文档编号：“面筋”APP– DSN – 1.0.0

“面筋”面试经验分享 APP
软件设计规格说明书

日期：2015-07-15

文档变更历史记录

序号	变更日期	变更人员	变更内容详情描述	版本
1	2015/07/15	潘浩彬	创建	0.9.9
2	2015/07/15	潘浩彬	增加部署图	1.0.0

目录

1、引言

1.1 编写目的

根据《需求规约说明书》等相关文档，综合考虑在实际生活场景的各种应用，本需求的编写目的在于平等地进行面试经验分享，倡导共同学习，共同成长的学习方式，使得求职者能够更好地去了解公司，找清楚自身的定位。

1.2 读者对象

本需求的预期读者是与本软件开发有联系的决策人，开发组成人员，扶助开发者，支持本项目的领导和公司，软件测试员。

1.3 软件项目概述

- 项目名称

  “面筋”面试经验分享平台

- 简称

  “MianJin”

- 项目代码托管地址

  https://github.com/joyeecheung/get-offer

- 软件项目的大致功能和性能要求

  本项目主要功能是以 BBS 的方式来分享用户的面试经历，入职信息，实习/就业感受以及知识

交流。

在性能上并没有很高的要求，用户只需要通过访问该网站，并能轻松查询相关的帖子。

1.4 文档概述

该文档主要从软件分析和设计的方向来阐述整个软件开发的流程和思路，文档中包含了软件相关业务流程的介绍以及图示，还包含了软件的组织形式和相关配置。本次开发采用了敏捷开发的方式，通过不断迭代的方式，逐渐明确需求，着重强调代码的高耦合低内聚，以较好的组织方式对代码进行组织和重构。

1.5 定义

文档中采用的缩略词以及相关术语:

MJ : 该软件”面筋”，

Topic : 用户发布的话题，

Reply : 用户回复的消息，

Node : App 拥有的模块

1.6 参考资料

Nodejs	www.nodejs.org
Polymer	docs.polymerchina.org

Mongodb www.mongodb.org

2. 软件设计约束

2.1 设计目标和原则

本设计欲实现主要的用例有:

- 1. 用户通过加入不同板块来发表自己的经验和看法
- 2. 用户检索并查看相关板块，来获取信息
- 3. 用户可以发起回答或者发表留言表示感谢或质询

2.2 设计约束

硬件平台 : 移动端浏览器（包括 Apple 设备, Android 设备）

OS 要求 : IOS 以及 Android OS

开发语言 : JavaScript

开发工具 : Nodejs, MongoDB, Polymer

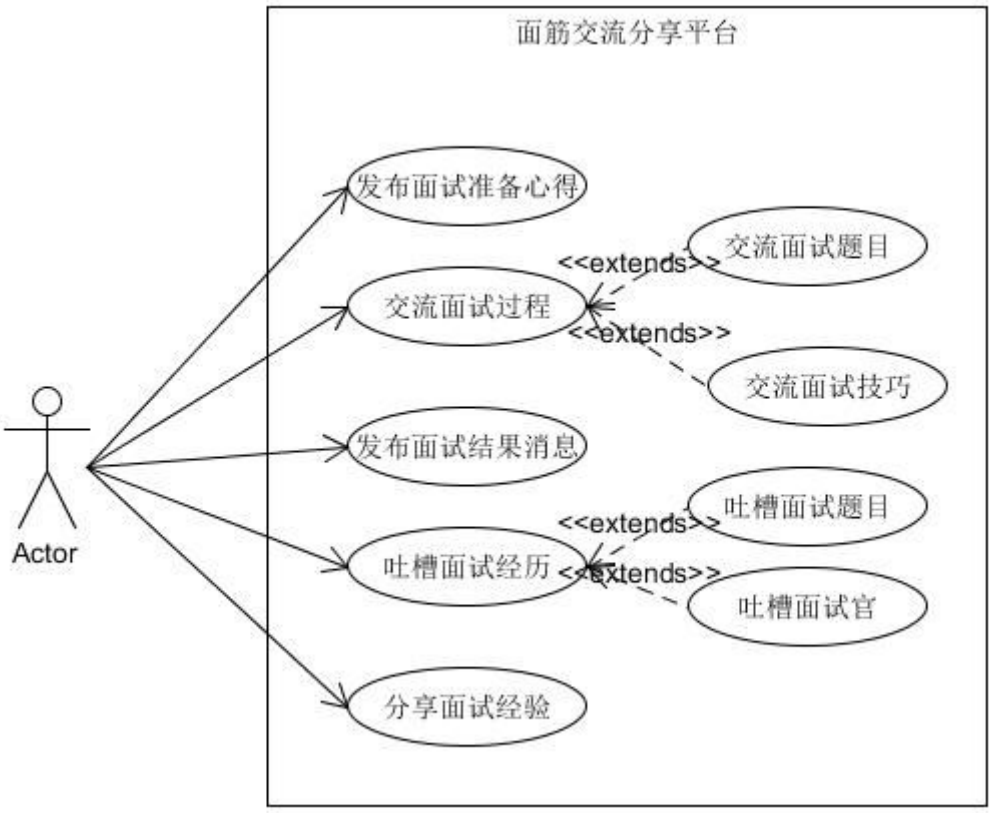
灵活性和配置要求:

