**专业方向综合项目 项目方案书**

**张尧 陈东 姜其升**

**TEam 12**

**MEETING！**

基于微信小程序的协同日程安排应用

**目录**

[1 项目背景 2](#_Toc44771003)

[2 项目介绍 3](#_Toc44771004)

[3 项目的技术运用 4](#_Toc44771005)

[4 行业形势可行性分析 7](#_Toc44771006)

[4.1 技术可行性分析 7](#_Toc44771007)

[4.2 人员可行性分析 7](#_Toc44771008)

[4.3 市场可行性分析 7](#_Toc44771009)

[5 项目目标 9](#_Toc44771010)

[a) 识别重要的影响领域 9](#_Toc44771011)

[b) 识别项目的价值所在 9](#_Toc44771012)

[c) 定义合适的度量方法 9](#_Toc44771013)

[d) 为实现 MOV 设定事件框架 9](#_Toc44771014)

[开发目标 10](#_Toc44771015)

[直接目标 10](#_Toc44771016)

[6 项目中的术语解释 11](#_Toc44771017)

[事件、日程 11](#_Toc44771018)

[会议组、团队日历 11](#_Toc44771019)

[勿扰模式 11](#_Toc44771020)

[7 资源需求 12](#_Toc44771021)

[7.1 识别资源 12](#_Toc44771022)

[7.2 分配资源 12](#_Toc44771023)

[8 项目初步时间计划 14](#_Toc44771024)

[9 产品交付 16](#_Toc44771025)

[9.1 产出 16](#_Toc44771026)

[9.2 运行环境 16](#_Toc44771027)

[10 验收标准 17](#_Toc44771028)

# 1 项目背景

在众多需要多人参与的任务中，例如会议安排、团队协作、各种约会等等，在这些事件中我们可能都亲身体验过这样的一个痛点，事件参与成员的空闲时间不统一，就难以为一件共同事件安排计划。作为大学生的我们更能体会到这一点。我们每个人在课外都有着多多少少的项目，每个人上课时间不同，每个项目的团队成员不同，大家都有着各自的计划，而且很多时候项目都是同等重要的，所以有效的安排团队事件将很大程度上提升我们的工作效率，增加团队默契。

如今市面上存在着各式各样的日程管理软件，且几乎都为独立app的形式。其中大多数日程管理软件都只是专注与个人日程的管理，而忽略了团队事件与个人日程之间的冲突。当然也中不少日程管理应用功能全面，但从另一个角度来说这也增加了软件使用的复杂程度。而多数用户都会因为功能复杂和难以上手的原因放弃使用这样的日程管理应用。此外在团队事件都存在这样的一个特点，团队在一定的时间内是相对固定，而且多数的团队都会拥有各自的交流空间，而这其中大多数的都会选择微信群。所以这里存在着几个机会，Meeting系统在保留最基础的个人日程管理功能的基础上，增加对与团队事件的管理，对功做加法和减法，尽可能保证功能的必要性和使用的简单性。同时通过微信小程平台呈现将避免用户下载独立应用，降低产品接触目标用户的难度，与微信群之间建立良好互动。

# 2 项目介绍

“Meeting”是一款允许允许共享协作的日程安排小程序。

本系统将在提供日程管理的功能基础上，提供小组事件以及共享事件的管理。以小组中所有成员的日程信息为基本数据，为组长快速确定小组事件时间提供便捷。作为日程管理应用，Meeting允许个人向日程中添加事件管理事件。对于小组及共享事件管理模块，Meeting允许用户创建组并成为组长或以组长的权限删除组或以小组成员身份退出组。用户可以通过微信群中的邀请链接加入工作组。成为小组成员后可以查看组中其他成员的日程。小组长创建小组事件时可以获得基于小组成员日程信息的小组共有空闲时间推荐。此外，小组成员还可以针对事件设置隐私，以及对小组设置勿扰等。

