

中国科学技术大学

学士学位论文



面向幼教管理的家长教师 社交软件的设计与实现

姓 名 : 张鑫语
院 系 : 信息学院电子工程与信息科学系
学 号 : PB11210105
导 师 : 李斌 教授
完成时间 : 二〇一五年五月

University of Science and Technology of China
A dissertation for bachelor's degree



**The Design and Implementation of
Social Network Software for the
Preschool Education with
Management Concern**

Author : Xinyu Zhang
Department :
Student ID : PB11210105
Supervisor : Prof. Bin Li
Finished Time : May, 2015

目 录

目 录	I
插图索引	V
摘 要	VII
ABSTRACT	IX
第一章 绪论	1
1.1 研究背景和意义	1
1.2 国内外发展现状	1
1.3 课题主要内容与工作	3
1.4 论文结构	3
第二章 相关技术与知识	5
2.1 经典 EIM 系统总体介绍	5
2.2 社交 SNS 类型软件总体介绍	5
2.3 基于 EPM 管理系统的社交软件相关技术简介	8
2.3.1 Nosql 简介	9
2.3.2 Mongodb 简介	10
2.3.3 Nodejs 简介	11
2.3.4 Angularjs 简介	12
2.3.5 Hybrid HTML5 Mobile App 简介	12
2.3.6 WebSocket 简介	13
第三章 系统需求分析	15
3.1 管理端功能性需求分析	15
3.1.1 所有功能	15
3.1.2 教师基本功能	16
3.1.3 教师权限控制功能	16
3.2 社交端功能性需求分析	17
3.3 业务流程需求分析	17
3.4 非功能性需求分析	18

第四章 系统概要设计	21
4.1 技术方案介绍	21
4.2 数据库架构设计	21
4.3 系统后端功能架构设计	23
4.3.1 思路	23
4.3.2 系统模块架构	23
4.3.3 系统每个模块作用	25
4.3.4 系统模块设计要点	25
第五章 系统详细设计与实现	27
5.1 数据库架构设计	27
5.1.1 用户	27
5.1.2 学校	28
5.1.3 班级	28
5.1.4 状态	29
5.1.5 媒体 Media (文件的存储)	29
5.1.6 验证 Token	30
5.2 服务器端功能实现	30
5.2.1 Top Level 最上层代码	30
5.2.2 app.config 模块	31
5.2.3 app.model 模块和 app.thunkify 模块	31
5.2.4 app.log 模块	31
5.2.5 app.privilege 模块	32
5.2.6 app.notice 模块	32
5.2.7 app.data 命名空间	32
5.3 Web 管理端界面设计	33
5.4 Mobile App 社交界面设计	37
第六章 系统测试	41
6.1 测试环境和总体方案	41
6.1.1 系统功能性测试	41
6.1.2 系统界面测试	41

6.2 系统功能测试	41
6.2.1 学校的创建, 修改	41
6.2.2 班级的创建, 修改	42
6.2.3 网盘, 相册操作	43
6.2.4 教师, 学生的邀请, 转班, 删除	43
6.2.5 帮手的邀请, 修改, 删除	44
6.3 系统界面测试	45
6.3.1 电脑端	45
6.3.2 手机端	45
6.3.3 测试结果	45
第七章 总结与展望	47
7.1 总结	47
7.2 展望	47
参考文献	49
致 谢	51

插图索引

1.1 亲宝宝	2
1.2 好孩子	2
2.1 Enterprise Information Management System	5
2.2 facebook	6
2.3 微信	6
2.4 陌陌	7
2.5 airbnb	7
2.6 Spotify	8
2.7 Nosql	9
2.8 mongodb	10
2.9 Node	11
2.10 angular	12
2.11 Hybrid App Dev	12
2.12 websocket	13
3.1 业务流图	18
4.1 由业务流图而来的数据库设计	22
4.2 Node 模块关系	24
5.1 用户模型具体定义	27
5.2 学校模型具体定义	28
5.3 班级模型具体定义	28
5.4 状态模型具体定义	29
5.5 文件模型具体定义	29
5.6 验证 Token 模型具体定义	30
5.7 Web 管理端学校信息界面	33
5.8 Web 管理端班级列表界面	34
5.9 Web 管理端相册界面 1	34
5.10 Web 管理端相册界面 2	34
5.11 Web 管理端网盘界面 1	35
5.12 Web 管理端网盘界面 2	35
5.13 Web 管理端教师列表界面	36
5.14 Web 管理端人员发布通知界面	36
5.15 Web 管理端人员转班界面	36

5.16 Web 管理端帮手界面	37
5.17 App 时间线界面	37
5.18 App 发布状态界面	38
5.19 App 聊天界面	38
5.20 App 我的班级和学校界面	39

摘要

本文介绍了我实现的一个幼儿教育领域应用从需求调研到产品设计，到最后实现的流程和部分细节。在第一章绪论中，介绍了该应用的应用场景也就是：幼儿园老师和家长之间交互，幼儿园管理层对于老师的有效管理，家长对于自己孩子生活点滴的记录，家长之间针对孩子的社交。之后在第二章相关技术介绍中，对实现该产品的相关技术 Nodejs, Mongodb, Angularjs 做了简要介绍。在第三章系统需求分析中，详细而且细粒度地说明了用户到底需要哪些具体的功能，并给出了工作流图。在第四章系统概要设计中，给出了该系统的代码设计架构和数据库设计出发点。在第五章系统详细设计与实现中，给出了具体 Nosql 数据库结构化的数据定义和代码设计中各个模块的功能和部分实现细节。在第六章中，介绍了系统测试的内容和步骤。最后第七章总结了该系统的意义，和未来的工作计划。

关键词： 网络后端开发，数据库设计，移动开发，需求调研，界面设计，网络前端开发，NoSQL

ABSTRACT

This paper introduce the development of an application for areas of early childhood education, including NeedFinding, BenchMarking, Product Design, Implementation, Testing, Future Plan. In the first chapter, it explains the application scenarios: interaction between teachers and parents, management of teachers and classes, life records on children from parents, social networking between parents based on children. In the second chapter, it introduce technologies like Nodejs, Mongodb, Angularjs that used in this application. In the third chapter, it introduces the users' need, what functions they want and workflow chart. In the fourth chapter, it gives the code design structure and database design. In the fifth chapter, it provides the details of the database schema and the api interfaces and implementation details for each module in this application. In the sixth chapter, it introduces the testing procedure of this system. In the seventh chapter, it gives a brief summary of the significance of this application and future plan.

Keywords: Web Backend Development, DataBase Design, Mobile Development, NeedFinding, Interface Design, Web Frontend Development, NoSQL

第一章 绪论

1.1 研究背景和意义

该项目属于产品型而并非 research, 意在设计一种产品, 以 App 和 PC Web 端的形式来满足幼儿教育领域的用户需求.

当下随着互联网 Online To Offline 大潮的愈演愈烈, 人们的日益增长的需求, 互联网, 移动计算, 办公, 服务社交化极大地改变了人们生活的方式, 并且让市场愈发细化, 出现了大量特定领域的应用和软件。而这里设计并实现的一款产品, 是为了满足幼教领域中不同人群的各种需求 :

- 幼儿园管理层
- 幼儿园教师
- 宝宝家长

需求列表可以简单地概括为 :

- 幼儿园老师和家长之间交互
- 幼儿园管理层对于老师的有效管理
- 家长对于自己孩子生活点滴的记录
- 家长之间针对孩子的社交

该软件能够让幼儿园的管理更加轻松, 家长与老师和其他家长的联系更加紧密, 对于孩子的爱护更加贴心到位。

1.2 国内外发展现状

国内针对幼教母婴的类似软件已经存在, 比如 :



图 1.1 亲宝宝

亲宝宝是一个针对家长设计的 app，让亲友能够方便地记录宝宝的成长记录，并且有一定的分享功能。



图 1.2 好孩子

好孩子是一个面向幼儿园的人员管理系统，有一定的社交功能，但社交功能不强，而且因为产品设计问题，现在已经从 itunes 中下架，而且只支持 iphone 端。

现在大部分幼儿园并没有对于人员的管理的信息系统，老师和家长之间的交流方式主要通过微信，短信，QQ 群。

1.3 课题主要内容与工作

需要实现一个 Web 端，Web 端的功能主要是管理，面向用户为幼儿园管理层和幼儿园教师，管理层主要功能：

- 班级管理
 - 班级中文件，相册
 - 班级中人员的转班，入学
 - 教师给父母发送学生的教师评语
 - 班级中人员的转班，入学
 - 发布班级通知
- 学校管理
 - 学校中文件，相册
 - 学校中班级的管理
 - 发布学校通知
- 不同教师拥有不同的权限

在管理端的基础上，实现 App 端，App 端的功能主要是社交，老师，管理者，家长都会参与其中，社交的主要功能有：

- 每个用户都可以发布时间线
- 每个用户在自己主页看到同班用户发的时间线
- 可以在每个学校主页看到这个学校所有人的发的时间线
- 可以提出加入班级申请

1.4 论文结构

在第一章，我大致讲解了该产品的应用背景，简述了应当具有的功能，在第二章，我会介绍该产品使用到的技术和原则，第三章，我会进行详尽的需求分析。第四章，会给出系统的概要设计。在第五章，详细介绍系统的设计与实现细节。第六章会大致讲解系统的测试，最后第七章给出该项目的总结和展望。