需要配置 Nodejs 环境，MongoDB 环境以及 Polymer

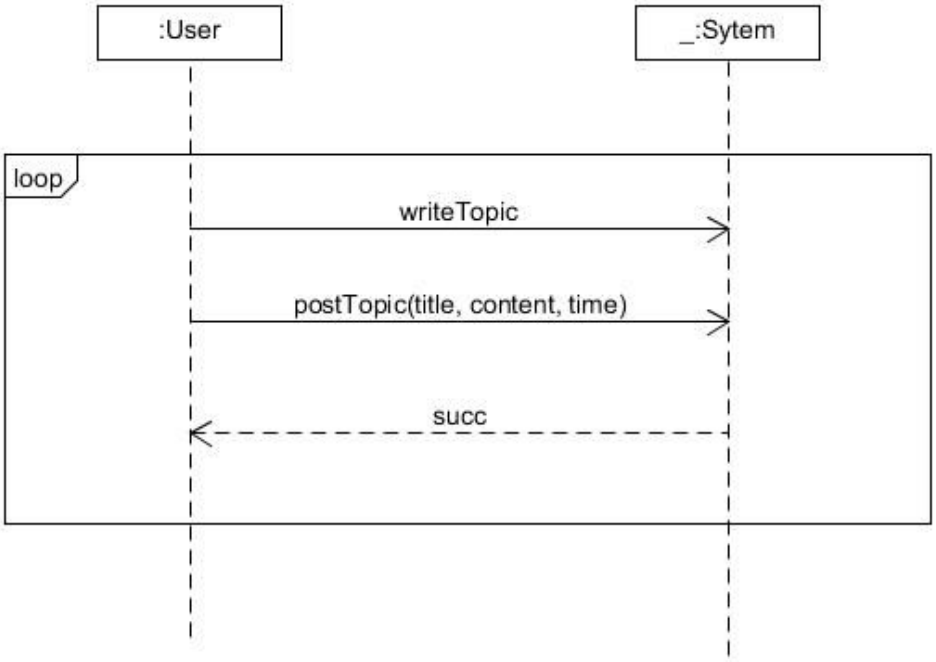
3.软件设计描述

3.1 总体结构设计

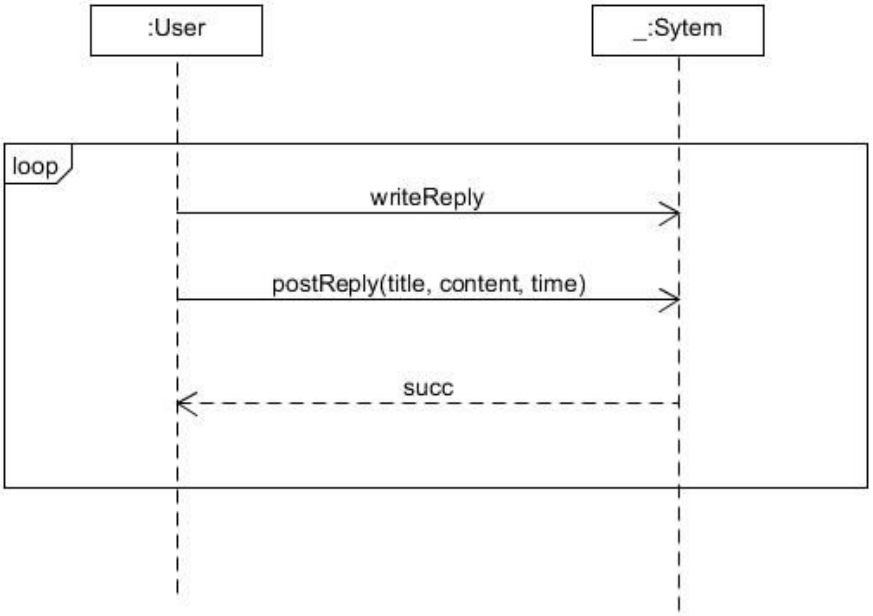
1. 根据用户的需求，绘制出系统用例



2. 根据用户的需求，绘制出系统顺序图

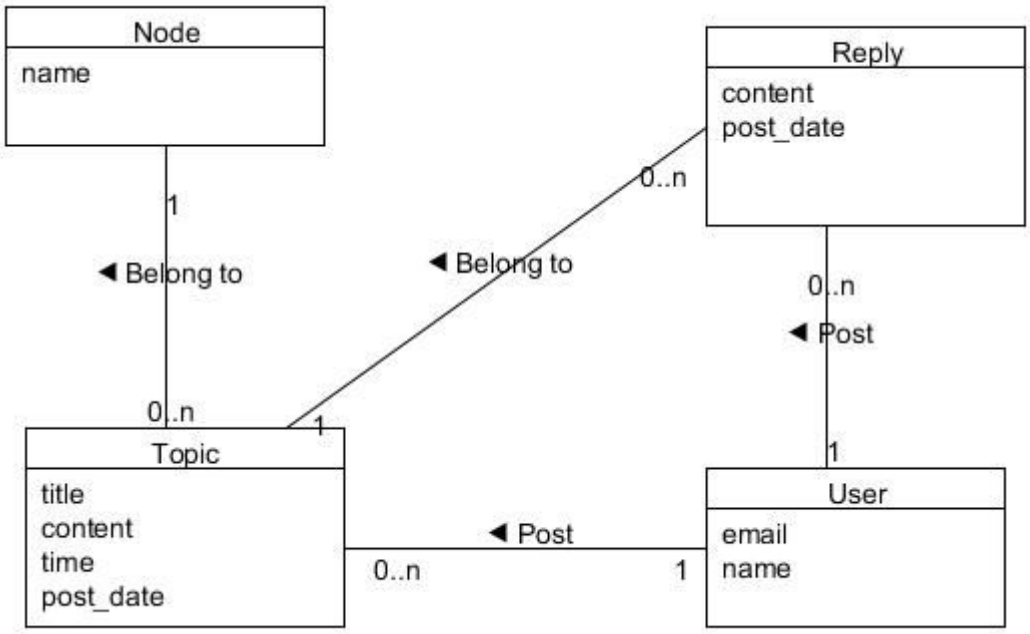


发布 Topic



用户回复

3.软件设计的领域模型:



3.2 数据设计

- 采用 MongoDB 为本次软件的数据存储，其弱一致性，能够保证用户的访问速度
- 文档结构的存储方式，能够更便捷的获取数据。
- 由 3.1 的领域模型，大致建立了如下 3 个数据库表

(1) node

字段:[

String name

]

(2) topic

字段:[

String title,

String content,

```
String user_name,
String user_email,
String node_name,  ; 外键关联
Date post_date,
Date last_update,
Int reply_count
]
```

(3) reply

```
字段: [
String content,
String user_name,
String user_email,
ObjectId topic_id, ;  外键关联
Date post_date
]
```

3.3 详细设计

设计主要分为前端, 后台以及数据库和测试

前端设计:

前端采用 Polymer, 实现了 web 组件的可重用, 该框架使用一个全新, 更快的数据绑定系统, 大大地精简了代码量, 而且以市面上主流浏览器为基准，使得其兼容性得到较为完美呈现，大大提高了前端的性能,兼容性以及组件的可重用