本系统将业务范围集中在一周内的个人日程及小组事件管理上，尽最大可能将功能集中并考虑详尽，主要提供的核心功能为工作组中共享事件的自动推荐和创建、管理，辅以小组成员及权限的管理、个人事件的管理等功能。而将不会涉及到超过一周的事件管理、万年历查看功能等与核心功能无关的功能。

本系统的最终目标是提供一个能够便捷地通过微信小程序和微信群的良好联动生态，提供给用户管理个人短期内日程、便捷创建工作组并快速约定集体事件时间的应用，增加小组合作或活动策划的效率。

小组

近期日程管理

共享及小组事件管理

个人

“Meeting”



# 3 项目的技术运用

**项目后端采用Java的SpringBoot框架，前端使用微信小程序开发框架，基本关系型数据使用MySQL；**

**其他技术工具包括：**

**Redis**

Redis是一个开源的使用 ANSI C语言编写，支持网络，可基于内存也可持久化的日志型，Key-Value数据库，并提供了多种语言的 API ,相比 Memcached 它支持存储的类型相对更多 (字符，哈希，集合，有序集合，列表等)，同时Redis是线程安全的。

客户端连接 Redis 使用的是 TCP协议，直连的方式每次需要建立 TCP连接，而连接池的方式是可以预先初始化好客户端连接，所以每次只需要从 连接池借用即可，而借用和归还操作是在本地进行的，只有少量的并发同步开销，远远小于新建TCP连接的开销。另外，直连的方式无法限制 redis客户端对象的个数，在极端情况下可能会造成连接泄漏，而连接池的形式可以有效的保护和控制资源的使用。

Jedis 在实现上是直连 redis server，多线程环境下非线程安全，除非使用连接池，为每个 redis实例增加 物理连接。

我们的项目使用Redis来支持微信用户登录服务相关。当新用户登录时，后端会向微信发送请求以获取该用户的“openid”、“session\_key”和后端将分配一个新的“userid”，这是我们在关系数据库中对用户的主键。这三个条目都存储在Redis作为后端uuid生成的密钥的值。下次用户登录时，微信小程序会向微信发送请求，查看其会话密钥是否过期。如果不是，小程序告诉后端使用现有的条目。然后小程序通信从上一次使用uuid的后端，后端将检查相应的“userid”告诉哪个用户在线。

**MyBatis**

MyBatis 是支持普通 SQL查询，存储过程和高级映射的优秀持久层框架。MyBatis 消除了几乎所有的JDBC代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的 XML或注解用于配置和原始映射，将接口和 Java 的POJOs（Plain Ordinary Java Objects，普通的 Java对象）映射成数据库中的记录。

MyBatis是半自动的ORM框架，原来叫ibatis， 10年apache软件基金组织把它托管给了goole code，就重新命名了MyBatis，功能相对以前更强大了。它相对全自动的持久层框架Hibernate，更加灵活，更轻量级，这点我还是深有体会的。MyBatis的一个强大特性之一就是动态SQL能力了，能省去我们很多串联判断拼接SQL的痛苦，根据项目而定，在一定的场合下使用，能大大减少程序的代码量和复杂程度，不过还是不是过度太过复杂的使用，以免不利于后期的维护和扩展。

**HikariCP连接池**

HikariCP是由日本程序员开源的一个数据库连接池组件，代码非常轻量，并且速度非常的快。根据官方提供的数据，在i7,开启32个线程32个连接的情况下，进行随机数据库读写操作，HikariCP的速度是现在常用的C3P0数据库连接池的数百倍。在SpringBoot2.0中，官方也是推荐使用HikariCP。优势是

