

# 建立博客网站

---

——可行性分析与需求分析

## 1. 引言

### 1. 撰写报告的目的

撰写此报告的目的在于，分析建立个人网站的需求，以及个人能力之上可以实现的功能。

### 2. 项目背景

个人网站包括个人博客系统，以及个功能模块。目前已经有很多个人博客系统的模板可供参看。

## 2. 可行性分析

### 1. 基本需求

主要分为：个人博客管理、系统管理以及主要功能实现三个方面。

1. 个人博客管理需要实现：注册登录，个人资料管理，浏览，评论留言，撰写与发表文章，上传下载素材以及附件。
2. 系统管理：包括用户管理，博客分类与归类，公告管理，评论管理，链接以及系统维护。
3. 功能模块，目前准备实现手写数字识别。用户在网站页面用鼠标画一个数字，然后系统进行识别。这里需要是现实的是：一个28\*28（像素点）的手写区域、一个已经训练好的识别手写数字的CNN网络。并要求此功能能够被重复调用。
4. 附加功能（暂未打算实现）：实现一个交易系统。按指令驱动来交易。提供一种证券（FtCoin），允许50人左右的账户。

### 2. 开发需求

需要一个域名，需要MySQL数据库。开发方面，需要用JS、HTML、CSS以及Node.js来写前端。然后需要用Django来实现后端。

## 3. 项目功能需求

1. 本网站需要实现个人博客的主要功能，需要允许账户的登录与登出。账户分为两类，一类是管理员账户，另一类是普通的注册用户。具有网站的管理权限的只有管理员。可以自定义网站的系统界面，管理数据。这里的系统界面包括，标签栏，归档栏，图片，网页端markdown编辑器，访客流量显示，点赞显示。用户可以做到的是发表博客，评论，使用主站点的功能模块。
2. 交易系统。因为会存在管理员，就无法保障交易内幕，所以采取分布式记账的方式，所有的交易建立在链上，每一笔交易都是链上的一个节点，订单的内容通过函数来实现，系统撮合交易后，向所有的用户广播。用户可以提交自己的订单，并进行一系列的交易。

### 3. 设计方法

网站的标签栏会提供一个注册和登录的标签。对于注册和登录界面，需要用Node.js去写一个专门的界面。用户注册和登录之后会自动转跳到网站的主界面。主界面上回按时间倒叙显示所有人的文章以及使用手写数字、交易的动态。目前打算是将这三类不同的信息放在不同子站点里面。管理员登录后可以进行管理操作。

### 4. 功能性分析

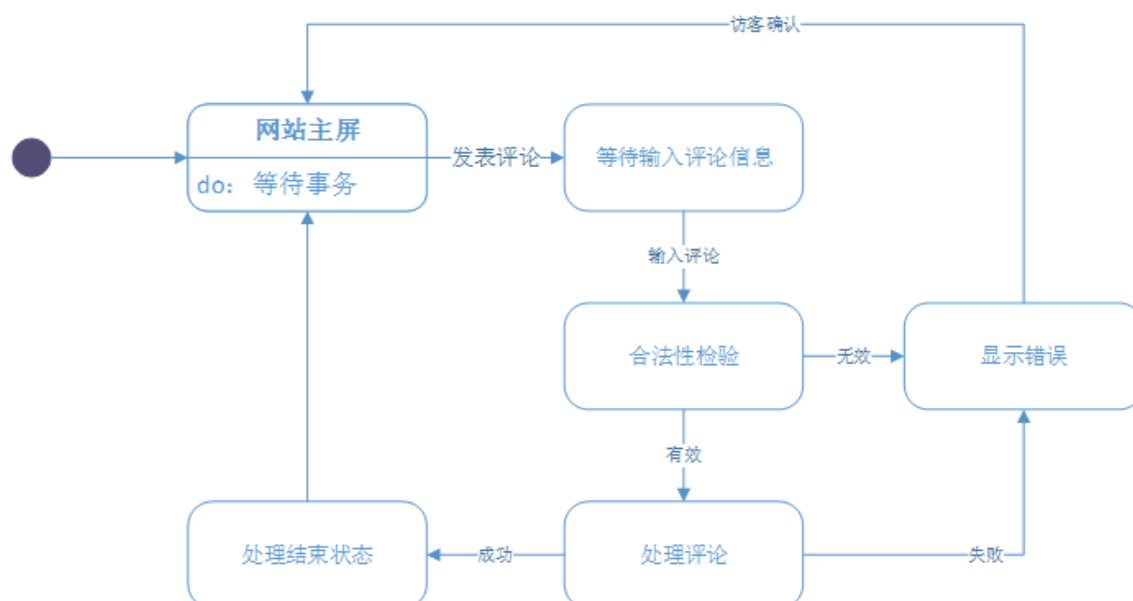
1. 网页浏览
2. 所有用户的登录登出
3. 博客的发表，修改，评论；文件的上传、显示、修改、下载
4. 用户在网站上的留言，以及对留言的回复（评论）
5. 管理员管理用户
6. 管理员设置网站的主要界面
7. 插件管理：包括添加、修改、删除插件
8. 在28\*28像素点内画一个数字的功能模块，识别数字的模块
9. 交易撮合，交易广播