第二章 相关技术与知识

2.1 经典 EIM 系统总体介绍

EIM 系统的英文全称是 Enterprise Information Management System. 在这个应用中需要首先实现一个适合幼儿园的学生教师人员管理的信息系统，作为整个系统的数据基础。

一个 EPM 系统一般使用关系型数据库，提供 Web 接口，让不同用户登录，同时操作。并且还需要有一定程度上的权限管理机制。在当下市场中，有很多现成的商业化 EPM 系统，为公司提供 General 的解决方案。

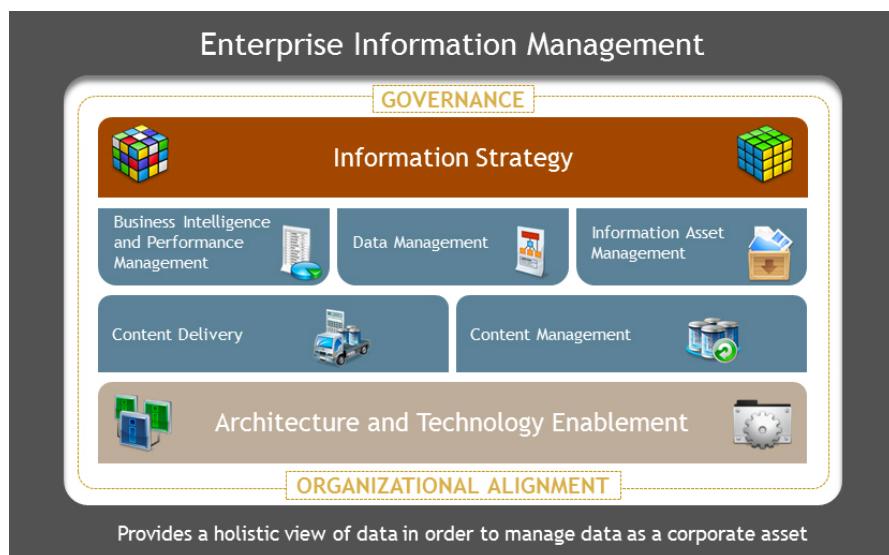


图 2.1 Enterprise Information Management System

但这些系统的问题往往是

- 不领域相关，而幼儿园有十分独特的人员管理需求
- 提供了大量不必要的功能
- 因为功能太多，也造成使用昂贵
- 不能加以扩展，比如支持社交

2.2 社交 SNS 类型软件总体介绍

社交 SNS 软件是互联网浪潮的一大标志，它满足了人们的聊天，分享，交友的需求。在这个应用中，基于管理端的数据，在老师和家长中做班级为单位的的社交，所以要实现一个适合该应用场景的社交 App.

一般一个社交 SNS 软件的扩展性，速度，及时性十分重要，而原子性并不强求。所以大多使用非关系型数据库，往往都会提供手机 App，不一定提供 Web PC 应用。在当下市场中，有很多社交软件，General 的社交软件最出名的是 Facebook

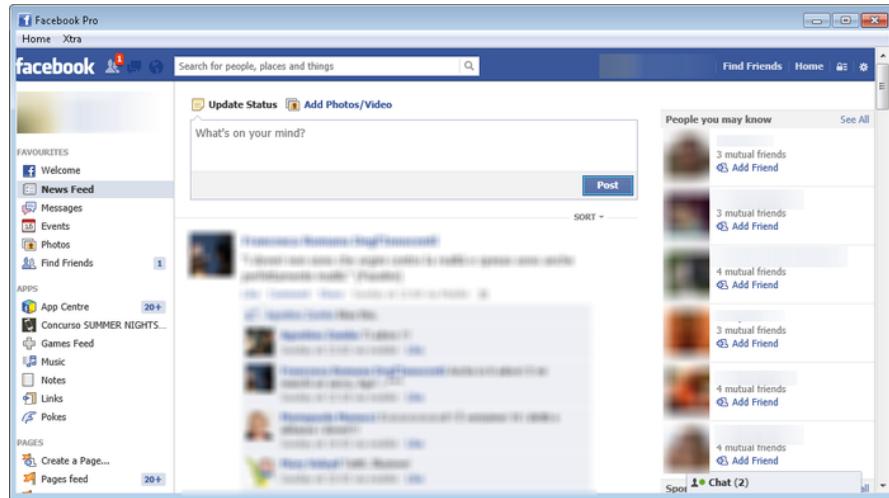


图 2.2 facebook

而对于细分的市场，各种软件百花齐放。

微信，基于熟人社交



图 2.3 微信

陌陌，基于陌生人社交



图 2.4 陌陌

Airbnb, 基于租房社交



图 2.5 airbnb

Spotify, 音乐社交, 基于 Facebook



图 2.6 Spotify

在这里，要实现的社交是基于细分的幼儿教育市场。

2.3 基于 EPM 管理系统的社交软件相关技术简介

对于该应用，由于要加入社交元素，加之人员管理系统只需要针对幼教领域来设计功能，所以使用了非关系型数据库 mongodb。

在 Web 后端使用了 Nodejs 来进行程序开发，web 前端操作逻辑比较复杂，所以使用了 angularjs 前端框架。因为前后端交互比较频繁，大量使用了 websocket 而不是 Http Ajax 访问来进行前后端交互。

App 的开发往往比较费时，尤其是考虑到如何跨平台。在这里我选用了 Hybrid Html5 Mobile 开发框架，可以使用 Web 技术来开发 ios & android 平台 app，一份代码，两个平台使用。而且 Web 技术也降低了开发门槛和学习曲线。

2.3.1 Nosql 简介



图 2.7 Nosql

NoSQL 有时也称作 Not Only SQL 的缩写，是对不同于传统的关联式数据库的数据库管理系统的统称。（注依据 Martin Fowler，NoSQL 不是英文 Not Only SQL，因为这会是 NOSQL 而不是 NoSQL）

两者存在许多显著的不同点，其中最重要的是 NoSQL 不使用 SQL 作为查询语言。其数据存储可以不需要固定的表格模式，也经常会避免使用 SQL 的 JOIN 操作，一般有水平可扩展性的特征。NoSQL 的实现具有二个特征：使用硬盘，或者把随机存储器作存储载体。

当代典型的关联式数据库在一些数据敏感的应用中表现了糟糕的性能，例如为巨量文档建立索引、高流量网站的网页服务，以及发送流式媒体关系型数据库的典型实现主要被调整用于执行规模小而读写频繁，或者大批量极少写访问的事务。

NoSQL 的结构通常提供弱一致性的保证，如最终一致性，或交易仅限于单个的数据项。不过，有些系统，提供完整的 ACID 保证在某些情况下，增加了补充中间件层（例如，CloudTPS。有两个成熟的系统有提供快照隔离的列存储：像是 Google 基于过滤器系统的 BigTable，和滑铁卢大学发展的 HBase。这些系统，自主开发，使用类似的概念来实现多行（multi-row）分散式 ACID 交易的快照隔离（snapshot isolation）保证为基础列储存，无需额外的资料管理开销，中间件系统部署或维护，减少了中间件层。

少数 NoSQL 系统部署了分布式结构，通常使用分散式杂凑表（DHT）将数

据以冗余方式保存在多台服务器上。依此，扩充系统时候添加服务器更容易，并且扩大了对服务器失效的承受能程度

2.3.2 MongodB 简介



图 2.8 mongodb

是由 C++ 语言编写的开源数据库系统。在高负载的情况下，添加更多的节点，可以保证服务器性能。MongoDB 旨在为 WEB 应用提供可扩展的高性能数据存储解决方案。

MongoDB 是一种强大、灵活、追求性能、易扩展的数据存储方式。是面向文档的数据库，不是关系型数据库，是 NoSQL(not only SQL) 的一种。所谓的面向文档，就是将原来关系型数据库中的“行”的概念换成了更加灵活的“文档”，以文档为存储单位。文档的值可以是数组、文档等复杂的数据模型。并且文档的键不会事先定义也不会固定不变。mongoDB 设计的主要思想之一就是，将能交给客户端的操作都要从服务端转移到客户端，例如生成 objectid 等操作。