1.字节码更加精简，所以可以加载更多代码到缓存。

2.实现了一个无锁的集合类型，来减少并发造成的资源竞争。

3.使用了自定义的数组类型，相对与ArrayList极大地提升了性能。

4.针对CPU的时间片算法进行优化，尽可能在一个时间片里面完成各种操作。

**Lettuce**

Lettuce是一个高性能基于Java编写的Redis驱动框架，底层集成了Project Reactor提供天然的反应式编程，是 一种可伸缩，线程安全，完全非阻塞的Redis客户端，多个线程可以共享一个RedisConnection,它利用Netty NIO 框架来高效地管理多个连接，从而提供了异步和同步数据访问方式，用于构建非阻塞的反应性应用程序。通信框架集成了Netty使用了非阻塞IO，5.x版本之后融合了JDK1.8的异步编程特性，在保证高性能的同时提供了十分丰富易用的API，

支持Redis的新增命令ZPOPMIN, ZPOPMAX, BZPOPMIN, BZPOPMAX。

支持通过Brave模块跟踪Redis命令执行。

支持Redis Streams。

支持异步的主从连接。

支持异步连接池。

新增命令最多执行一次模式（禁止自动重连）。

全局命令超时设置（对异步和反应式命令也有效）。

**Apache Shiro**

Apache Shiro 是一个功能强大、灵活的，开源的安全框架。它可以干净利落地处理身份验证、授权、企业会话管理和加密。

实际上，它实现了管理应用程序安全性的所有方面，同时尽可能避免出现问题。它建立在完善的接口驱动设计和面向对象的原则之上，可以在任何你想象得到的地方实现自定义行为。但是，对于所有事情来说，默认情况下都是合理的，这与应用程序安全性是一样的。至少这是我们所追求的。

我们 POST 用户名与密码到 /login 进行登入，如果成功返回一个sessionKey，失败的话直接返回 401 错误。

之后用户访问每一个需要权限的网址请求必须在 header 中添加 Authorization 字段，例如 Authorization: token ，token 为密钥。

后台会进行 token 的校验，如果有误会直接返回 401。

**微信小程序**

微信小程序是一种不用下载就能使用的应用，也是一项创新，经过将近两年的发展，已经构造了新的微信小程序开发环境和开发者生态。微信小程序也是这么多年来中国IT行业里一个真正能够影响到普通程序员的创新成果，已经有超过150万的开发者加入到了微信小程序的开发，与我们一起共同发力推动微信小程序的发展，微信小程序应用数量超过了一百万，覆盖200多个细分的行业，日活用户达到两个亿，微信小程序还在许多城市实现了支持地铁、公交服务。微信小程序发展带来更多的就业机会，2017年小程序带动就业104万人，社会效应不断提升

本项目采用微信小程序的客户端形式，充分利用跨平台一次开发的遍历，也是让我们在同类产品中具有更多可得性的地方。

# 4 行业形势可行性分析

## 4.1 技术可行性分析

小程序是一种新的开放能力，开发者可以快速地开发一个小程序。小程序可以在微信内 被便捷地获取和传播，同时具有出色的使用体验。微信小程序隶属于 Web 开发的范畴，相较于独立的应用，技术上手难度较低，适合经验较少的开发人员学习和使用。并且，微信小 程序提供了丰富的框架组件和 API 接口，包括：界面、视图、内容、按钮、导航、多媒体、位置、数据和网络等，为小白开发者提供开发文档、开发者工具、设计指南等一系列工具， 帮助开发者快速接入并完成小程序开发。此外，依托微信小程序平台，一次开发即可同时在Android 和 ios 设备上使用，将在很大程度上降低开发成本。本系统拟使用微信小程序作为前端平台，采用 SpringBoot 框架作为后端服务架构。在数据库方面，我们拟使用 Redis 存储用户有关信息，进行鉴权相关操作，使用 MySQL 进行其他基本用户数据的存储。

技术为项目带来的优势

依托微信广大的用户群体，小程序可以在微信内被便捷的获取和传播，在一定程度上降低产品与用户的接触难度。一次开发即可同时在 Android 和 ios 设备上使用，将在很大程度上降低开发成本。通过微信小程序与微信群的良好生态联动，将成为该项目的一个重要竞争点。日程管理应用具有单次停留时间较短，使用频次较高的特点，使用额外安装独立应用将增加使用的复杂度，。由此来看，微信小程序是更好的解决方案。

