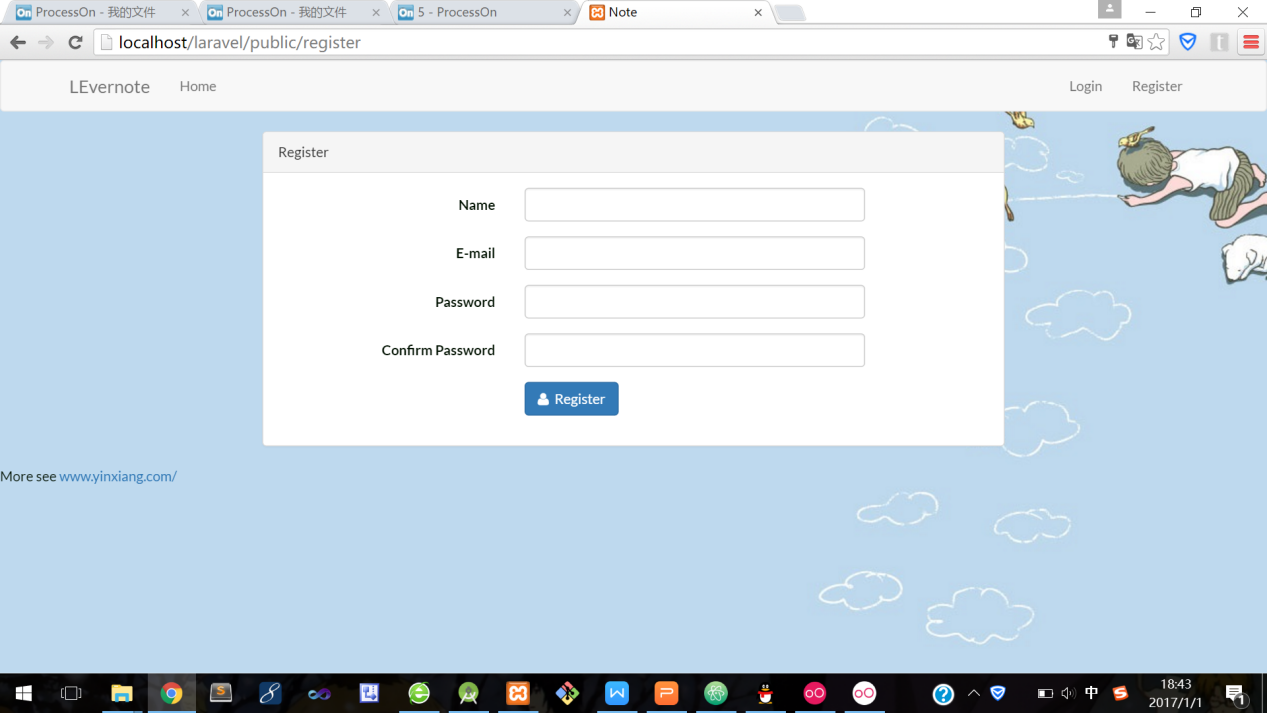
可以选择刷新验证码，手机移动端是点击刷新按钮进行刷新验证码，网页端为点击验证码图片，又或者直接填写当前验证码，输入错误验证码，会提示错误，重新输入，