2.3.3 Nodejs 简介

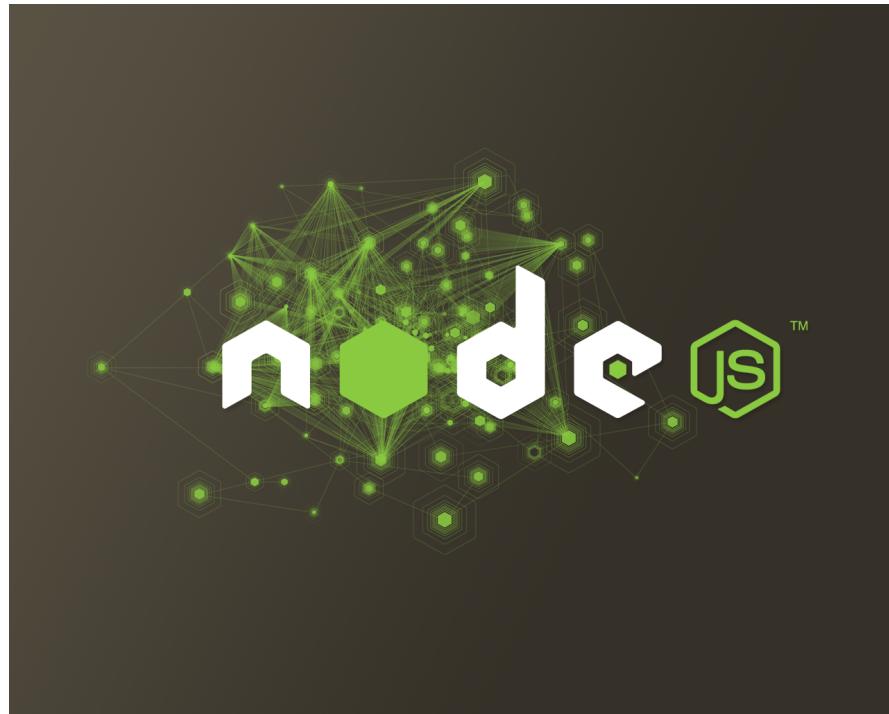


图 2.9 Node

Node.js 是一个基于 Chrome JavaScript 运行时建立的平台，用于方便地搭建响应速度快、易于扩展的网络应用。Node.js 使用事件驱动，非阻塞 I/O 模型而得以轻量和高效，非常适合在分布式设备上运行的数据密集型的实时应用。

V8 引擎执行 Javascript 的速度非常快，性能非常好。

Node 是一个 Javascript 运行环境 (runtime)。实际上它是对 Google V8 引擎进行了封装。V8 引擎执行 Javascript 的速度非常快，性能非常好。Node 对一些特殊用例进行了优化，提供了替代的 API，使得 V8 在非浏览器环境下运行得更好。

2.3.4 Angularjs 简介



图 2.10 angular

AngularJS 是一款开源 JavaScript 函式库，由 Google 维护，用来协助单页面应用程序运行的。它的目标是透过 MVC 模式（MVC）功能增强基于浏览器的应用，使开发和测试变得更加容易。

函式库读取包含附加自定义（标签属性）的 HTML，遵从这些自定义属性中的指令，并将页面中的输入或输出与由 JavaScript 变量表示的模型绑定起来。这些 JavaScript 变量的值可以手工设置，或者从静态或动态 JSON 资源中获取。

2.3.5 Hybrid HTML5 Mobile App 简介

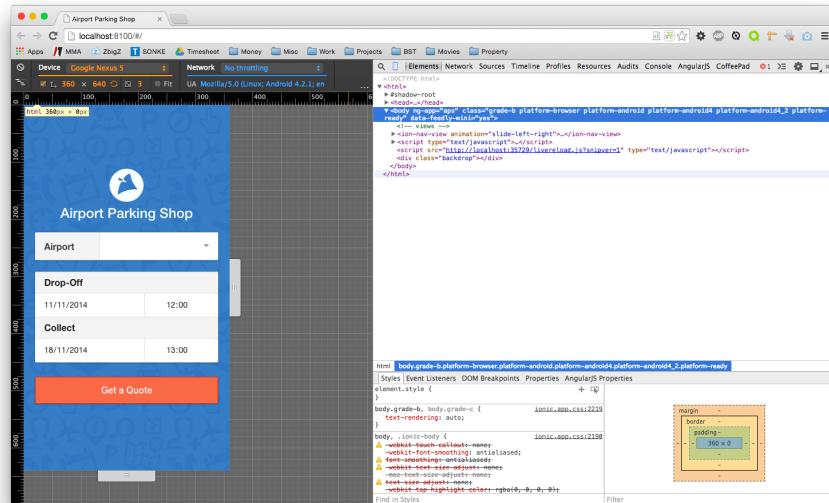


图 2.11 Hybrid App Dev

Hybrid App 是指介于 web-app、native-app 这两者之间的 app, 它虽然看上去是一个 Native App, 但只有一个 UI WebView, 里面访问的是一个 Web App, 但其通过 JSB (javascript Binding) 来和系统底层的 api 交流, 来达到 native app 的功能, 但却可以使用 Web 技术来渲染 UI。

另一种方式是使用虚拟的 HTML DOM, 把 Html Dom 来翻译为系统的 UI 布局代码, 来达到原生应用的 UI 渲染效率。

现有有很多 Html5 混合手机开发框架, React Native, ionic 等等, 我选用了 ionic.

2.3.6 WebSocket 简介

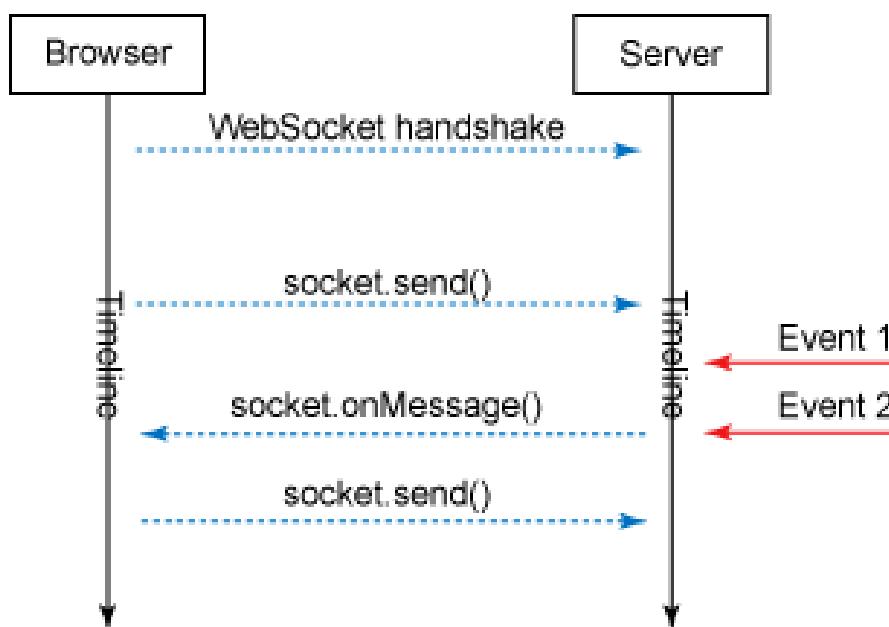


图 2.12 websocket

WebSocket 是 HTML5 开始提供的一种在单个 TCP 连接上进行全双工通讯的协议。WebSocket 通信协议于 2011 年被 IETF 定为标准 RFC 6455, WebSocket API 被 W3C 定为标准。

在 WebSocket API 中, 浏览器和服务器只需要做一个握手的动作, 然后, 浏览器和服务器之间就形成了一条快速通道。两者之间就直接可以数据互相传送。

第三章 系统需求分析

3.1 管理端功能性需求分析

3.1.1 所有功能

管理端面对两种用户，分别为幼儿园管理者和教师，他们分别有不同的需求。下面首先分析幼儿园管理者角色所要求的功能：

- 注册/登录
- 将自己的幼儿园录入系统
 - 幼儿园创建
 - 删除
 - 编辑相关信息
- 幼儿园有自己的
 - 相册
 - * 上传
 - * 修改图片名和介绍
 - 云文件
 - * 上传
 - * 修改名字和介绍
 - * 创建文件夹
 - * 移动文件
- 发布校园通知
- 在幼儿园录入系统后，在每个隶属于自己的幼儿园名下录入班级
 - 创建班级
 - 删除
 - 编辑相关信息
- 每个班级都有自己的
 - 相册
 - 云文件

- 发布班级通知
- 班级内部的人员管理
 - 邀请教师/学生
 - 删除教师
 - 教师转班
 - 申请删除学生
 - 申请学生转班
- 对学生写评语
- 如果班级和学生数量比较大，则少数的管理者不可能承担这么多的工作，而且管理者本身不愿意也不应当干学生数据修改等底层操作
 - 委派帮手
 - 以给予手下教师不同权限的形式
 - 比如有的教师可以做班级内的人员管理，有的教师允许发布校园通知

以上是管理者所拥有的功能，事实上也是该系统 PC 管理端要提供的全部功能，因为管理者权限最高，应当能够使用到全部的功能。

对于教师，分为基本功能和权限下功能。基本功能所有教师都有，权限下每个功能只有管理者分配了权限，才可以拥有。

3.1.2 教师基本功能

- 管理自己所在班级的
 - 相册
 - 班级文件
- 发布自己所在班级通知
- 对自己班级内的学生写评语

3.1.3 教师权限控制功能

- 校园层级上的功能
 - 发布幼儿园通知
 - 修改幼儿园基本信息

- 班级层级上的功能

- 创建班级
- 删除班级
- 编辑相关信息

3.2 社交端功能性需求分析

社交端功能体现在手机 App 上，没有用户的权限之说。下面对社交端所需的功能做一个列表。

- 不提供好友功能
- 用户自动被分入相应班级的聊天群
- 班级/学校有时间轴，由其中父母和教师发布的状态组成
- 发布状态，可以是文字，图片或者是视频
- 看到同班(可选为同校)的其他父母发的状态，并且评论或者点赞
- 每个用户发布的状态会同步到该用户所在的班级和学校
- 查看
 - 学校通知
 - 班级通知
 - 教师评语
- 浏览自己孩子所在班级和学校的
 - 相册
 - 文件
- 私信聊天功能