## 4.2 人员可行性分析

团队成员具有较为丰富软件丰富的软件项目开发经验以及成功的项目合作经历。

团队成员张尧、陈东、姜其升曾共同参与过基于 Web 生活分享平台“WeLog”的开发， 其中张尧主要负责前端，陈东、姜其升主要负责后端开发以及数据库配置。所以三人有较为完整的前后端开发经验以及较为成功的合作经历。三人都具有较为系统全面的软件开发与管理开发知识。在此次项目中陈东和姜其升均担任后端开发之责，此外陈东在本项目中还担任产品经理，姜其升还担任架构设计。张尧拥有丰富的前端开发经验，在多个项目中担任前端开发之职，具有较强的前端开发能力。于本项目中担任客户代表职责以及前端的开发。

上述三人还组成了项目管理部门，除此之外团队中还有分析工程师、测试工程师与数据库工程师，均有丰富的理论知识基础和实践经验。

## 4.3 市场可行性分析

**政治环境**

《国务院关于积极推进“互联网+”行动的指导意见》中提到，要充分发挥互联网的创新驱动作用，以促进创业创新为重点，推动各类要素资源聚集、开放和共享，大力发展众创空间、开放式创新等，引导和推动全社会形成大众创业、万众创新的浓厚氛围，打造经济发展新引擎。

**市场环境**

微信在在校生中的普及率极高，借助微信平台积累起的用户量与微信的社交分享属性， 可以大大降低产品接触目标用户的难度。此外本项目相比其余同类产品具有独特的竞争优势。

**法律因素**

本次开发是在学习中的一次开发，所使用软件都可以从网上下载，个人开发者具有免费的开发使用权，且本项目暂时不考虑用作商业用途，并无侵权或者抄袭等违法情况，也没有被申请过专利，所以，在法律方面是可行的。

**用户使用可行性**

所开发软件操作应该尽可能简单，以便满足不同的人们都可以很方便的操作使用。同时用户群体中对日程管理类应用具有简单以及团队事件管理的需求。

# 5 项目目标

Meeting 系统的目标是提供一个能够便捷地通过微信小程序和微信群的良好联动生态， 提供给用户管理个人短期内日程、便捷创建工作组并快速约定集体事件时间的应用，增加小 组合作或活动策划的效率。

## a) 识别重要的影响领域

|  |  |
| --- | --- |
| **潜在领域** | **影响事件** |
| 战略 | 增大知晓度 |
| 用户 | 更好的产品和更佳的使用体验 |
| 运作 | 提高效率 |
| 财务 | 降低成本 |

## b) 识别项目的价值所在

提供给用户管理个人短期内日程、便捷创建工作组并快速约定集体事件时间的应用，简化日程管理类应用的使用复杂性，增加小组合作或活动策划的效率，提升用户的使用体验。

## c) 定义合适的度量方法

项目计划首先于同济大学校内试运营，因此计划第一年达成校内累计用户超过 10000 人， 日 PV(单日访问人数)平均超过 1000 人，校内用户知晓度达 30%以上。

第二年预期支持校外用户的使用，预期达成校外累计用户超过 50000 人，日 PV(单日访问人数)平均超过 1250 人，校外用户知晓度达 5%以上。

## d) 为实现 MOV 设定事件框架

|  |  |
| --- | --- |
| **年** | **MOV** |
| 1 | 校内试运营，一年内累计用户超过 10000 人，日 PV(单日访问人数)平均超过 1000  人 |
| 2 | 支持校外用户使用，预期达成校外累计用户超过 50000 人，日 PV(单日访问人数)平  均超过 1250 人 |