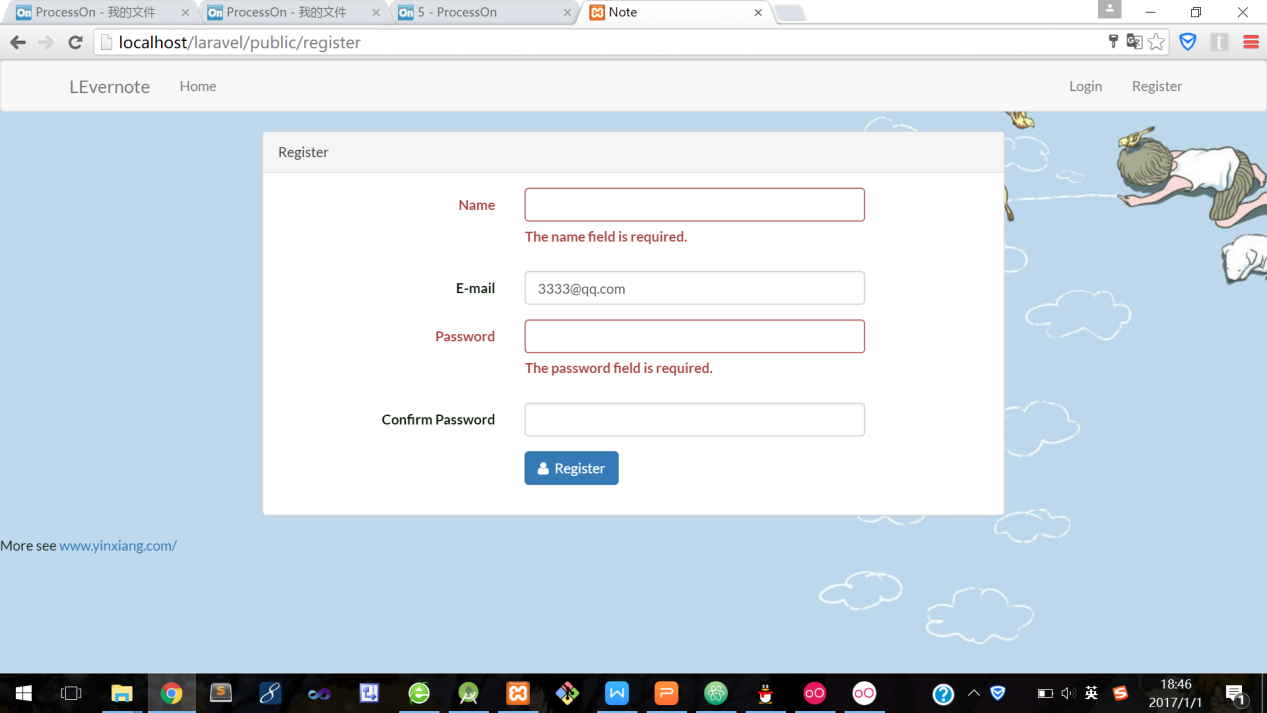
正确输入则跳转到task页面：

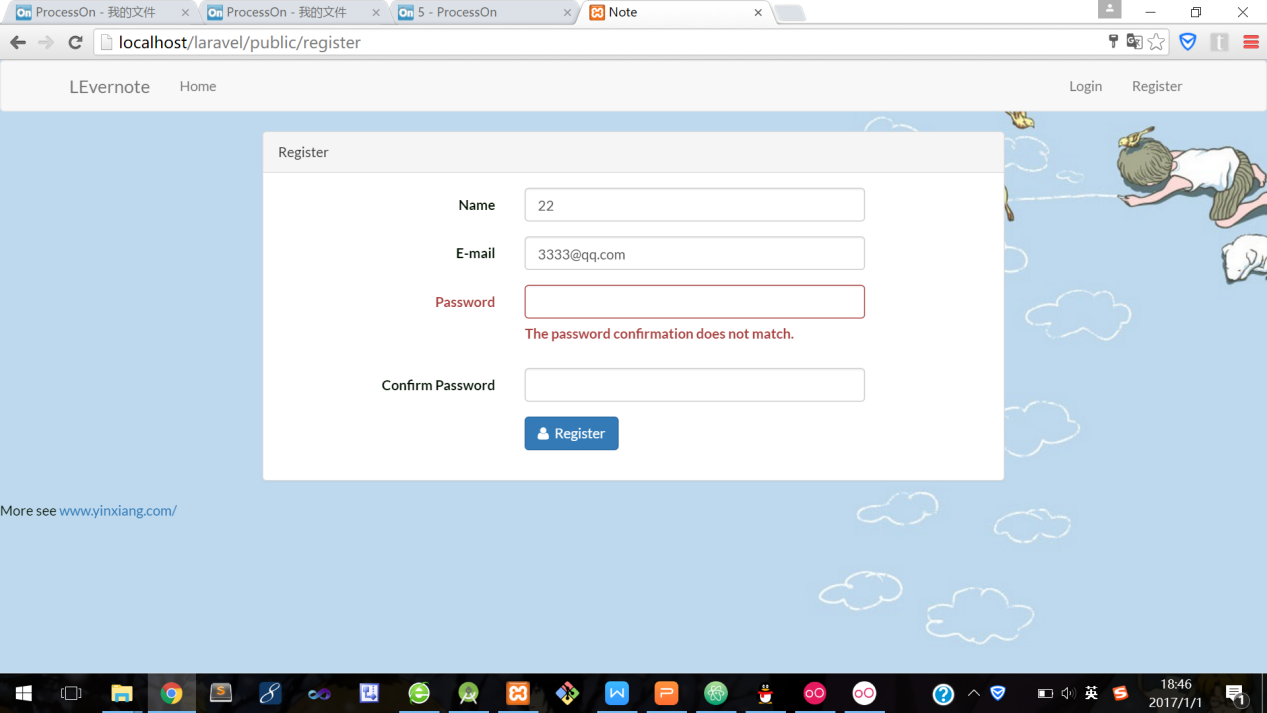


### 4.2.4注册

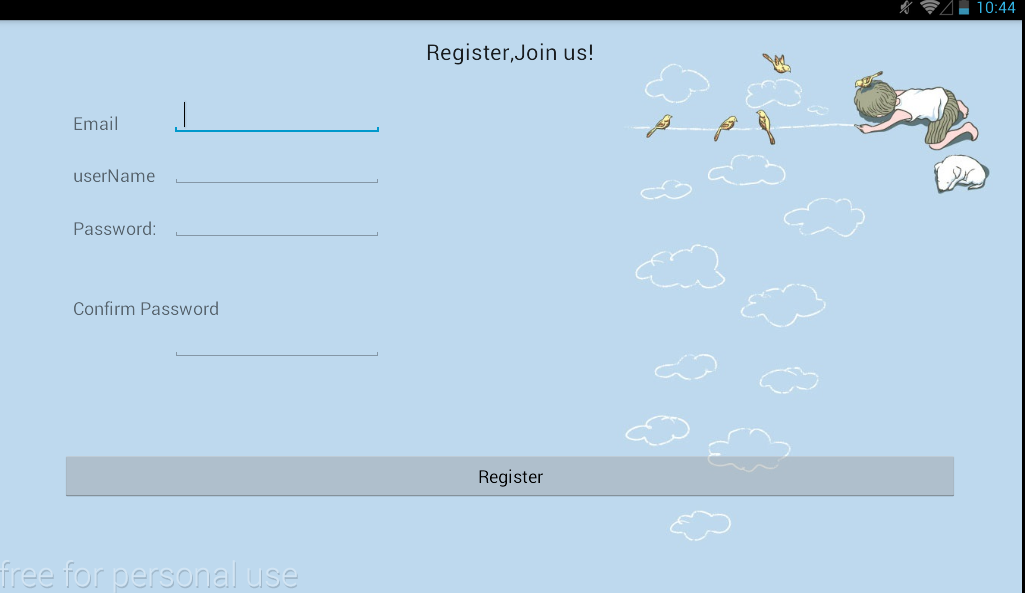
1. PC端

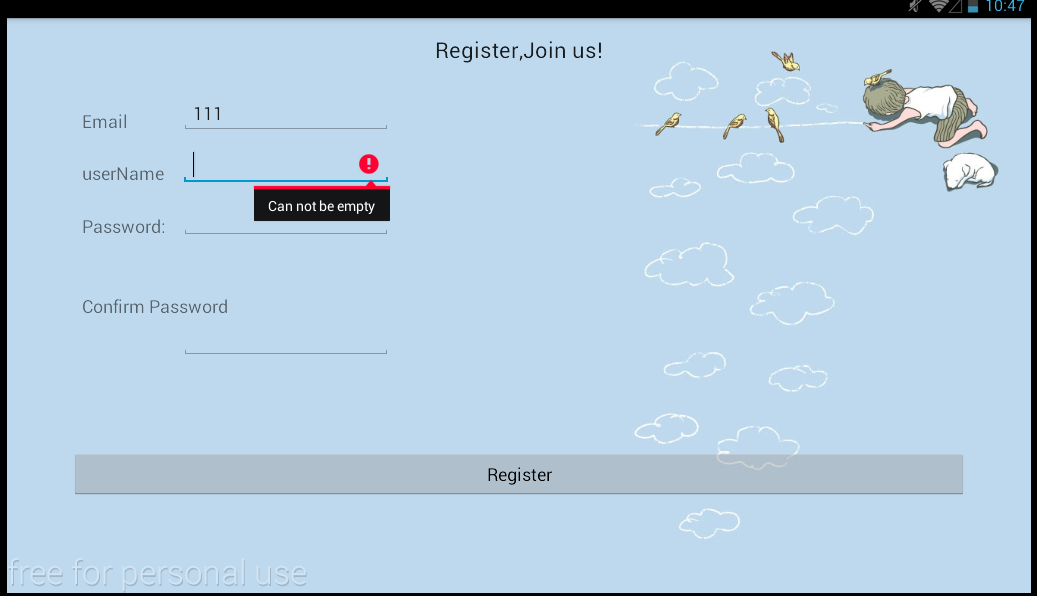


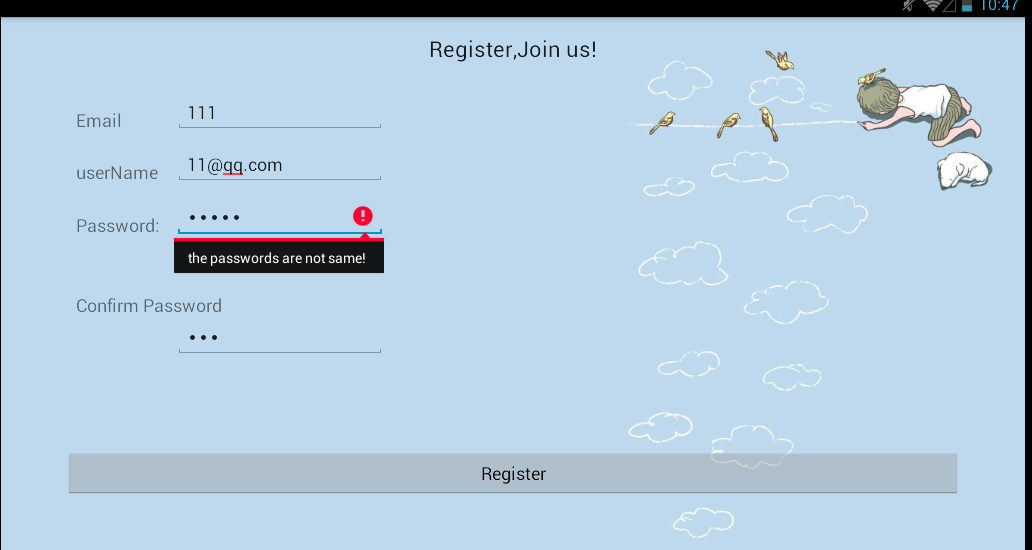




1. 手机移动端





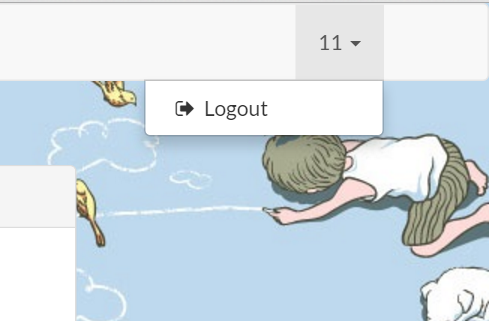


注册成功后后直接跳转到task页面。

### 4.2.5登出

1. PC端

在登陆后的task主页中，用户名下拉列表可以实现登出，如下，登出后后返回到欢迎主页。