3.3 业务流程需求分析

业务流由下面的业务流图说明：

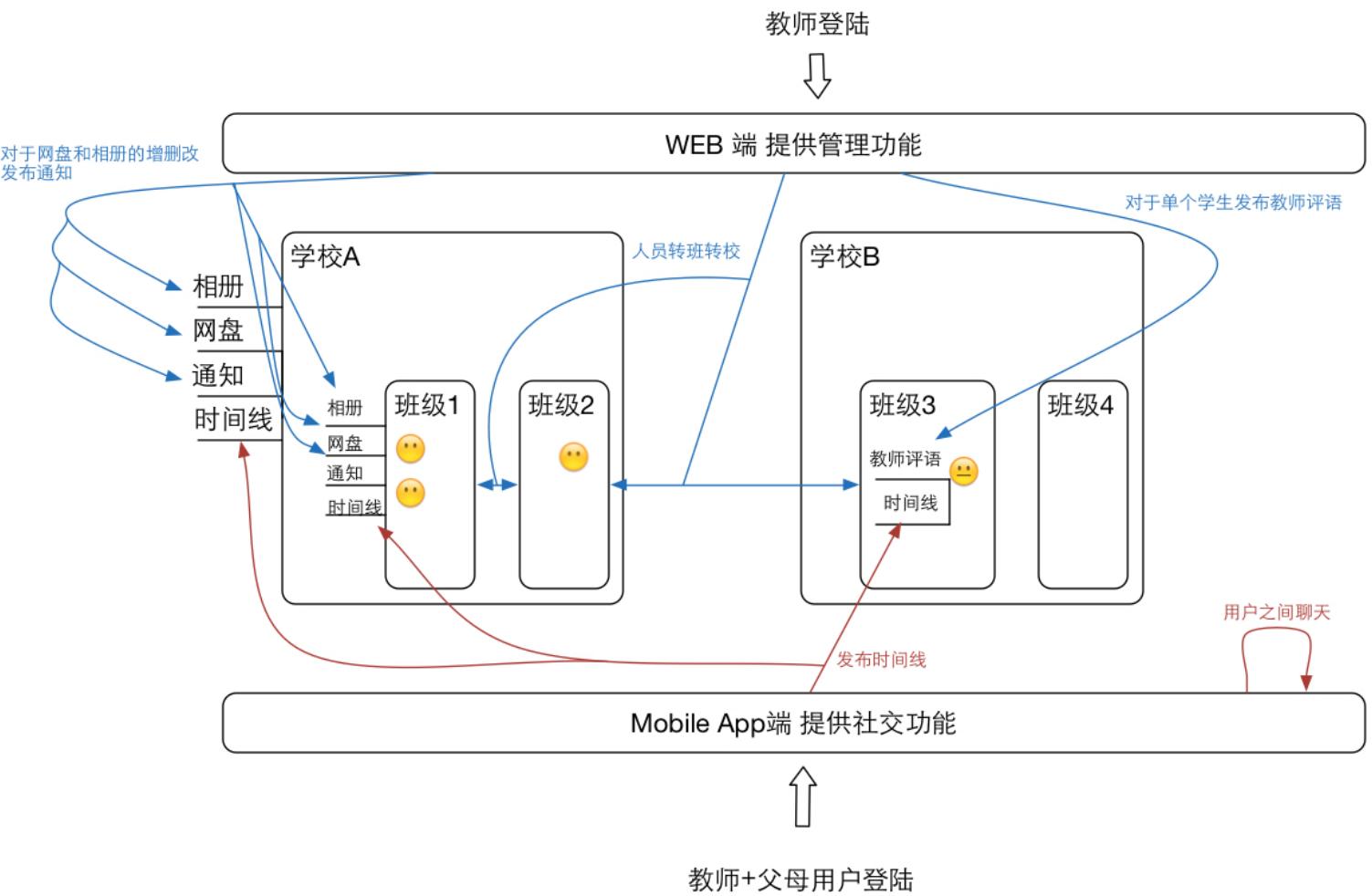


图 3.1 业务流图

业务流程为，教师通过 Web 端登陆进行

- 对于幼儿园数据的管理
- 发布通知
- 给父母发送教师评语

父母和教师用户都可以从 Mobile App 端登陆

- 以班级为单位的群聊
- 私信
- 发布时间线
- 用户的时间线可以同步在其所在的学校，班级

3.4 非功能性需求分析

用户有一些非功能性的需求，下面列出来

- 保护学生用户（也就是父母）的权利

- 转班，入学，从班级中删除等操作必须要通过父母的短信验证
- 如果是父母自行从班级中撤出，不需要验证
- 管理者账号有权加入任何班级群
- 对于教师账号，为了方便管理端的操作
 - 教师加入某个班级需要教师短信验证
 - 而对于教师转班，从班级中撤出操作，可以管理者直接操作，不需要验证

第四章 系统概要设计

4.1 技术方案介绍

该应用在服务器端采用 Node.js 编写，其好处是 Web 开发生态环境好，现成的服务和工具多，在这里我使用 npm 提供的 private node modules 和 github 提供的付费 organization 的服务，将整个大程序设计为由分成多个 Node.js 的模块组成。

4.2 数据库架构设计

下面

根据我们之前给的需求图，我们设计出了我们的 mongodb 数据库架构。见下图：

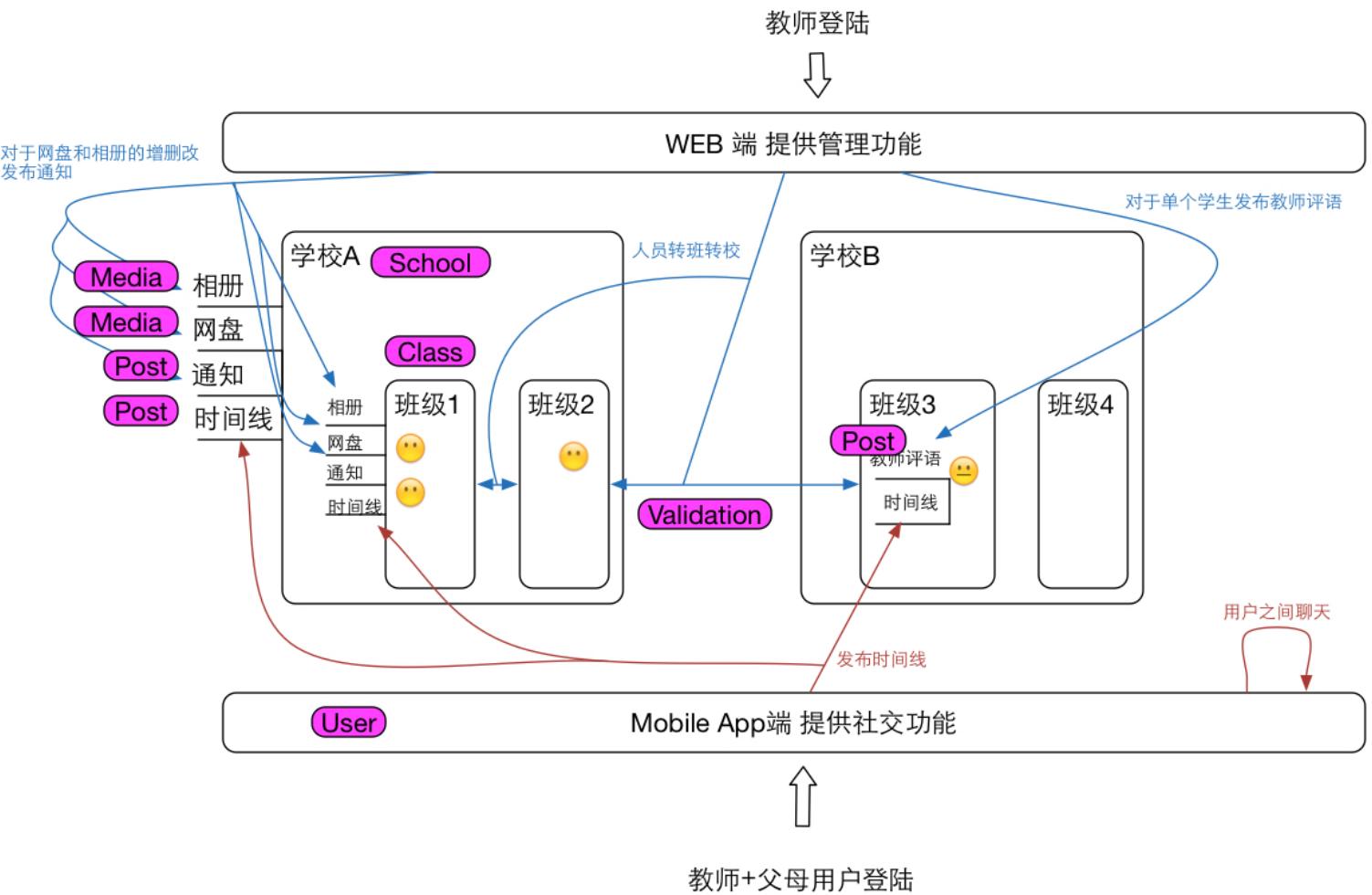


图 4.1 由业务流图而来的数据库设计

由上图可见，其中紫色矩形部分为数据库中定义的 model，分别为：

Post

用户发布在时间线的状态

User

用户模型

School

学校模型

Class

班级模型

Media

文件模型，用于网盘和相册

ValidationToken

验证 Token，用于转班验证，密码 reset 等

4.3 系统后端功能架构设计

4.3.1 思路

在系统后端，我运行了两个服务器，一个是普通的 http 服务器，另一个是 websocket 服务器。Http 服务器来处理普通的 http 请求，并提供一些 restful 的接口。websocket 服务器和 WEB 端的管理用户建立 socket 连接，实现双向的通信，以达到