## 开发目标

项目当前大周期为 1 年，于 2020 年 5 月 14 日起。在当前大周期结束前，预期完成“开会！MEETING！”前后端主体框架开发(如用户注册、登录等功能)， 并完成包括但不限于个人事件的添加与管理、工作组的创建及退出、邀请自动加入同一个工作组、查看工作组中他人日程、自动推荐公共事件时间、勿扰模式等功能模块的开发，以及商务合作、广告等营收方式的技术开发。该部分所提到的开发涉及前端开发和后端开发两部分，均需在 Scrum 开发的每个 Sprint 中配合完成。

## 直接目标

项目直接目标分为 2 个部分：同济大学校内用户数量与知晓度以及校外用户数量与知晓度。

项目计划首先于同济大学校内试运营，因此计划在当前大周期结束前，预期达成校内累计用户超过 20000 人，日 PV(单日访问人数)平均超过 500 人，日 UV(单日访问次数)平均超过 1000 次，校内用户知晓度达 50%以上。

就校外用户而言，项目大周期结束前，预期支持校外用户的使用，预期达成校外累计用户超过 50000 人，日 PV(单日访问人数)平均超过 1250 人，日 UV(单日访问次数)平均超过

2500 次，校外用户知晓度达 25%以上。

# 6 项目中的术语解释

## 事件、日程

这两个名词含义相同，在本系统内，可以指用户自己添加的一天天排定的程序，每一件事情就是一个事件或者日程，可以在程序中设定这一件事情标题、内容、开始和结束时间。也可以是会议组中团队设定的一件大家都需要来参与的事情，比如开会，当然也可以设定这一件事情标题、内容、开始和结束时间。

## 会议组、团队日历

这两个名词含义相同，在本系统内，可以指一群用户能共同查看的一些事件的集合。对应于我们生活中部门、课程项目组等等概念。

## 勿扰模式

当用户参与在会议组中，但由于有私事无法参加今日的任何时间段的事件，但又想让会议组忽略自己的时间不可用性继续开会执行事件，则可以打开勿扰模式。

# 7 资源需求

## 7.1 识别资源

|  |  |
| --- | --- |
| **劳动力** | **产品经理、前端开发人员、后端开发人员、数据库管理员、系统分析师、系**  **统设计师** |
| **设备** | 个人电脑、服务器 |
| **空间** | 图书馆一楼、交运学院自习区域 |
| **时间** | 大把时间 |
| **资金** | 租服务器的钱 |

## 7.2 分配资源

针对劳动力资源的分配紧张的问题，专门分析劳动力资源的需求。

|  |  |  |
| --- | --- | --- |
| **1 项目前期** | | |
| **团队建立** | 11 | 项目经理 |
| **举办启动会议** | 2 | 项目经理 |
| **识别利益相关者** | 4 | 项目经理 |
| **收集市场需求** | 11 | 项目经理、系统分析师 |
| **收集用户需求** | 6 | 项目经理、系统分析师 |
| **里程碑** | 0 |  |