1. 手机移动端

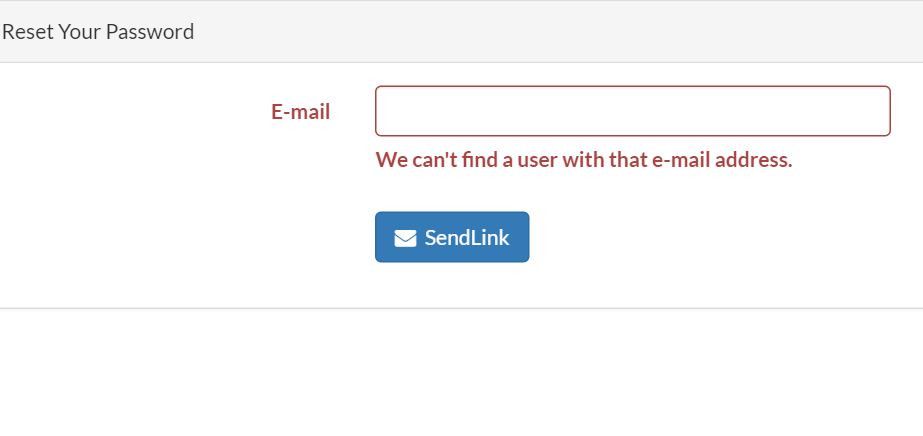
在task页面点击home按钮，可实现登出，



### 4.2.6找回密码

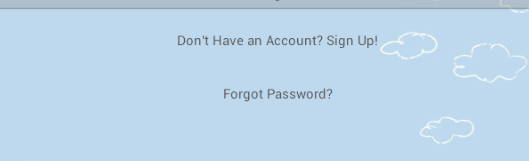
1. PC端

网页端重置密码是必须输入已经在数据库中存在的用户名：



1. 手机移动端

手机移动端暂未设置找回密码功能，仅仅设置了Forgot Password按钮，按钮对应函数方法尚未实现。



## 4.3界面分析

各个功能界面基本符合人机交互原则，而且用起来简单易懂，没有给用户带来难懂的操作界面。

# 5问题解决

## 5.1解决问题

### 5.1.1 EditText值比较

EditText类型值是不能直接进行比较大小的，需要通过调用toString（）方法和equals（）方法，如下：