- 更加安全
- 比 ajax http 请求更加快捷，高效

而这两个服务器提供的功能由一些 node modules 组成。首先先要设计程序由哪些部分组成，之后将这些部分设计为 node modules。

4.3.2 系统模块架构

这里先给出 node modules 的树形关系图，modules 之间关系的设计借鉴了 java 中 package 命名空间的规范。在给出关系图后，会讲解每个 module 的功能，和应当暴露出来的接口。node modules 树形关系图类似 java 程序的类图。

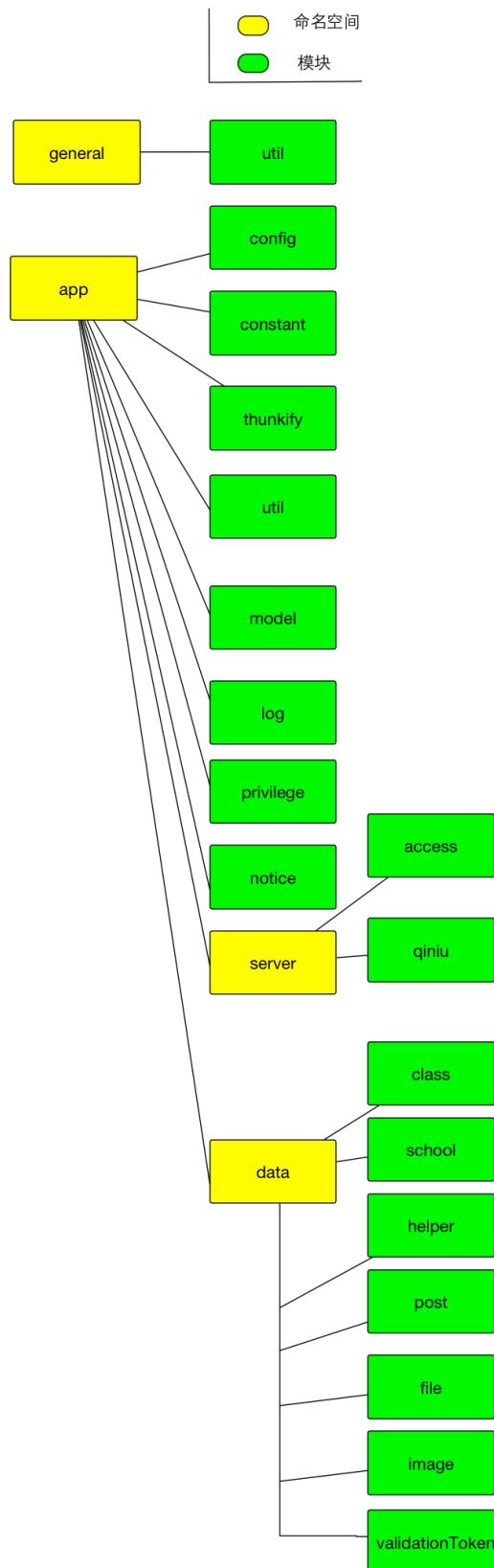


图 4.2 Node 模块关系

4.3.3 系统每个模块作用

general.util 放一些任何地方都可能被重用到，且与该应用关系不大的代码

app.config 配置信息

app.constant 使用的一些常量

app.model 数据库定义

app.thunkify 数据库的一个浅层包装，对数据库部分操作提供 promise 形式的接口

app.util 一些很多地方都可能被重用到，但与该应用场景耦合度较高的代码

app.log 日志系统

app.notice 发短信，邮件，手机端 Push Notification 通知系统

app.privilege 系统对于权限相关的操作

app.server.access 系统服务器的验证(登陆，注册)系统

app.server.qiniu 系统使用的七牛云服务操作

app.data.class 对于数据库中班级的抽象层

app.data.school 对于数据库中学校的抽象层

app.data.helper 对于数据库中帮手的抽象层

app.data.file 对于数据库中文件的抽象层

app.data.image 对于数据库中相册的抽象层

app.data.validationToken 对于数据库中验证的抽象层

4.3.4 系统模块设计要点

1. 要保证这些 modules 直接高内聚，低耦合，直接互相依赖度低，都可以被当做独立的工具使用
2. 使用 Github organization，每个 modules 都对应 organization 中的一个 repository.
3. 将每个 repository 都发布为私有的 NPM 模块，这样可以在任何地方使用 npm install 来进行模块的安装，使得该应用的每个部分都成为独立的工具

4. 最上层的程序应当很短，只是调用相应模块，创建服务器并将模块注入其中，之后服务器开始运行

第五章 系统详细设计与实现

5.1 数据库架构设计

下面我给出数据库的详细设计，因为使用的是 nosql，所以并没有 sql 经典的设计表，而是结构化的数据定义

5.1.1 用户



图 5.1 用户模型具体定义

5.1.2 学校

```
object {9}
    avatar : String 学校头像
    name : String 学校名称
    description : String 学校介绍
    classList [2]
        0 : 该学校的班级列表 为以下类型的数组
        1 : ObjectId Class对象的Id 每个对应一个班级
    upvotes : Number 被赞的次数
    timeLine [2]
        0 : 该学校状态墙 为以下类型数组
        1 : ObjectId 状态Post的Id 每个对应一个发布的状态
    album [2]
        0 : 该学校相册 为以下类型数组
        1 : ObjectId Media对象的Id 每个对应一张图片
    fileRoot : ObjectId Media对象的Id 学校网盘的根文件夹
    schoolPost [2]
        0 : 学校通知列表 为以下类型的数组
        1 : ObjectId 状态Post的Id 每个对应一个Post通知
```

图 5.2 学校模型具体定义

5.1.3 班级

```
object {12}
  avatar : String 班级头像
  name : String 班级名称
  description : String 班级介绍
  teacherList [2]
    0 : 该班级的老师列表 为以下对象的数组
    1 {2}
      userId : ObjectId User对象的Id 每个对应一个教师
      status : String 该教师在本班内的状态
  studentList [2]
    0 : 该班级的学生列表 为以下对象的数组
    1 {2}
      userId : ObjectId User对象的Id 每个对应一个学生
      status : String 该学生在本班内的状态
  public : Boolean 是否为公开课班级 (公开则可以被任何用户搜索并访问到)
  reserve : Boolean 如果为真, 则班级不能被删除
  upvotes : Number 被赞的次数
  timeLine [2]
    0 : 该班级状态墙 为以下类型数组
    1 : ObjectId 状态Post的Id 每个对应一个发布的状态
  album [2]
    0 : 该班级相册 为以下类型数组
    1 : ObjectId Media对象的Id 每个对应一张图片
  fileRoot : ObjectId Media对象的Id 班级网盘的根文件夹
  classPost [2]
    0 : 班级通知列表 为以下类型的数组
    1 : ObjectId 状态Post的Id 每个对应一个Post通知
```

图 5.3 班级模型具体定义

5.1.4 状态

图 5.4 状态模型具体定义

5.1.5 媒体 Media (文件的存储)

图 5.5 文件模型具体定义

5.1.6 验证 Token

```

    □ ▼ object {5}
    □ □ type : String 该validation token 对应那种类型的验证
    □ □ employer : Mixed 动态类型可以存储各种类型数据 本次验证所涉及到的雇主相关信息
    □ □ executor : Mixed 动态类型可以存储各种类型数据 本次验证发起人的相关信息
    □ □ who : Mixed 动态类型可以存储各种类型数据 本次验证针对人的相关信息
    □ □ anything : Mixed 其他与该验证有关的信息
  
```

图 5.6 验证 Token 模型具体定义

5.2 服务器端功能实现

下面给出系统关键模块暴露出的 api, 和一些 api 的实现解释和细节

5.2.1 Top Level 最上层代码

下面给出整个 app 最顶层程序的伪代码

```

var config = require(app.config)
server = http.createServer(config.port);
socket_server = socket.io(server);
var modules = config.server_modules;

modules.forEach(function(module_name){
  var md = require(module_name);

  md.socket.forEach(function(obj){
    socket_server.on(obj.name, obj.function)
  });

  md.server.forEach(function(obj){
    server.inject(obj.method, obj.path, obj.function);
  });
});

server.listen(function(err){
  if(!err)
    console.log(server started successfully.);
});
  