后台设计:

后台使用 nodejs 做开发，采用标准 MVC 架构以及 RESTful 架构风格，使得 MJ 的再开发更加简单易用，只需要了解其后台 API 即可开始使用，后台设计还使用了 markdown 以及 sanitize 中间件，使得网站更为安全可靠。

数据库设计:

MJ 采用了 MongoDB 作为其数据库，由于 MongoDB 自身的优势决定了其能以较快的速度满足用户的访问，同时在 nodejs 中使用 mongoose 来操作 MongoDB，

能确保数据库处于 Open 状态，确保 connection 激活，大大加快了数据库访问，并且节省了大量代码。

测试设计:

为了测试接口的可用性，采用了 nodejs 相关的工具来对 RESTful 的 API 进行测试，详细测试代码在 github 的 test 文件上，测试了包括连接，node 的 API，topic api, reply api, 以及主页的访问。

3.4 接口设计

- 在 MJ 设计的时候，考虑到兼容多平台的问题，因此在设计的时候，采用了 RESTful 风格，将前端与后台的通讯简化为 JSON 数据的传输，大大使得开发得以并行。

“面筋”的 API 大致如下:

GET ip/topic/:topic_name	获取该 topic_name 的相关讯息
POST ip/topic json_data	发布 topic
GET ip/node/:name	获取板块 name 相关讯息
GET ip/reply/:id	获取该 replayId 对应的 reply 相关信息
POST ip/reply json_data	发布回复

3.5 实际部署图