password.getText().toString().equals(confirmpassword.getText().toString())，比较输入框中输入的两个密码是否一致，一致才能注册成功。

### 5.1.2背景图片

PC端背景图片应放在xampp/htdocs/laravel/pubic目录下，此目录为laravel项目的根目录，其中htdocs目录为xampp的根目录。

### 5.1.3MVC

Model、view、controller直接可以使用命令行创建相应文件

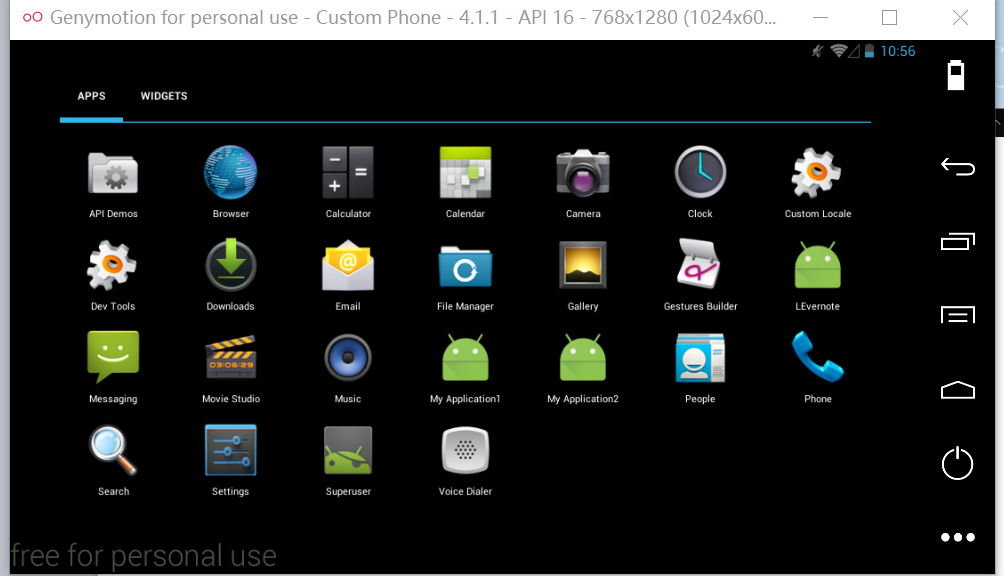
php make artisan:model model1；

php make artisan:view view1；

php make artisan:controller con1；

## 5.2编程环境

网页使用atom编辑器，php语言，xampp搭建服务器与数据库；手机移动端使用Android Studio 2.2.2版本，语言为java，使用Android模拟器进行运行测试。下图为安卓模拟器：



## 5.3总结

通过本次课程的大作业，主动的去学习了很多平时上课没有接触到的东西，例如laravel框架、安卓模拟器、xampp，Android Studio能实现自动保存代码，学习这些为自己以后其它课程的学习奠定了基础，以后也会自己去慢慢完善网页及手机移动端的功能，例如增加输入多次错误密码后禁止用户继续输入密码、增加找回密码功能、增加功能菜单等；学习laravel是从它的官方例子-task入手的，所以在完成用户登录界面后加入了task功能，并且也已经实现，只是功能只有简单的增加task和删除task，有待完善；学习不是一件容易的事，需要慢慢稳扎稳打。