```

其功能是

- 加载 app 的配置模块 config
- 根据 app 配置的端口创建 Http 服务器, 并由 Http 服务器创建 Websocket 服务器
- 加载 config 中设置的要向服务器中导入的模块列表
- 将所有加载的模块注入服务器
- 开启服务器, app 正式开始运行

5.2.2 app.config 模块

里面要配置的内容为

- 服务器要运行哪个端口
- 要加载的模块
- 日志的粒度
- 是否运行在 DEBUG 模式
- 数据库的细节配置，比如如何处理索引

5.2.3 app.model 模块和 app.thunkify 模块

数据持久化层的定义，前面已经说得很清楚了。

5.2.4 app.log 模块

是 app 的日志模块，对于一个 web app 来说，日志记录非常重要。日志记录有以下作用：

1. 检测黑客对于服务器攻击
2. 记录用户对于 app 的错误使用，方便下一版本修改
3. 记录用户使用 app 的习惯，用于更好的设计产品
4. 记录服务器出的错误，找出 bug

在该 app 中，我记录日志的方法为：

先写在一个文件中，之后用一个定时运行的程序，把该文件所有内容写入数据库。作为数据的持久化。

记录日志的种类为：

- DEBUG 调试信息
- TRACE 也是只有调试的时候会用，细节信息，比如进入了哪个函数等等
- ERROR 发生的系统错误
- WARN 系统的警告，比如疑似的攻击
- INFO 用户操作的记录

5.2.5 app.privilege 模块

该模块管理系统提供了管理教师用户的权限机制

分为：

- HumanResource 可以管理班级内的人员
- InfoManager 可以在任何班级和自己雇主的学校发布通知，和修改班级，学校介绍
- BOSS 拥有所有权限（不只是上面的两种操作，比如学校名称的修改）

通过或操作来完成权限的相加，与非操作来完成权限的相减。

5.2.6 app.notice 模块

使用了 sendcloud.sohu.com 和极光推送的服务，完成发短信，发邮件通知和手机端 Push Notification 的功能。

需要通知的操作一般为：

- 人员的转班，邀请等等
- 手机端聊天的提醒
- 忘记密码

5.2.7 app.data 命名空间

在空间下，是对于底层数据定义的抽象层

5.2.7.1 class 班级和 school 学校

- 提供学校班级的创建，修改，删除
- 这里删除使用软删除，不真正删除数据而是用一个 Deleted 的布尔标记
- 人员转班转校

5.2.7.2 post 状态

- 发送状态
- 删除所发状态

5.2.7.3 file 文件和 image 图片

- 文件的删除，修改，新建
- 文件夹的展开
- 删除依旧使用软删除
- 相册是一个只有一层的文件夹

5.2.7.4 app.validationToken 模块

处理申请 request 操作

- 比如用户提交转班，邀请某人加入某班的申请，该模块会创建一个 validationToken
- validationToken 创建的时候，会执行验证的 Pre 函数，该函数会根据该 Token 的类型，来执行验证的预操作，比如改变用户的状态。
- 当用户点击验证连接同意验证的时候，该模块会根据 Token 的类型，执行 Post 函数，完成验证的后续操作，真正改变用户班级归属。