## 4. 开发环境以及软硬件的需求

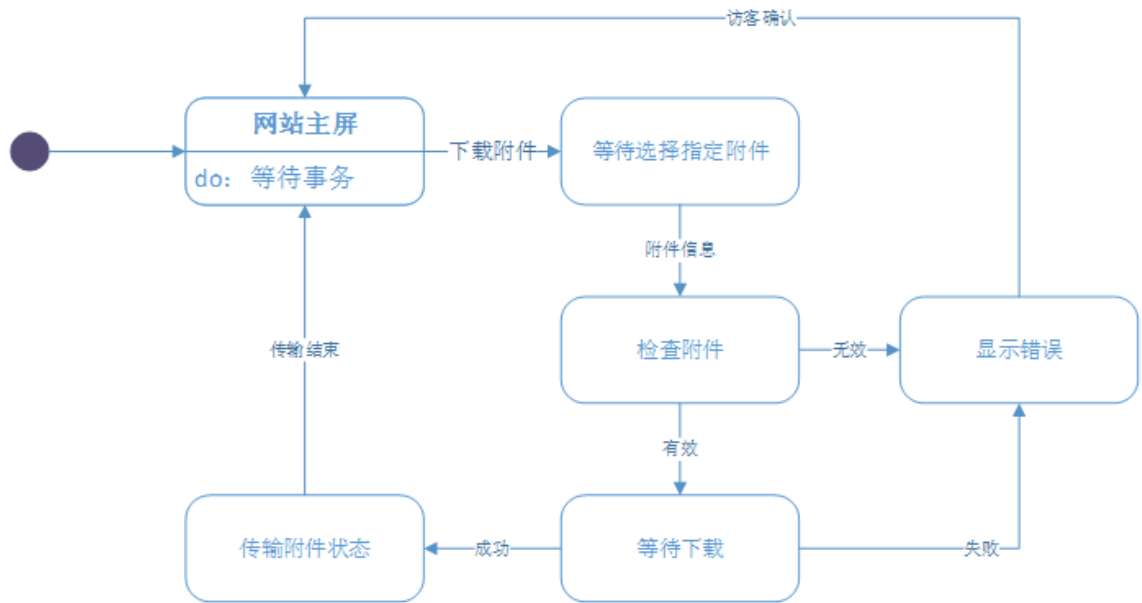
1. 开发工具：pycharm、webstorm
2. 涉及到的编程语言：javascript、django、html、css、bootstrap、sql、node.js
3. web服务器：macOS；环境为APache+MySQL+FTP配置环境

## 5. 系统行为图例

### 1. 发表评论



### 2. 下载附件



## 6. 数据字典描述

### 1. 顶层数据字典

实体名称	数据流	简述
访客	浏览、评论、下载	访问文章，添加博客和评论
管理员	管理、浏览、下载	对于网站和用户的管理操作

### 2. 一层数据字典

系统各功能名称	对应实体	数据流	简述
阅读博客	用户	浏览信息	访客浏览网站，可阅读符合条件的文章，并记录下访客的浏览信息，如IP地址、文章浏览量等信息
发表评论	用户	评论信息	访客浏览文章、图片、视频媒体时，可对此发表评论，网站记录评论信息，是否即时发表由站长开关评论审核而定
下载	用户、管理员	下载信息	访客、站长均可以下载文章附件，附件形式多样，如doc、ppt、mp3、mp4等，网站记录下载信息，访客、站长获得对应附件