5.3 Web 管理端界面设计

- 学校信息界面

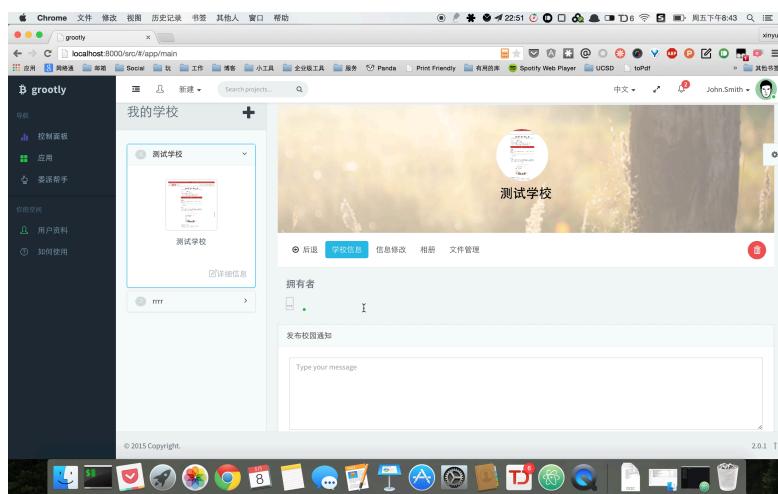


图 5.7 Web 管理端学校信息界面

- 班级列表界面

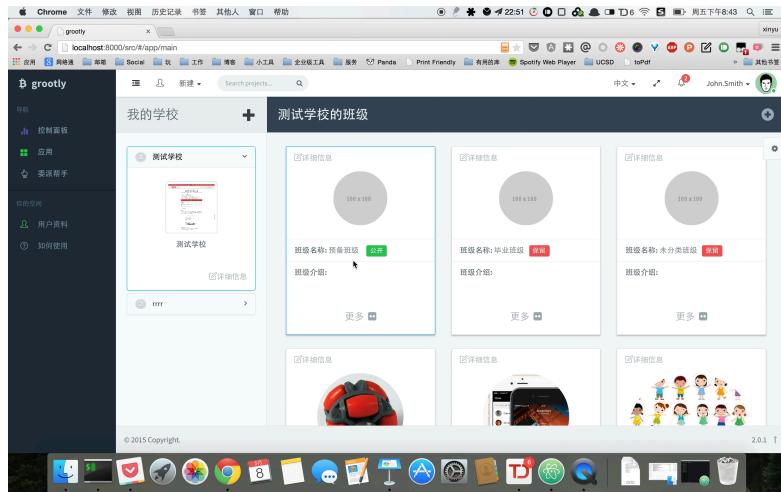


图 5.8 Web 管理端班级列表界面

- 网盘界面

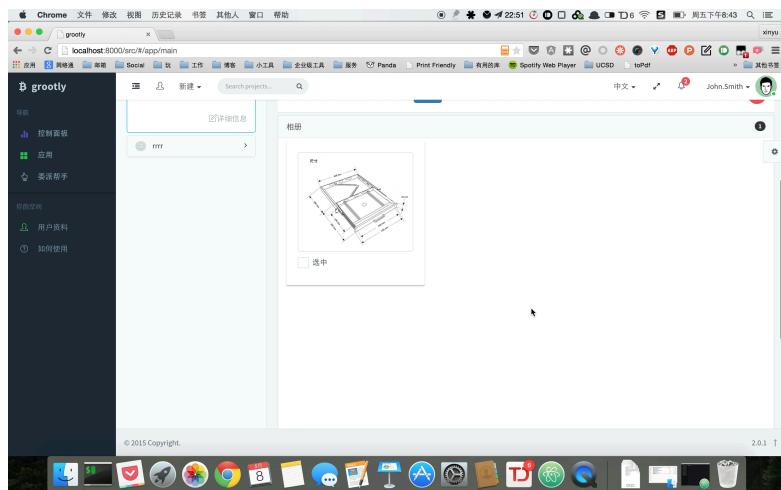


图 5.9 Web 管理端相册界面 1

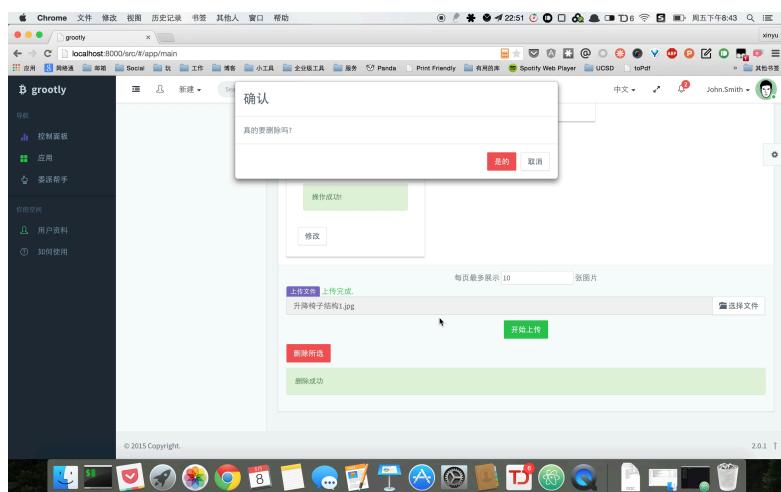


图 5.10 Web 管理端相册界面 2

- 相册界面

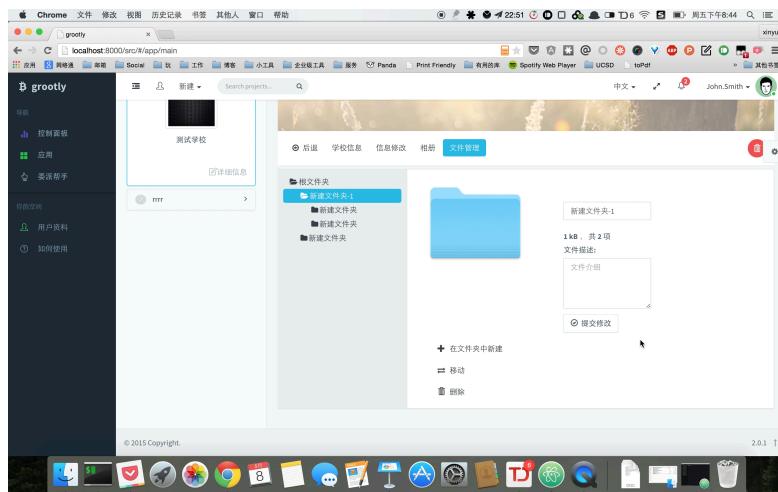


图 5.11 Web 管理端网盘界面 1

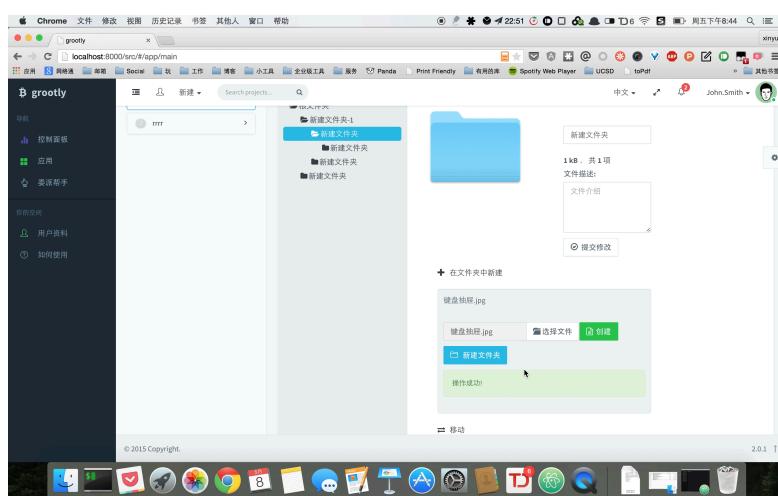


图 5.12 Web 管理端网盘界面 2

- 教师、学生的管理界面

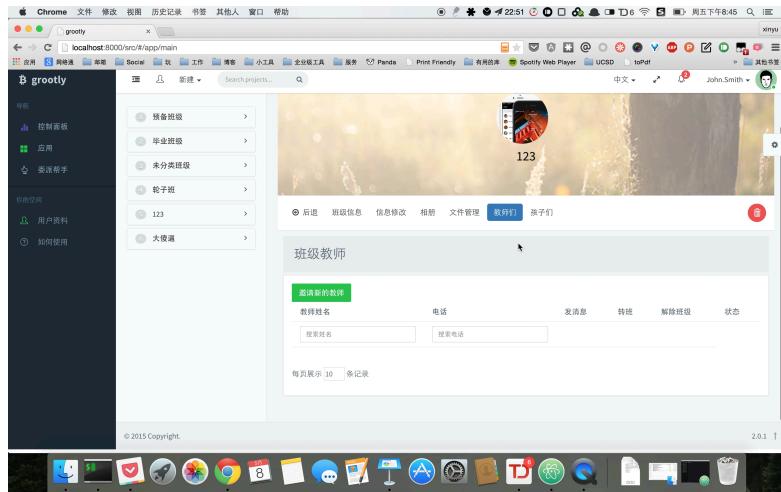


图 5.13 Web 管理端教师列表界面

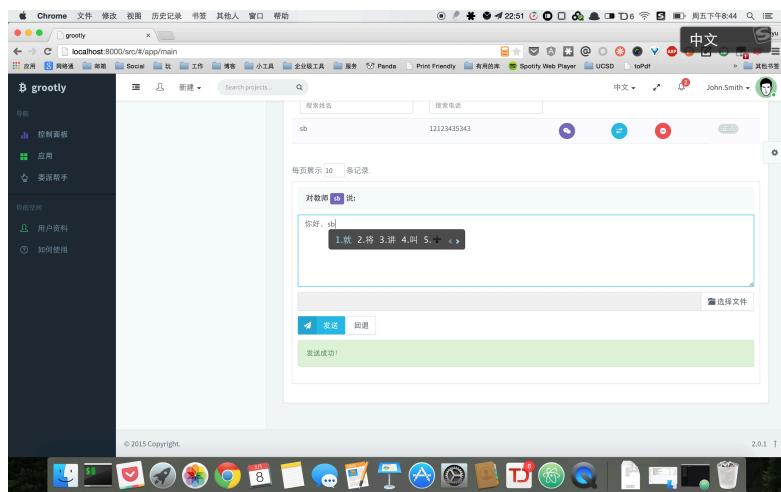


图 5.14 Web 管理端人员发布通知界面

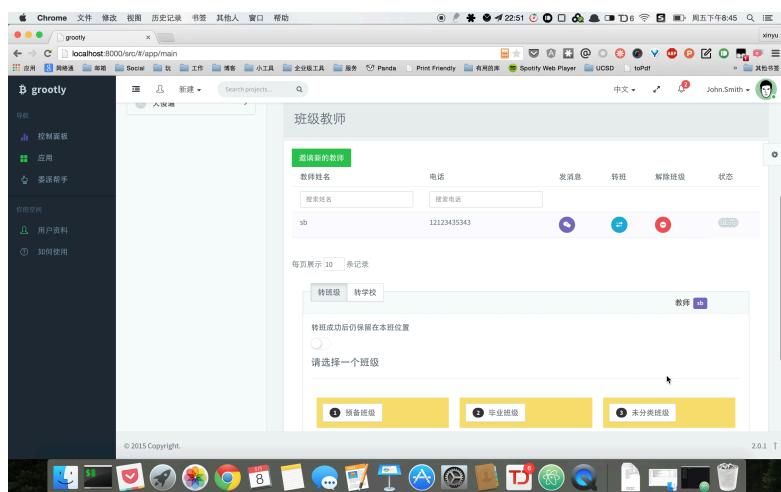


图 5.15 Web 管理端人员转班界面

• 帮手界面

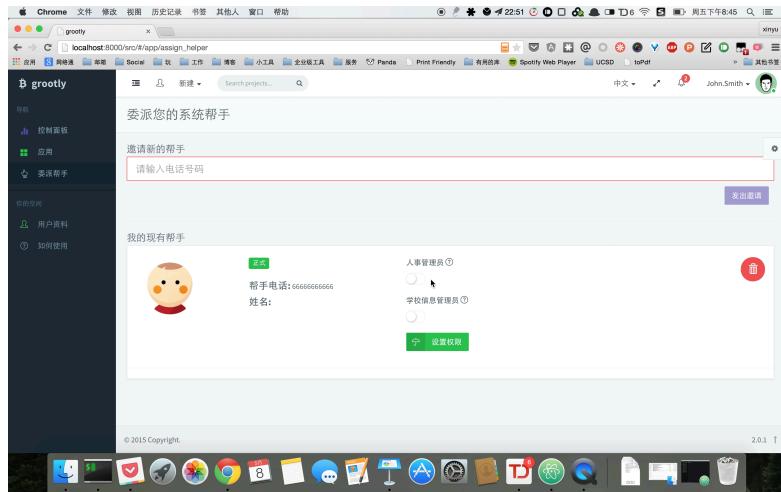


图 5.16 Web 管理端帮手界面

5.4 Mobile App 社交界面设计

- 浏览状态界面

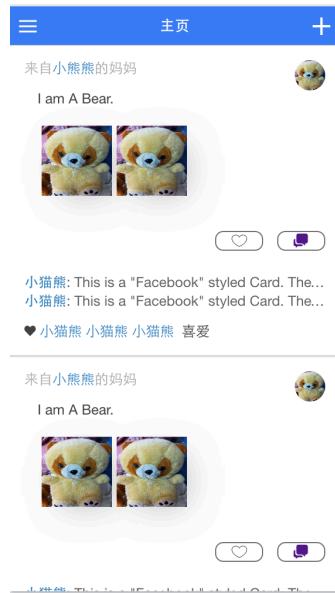


图 5.17 App 时间线界面

- 发布状态界面

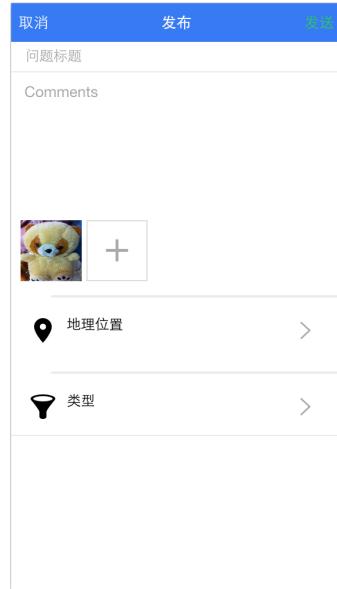


图 5.18 App 发布状态界面

• 聊天界面



图 5.19 App 聊天界面

• 我的班级和学校界面



图 5.20 App 我的班级和学校界面

第六章 系统测试

6.1 测试环境和总体方案

测试方案：

6.1.1 系统功能性测试

系统的手机端并没有完全完成，所以功能性测试主要针对于 Web 管理端。功能分为：

- 学校的创建，修改（删除功能暂时没有实现）
- 班级的创建，删除，修改
- 网盘，相册操作
- 教师，学生的邀请，转班，删除
- 帮手的邀请，修改，删除

测试环境为：Mac osx 10.10, Chrome 39

6.1.2 系统界面测试

分为 PC 端测试，和手机端测试

PC 端测试环境: Mac osx 10.10, Chrome 39

Mobile 端测试环境: iphone5s

6.2 系统功能测试

6.2.1 学校的创建，修改

表 6.1 学校的创建

测试名称	学校的创建
测试人	张鑫语
测试日期	2015-05-11
测试步骤	点击左边 Panel 的加号，输入学校数据
输入数据	学校名：测试学校，学校介绍：测试学校介绍
预期结果	测试学校被创建
实际结果	与预期结果完全一致

表 6.2 学校的修改

测试名称	学校的修改
测试人	张鑫语
测试日期	2015-05-11
测试步骤	点击相关学校，之后点击详细信息
输入数据	修改学校名为: 测试学校修改，学校介绍: 测试学校介绍
预期结果	测试学校被修改
实际结果	与预期结果完全一致

6.2.2 班级的创建, 修改

表 6.3 班级的创建

测试名称	班级的创建
测试人	张鑫语
测试日期	2015-05-11
测试步骤	点击测试学校，之后点击右边 Panel 上的加号 在出现的创建 Panel 中输入班级数据
输入数据	班级名: 测试班级，班级介绍: 测试班级介绍
预期结果	测试班级被创建
实际结果	与预期结果完全一致

表 6.4 班级的修改

测试名称	班级的创建
测试人	张鑫语
测试日期	2015-05-11
测试步骤	点击测试学校，之后点击右边 Panel 上的加号 在出现的创建 Panel 中输入班级数据
输入数据	班级名: 测试班级，班级介绍: 测试班级介绍
预期结果	测试班级被创建
实际结果	与预期结果完全一致

6.2.3 网盘, 相册操作

表 6.5 网盘操作

测试名称	网盘操作
测试人	张鑫语
测试日期	2015-05-11
测试步骤	点击测试学校, 在右边 Panel 选择文件管理, 之后点击根文件夹 在右下边 Panel 点击新建, 新建成功后, 修改文件信息并提交 之后点击删除 (都在右下部 Panel)
输入数据	任意文件
预期结果	可以在网盘内新建, 修改, 删除文件
实际结果	与预期结果完全一致

表 6.6 相册操作

测试名称	相册操作
测试人	张鑫语
测试日期	2015-05-11
测试步骤	点击测试学校, 在右边 Panel 选择相册, 在右下边 Panel 点击上传 上传图片成功后, 选中图片并修改信息后提交 之后点击右下角红色按钮 : 删除选中图片
输入数据	任意图片
预期结果	可以在相册内新建, 修改, 删除图片
实际结果	与预期结果完全一致

6.2.4 教师, 学生的邀请, 转班, 删除

因为教师和学生管理界面和实现几乎相同, 所以下面统称人员。

表 6.7 人员的邀请

测试名称	人员的邀请
测试人	张鑫语
测试日期	2015-05-11
测试步骤	点击相关学校, 在右边班级列表中, 点击测试班级 再点击右边 Panel 中的教师管理, 之后点击右边 Panel 的邀请新教师 然后输入教师电话
输入数据	11 位数字
预期结果	教师进入该班级
实际结果	与预期结果完全一致

表 6.8 人员的转班

测试名称	人员的转班
测试人	张鑫语
测试日期	2015-05-11
测试步骤	点击相关学校，在右边班级列表中，点击测试班级 再点击右边 Panel 中的教师管理，在教师列表中选择一个教师 点击其右边的 Exchange Icon 按钮，然后选择转向的班级
输入数据	班级的选择
预期结果	教师从本班级移入目标班级
实际结果	与预期结果完全一致

表 6.9 人员的删除

测试名称	人员的删除
测试人	张鑫语
测试日期	2015-05-11
测试步骤	点击相关学校，在右边班级列表中，点击测试班级 再点击右边 Panel 中的教师管理，在教师列表中选择一个教师 点击其右边的垃圾箱按钮，然后点击确定
输入数据	人员的选择
预期结果	教师从本班级中移走
实际结果	与预期结果完全一致

6.2.5 帮手的邀请，修改，删除

表 6.10 帮手邀请

测试名称	帮手邀请
测试人	张鑫语
测试日期	2015-05-11
测试步骤	点击最左边导航栏中的“我的帮手”， 之后在主界面上部输入电话号码，之后点击“发出邀请”
输入数据	11 位数字
预期结果	下面帮手列表添加了新的帮手
实际结果	与预期结果完全一致

表 6.11 帮手修改和删除

测试名称	帮手修改和删除
测试人	张鑫语
测试日期	2015-05-11
测试步骤	点击最左边导航栏中的“我的帮手”，之后在主界面中部的帮手列表中，修改帮手权限之后点击提交，然后点击右上角红色垃圾箱按钮。
输入数据	权限的选择
预期结果	提交后显示帮手权限被修改，删除后该帮手不出现在列表中
实际结果	与预期结果完全一致

6.3 系统界面测试

需要测试的系统界面：

6.3.1 电脑端

- 学校修改界面
- 班级界面
- 班级列表界面
- 网盘界面
- 相册界面
- 教师，学生的管理界面
- 帮手界面

6.3.2 手机端

- 发布状态界面
- 浏览状态界面
- 聊天界面
- 我的班级和学校界面

6.3.3 测试结果

经由第五章的系统界面设计截图，可以得知系统界面符合设计要求，操作流畅，为底层功能提供了比较好的 UI 接口。具体截图不在这里冗余。

第七章 总结与展望

7.1 总结

本文介绍了一个幼儿教育领域应用的设计，开发流程和部分实现细节。从用户的需求，讲到功能上，架构上的设计，再讲到实现细节。在这个过程中，自己学到了非常多的东西，感觉意义非凡。

- 用户需求调研
- 产品设计
- UI 设计
- 如何优良架构一个基于 Nodejs 的大型 Web 服务
- 互联网开发团队如何运作
- 等等...

可惜的是，用户需求调研，设计流程并没有留下太多的文档。从设计的角度上说，其学习价值受到了减弱。但该应用对于如何实现管理 + 社交混合系统，有一定程度上的参考价值。严格意义上说，该应用可以被归类为一种垂直领域社交类应用，也是有效细分市场的体现。

7.2 展望

可以肯定地说，该应用有一定应用前景，现在自己还没有完全实现，而且后端和 Web 前端代码很多东西当时实现的时候由于经验不足，所以并没有一个比较好的结构，还需要进行大量的代码重构。所以未来要做的事情有：

- 实现手机端的 UI 交互
- 重构服务器后端代码
- 重构 Web 前端管理功能代码
- 基于新的后端，实现手机端功能

参考文献

- [1] 搜狐新闻客户端 UED 团队. 设计之下——搜狐新闻客户端的用户体验设计. 电子工业出版社, 2014.
- [2] 腾讯公司用户研究与体验设计部. 在你身边, 为你设计. 电子工业出版社, 2013.
- [3] Fette I. The WebSocket Protocol. Internet Engineering Task Force (IETF), 2011..
- [4] Leavitt N. Communicating and Displaying Real-Time Data with WebSocket. Internet Computing, IEEE, 2012, 16(4):45–53.
- [5] Leavitt N. Will NoSQL Databases Live Up to Their Promise? Computer, 2010, 43(2):12–14.
- [6] Chodorow K. MongoDB: The Definitive Guide. 2nd ed., O'Reilly Media, 2013.
- [7] Maxim B, Pressman R. Software engineering: a practitioner's approach. 8th ed., McGraw-Hill Science/Engineering/Math, 2014.
- [8] Mardan A. Practical Node.js: Building Real-World Scalable Web Apps. 1st ed., Apress, 2014.
- [9] Nedelcu C. Nginx HTTP Server Second Edition. 2nd ed., Packt Publishing, 2013.
- [10] Wilken J. Ionic in Action: Hybrid Mobile Apps with Ionic and AngularJS. 1st ed., Manning Publications, 2015.

致 谢

在中国科技大学完成本科的四年里，所有的学习和工作，都是在老师和同学的指导和帮助下进行的。在完成论文之际，请容许我对他们表达诚挚的谢意。

首先要感谢李斌老师在毕设项目组中的指导和在创新课程所教授的设计思想和创新理念，它们让我提升了一个新的台阶，极大地丰富了自己对于产品设计的理解，甚至对未来职业规划多了一份认识。也要感谢创新课程组的所有老师和同学。

之后要感谢袁平波老师，我第一次尝试 Web 开发是在他的数据库课程中，从课程设计的项目中对于 Web 开发收获颇丰，而且正是袁老师的数据库课程让自己喜爱上了编程。

另外感谢大一班主任匡森老师和现任班主任卢汉成老师多年的关怀。感谢师兄师姐们的指点和照顾，一起走过了这段愉快而难忘的岁月。

感谢科大，感谢一路走过来的兄弟姐妹们，在最宝贵年华里，是你们伴随着我的成长。

最后，感谢我家人一贯的鼓励和支持，你们是我追求学业的坚强后盾。

张鑫语

2015 年 5 月 25 日