3. 二级数据字典

加工名	输入数据流	输出数据流	加工逻辑
内容合法性检查	访客评论	有效评论、无效评论	有效评论符合评论规则，评论规则规定某些关键词不能够出现，否则为无效评论
发表评论	用户	评论信息	访客浏览文章、图片、视频媒体时，可对此发表评论，网站记录评论信息，是否即时发表由站长开关评论审核而定
写入评论	有效评论	评论信息	有效评论写入评论信息表，记录相应文章ID与评论内容、时间等相关信息，是否即时发表由站长开关评论审核决定，发表后评论即可写入文章评论表

7. 数据库表

1. 管理员信息表

属性名	数据类型	约束条件	注释
登录名	Varchar(10)	Primary key, not null	
密码	Varchar(6)	Not null	
昵称	Varchar(20)	Not null	
头像	Nvarchar(MAX)	Not null	头像图片地址

2.

网站信息表

属性名	数据类型	约束条件	注释
序号	Int	Primary key, not null	
站名	Varchar(20)	Not null	
URL	Nvarchar(MAX)/td>		
数据时间	Date	Not null	网站系统最新的数据更新时间
模板地址	Nvarchar(MAX)	Not null	本地存储地址
插件地址	Nvarchar(MAX)		本地存储地址
附件地址	Nvarchar(MAX)		本地存储地址

## 8 框架以及图示

网站的框架

- Templates
  - UUBlog
    - Media 用户上传的数据
      - avatar
      - data
    - static
      - images 公共图片
      - scripts js引用
      - skins
        - default 各个博客模板对应的资源
      - ueditor
    - Templates
    - modeules 存放各个博客模板的所有模块
      - pub
        - config 博客设置目录
        - profile 用户信息设置
          - articles 文章
        - skins 博客的皮肤
          - default
          - temp1
          - tmep2

网站的首先是前端设计，在设计好前端之后，后端实现具体的功能。  
网页设计模型如下：



{\手写数字}

{\识别结果}

{\其他人的识别结果}  
{\统计展示}

- 识别手写数字
  - 在数据库里面保存MINIT数据库
  - 在本地训练好一张CNN网络
    - I=用户输入数字
    - I.转化 转化为28\*28px图像，并保存二进制文件
    - P=net.reco(I) 调用训练好的网络识别手写数字，返回识别的具体数字
    - print(P) 在“{\识别结果}”里面展示识别的结果
    - Bool t=judege(P) 用户判断返回的结果是否正确，正确t返回1，错误t返回0
      - 如果t=0：提示用户确认判断错误，确认后用户输入其本来认为正确的数字K
    - 将 (I, P/K) 导入新的数据库
    - 对网络重新训练
  - 结束
- 用户界面
  - 注册
    - 用户操作
      - 用户在提示下输入
        - 用户名
        - 密码
        - 确认密码
        - 验证码
      - 用户确认后提交
      - 系统分发ID：为数据库主键
    - 系统操作
      - 将新的数据存入mysql数据库中
      - 维护数据库
  - 登录
    - 用户操作
      - 输入账户
      - 输入密码
      - 确认后登录
    - 系统操作
      - 在数据库里面搜索用户名（主键）
      - 匹配密码
      - 正确后响应前端
- 博客系统
  - 录入系统
    - 调用markdown编辑器
    - 提供编辑预览



- 确认后保存
- 展示系统
  - 保存用户的文章到数据库系统
  - 在用户提交文本后，调用md文件
  - 调用md文件，在browser中展示
- 删改
  - 删除
    - 首先用户主键匹配
    - 密码匹配
    - 访问数据库中的文章
    - 删除文件
    - 删除链接
  - 修改
    - 首先用户主键匹配
    - 密码匹配
    - 访问数据库中的文章
    - 调出文件
    - 调出到编辑器中
    - 重新修改
    - 确认修改
    - 覆盖原文件
    - 保存文件到数据库系统

●