**2 项目策划**

|  |  |  |
| --- | --- | --- |
| **定义项目目的** | 5 | 项目经理 |
| **确定所需资源** | 3 | 项目经理 |
| **确定项目风险** | 3 | 项目经理 |
| **创建项目 WBS** | 3 | 项目经理 |
| **确定任务** | 5 | 项目经理、系统分析师、系统设计师 |
| **确定任务周期** | 3 | 项目经理 |
| **确定任务依赖** | 5 | 项目经理 |
| **创建甘特图** | 8 | 项目经理 |
| **复审甘特图** | 3 | 项目经理、系统分析师、系统设计师 |
| **里程碑** | 0 |  |
| **3 建 模** | | |
| **用例分析** | 11 | 系统分析师 |
| **需求确定** | 2 | 系统分析师 |
| **界面原型设计** | 14 | 前端工程师、系统分析师 |
| **接口设计** | 8 | 项目经理、前端工程师、后端工程师 |
| **Calendar 模块概要设计** | 6 | 系统设计师 |
| **User 模块概要设计** | 6 | 系统设计师 |
| **Event 模块概要设计** | 13 | 系统设计师 |
| **数据库设计** | 11 | 数据库管理员、系统设计师 |
| **概要设计文档** | 4 | 系统设计师 |
| **Calendar 模块详细设计** | 11 | 系统设计师 |
| **User 模块详细设计** | 11 | 系统设计师 |
| **Event 模块详细设计** | 16 | 系统设计师 |
| **详细设计文档** | 4 | 系统设计师 |
| **里程碑** | 0 |  |
| **4 构 建** | | |
| **服务器搭建** | 2 | 后端工程师、前端工程师 |
| **数据库建立** | 2 | 数据库管理员 |
| **日历界面构建** | 16 | 前端工程师 |
| **会一组成员界面构建** | 16 | 前端工程师 |
| **事件详情界面构建** | 16 | 前端工程师 |
| **User 模块编码** | 16 | 后端工程师 |
| **Calendar 模块编码** | 16 | 后端工程师 |
| **Event 模块编码** | 16 | 后端工程师 |
| **里程碑** | 0 |  |
| **5 测 试** | | |
| **单元测试** | 11 | 测试员、前端工程师、后端工程师 |
| **集成测试** | 11 | 测试员、后端工程师 |
| **系统测试** | 11 | 测试员、前端工程师、后端工程师 |
| **验收测试** | 11 | 测试员、项目经理 |
| **测试文档** | 3 | 测试员 |
| **里程碑** |  |  |
| **6 项目上线** | | |
| **用户帮助文档** | 11 | 项目经理、前端工程师、后端工程师 |
| **处理用户问题** | 4 | 项目经理、前端工程师、后端工程师 |
| **里程碑** | 0 |  |

# 8 项目初步时间计划

本项目自2020年5月30日开始，最晚于2020年6月27日结束，应尽早完成。项目中产生的交付产物为：需求规约、需求分析规约、系统分析、系统设计、系统测试文档、项目程序客户端和服务器端、用户帮助文档等。项目过程中的里程碑有：项目前期里程碑、项目策划里程碑、建模里程碑、构建里程碑、测试里程碑、上线里程碑。里程碑象征着一阶段的完成，有产出的交付文档或者程序，但在实际中，为了加快进度，一个阶段还未结束，可能另一阶段的任务已经开始并行着手解决。

|  |  |  |
| --- | --- | --- |
| 任务 | 截止日期 | 负责人员 |
| 系统分析与设计 |  |  |
| 数据库设计 | 5月30日 | 陈东 |
| 登陆注册微信接口模块分析设计 | 5月30日 | 姜其升 |
| 事件模块分析设计 | 5月30日 | 张尧 |
| 日历（会议组）模块分析设计 | 5月30日 | 张尧 |
|  |  |  |
|  |  |  |
| 服务器配置 | 5月31日 | 陈东 |
| 数据库部署 | 5月31日 | 陈东 |
|  |  |  |
| 登录注册微信接口模块 |  |  |
| 用户账户 | 6月10日 | 姜其升 |
| 微信方面用户安全认证 | 6月10日 | 姜其升 |
|  |  |  |
| 事件模块 |  |  |
| 创建个人事件 | 6月6日 | 陈东 |
| 修改个人事件 | 6月7日 | 陈东 |
| 删除个人事件 | 6月7日 | 陈东 |
| 事件创建修改显示前端界面 | 6月10日 | 张尧 |
|  |  |  |
| 日历（会议组）模块 |  |  |
| 用户查看会议组成员事件 | 6月20日 | 陈东 |
| 创建团队事件 | 6月10日 | 陈东 |
| 修改团队事件 | 6月13日 | 陈东 |
| 创建团队日历 | 6月12日 | 姜其升 |
| 解散团队日历 | 6月14日 | 姜其升 |
| 分享团队日历 | 6月16日 | 姜其升 |
| 加入团队日历 | 6月12日 | 姜其升 |
| 退出团队日历 | 6月23日 | 陈东 |
| 设置勿扰模式 | 6月20日 | 姜其升 |
| 推荐组内时间 | 6月23日 | 姜其升 |
| 日历（会议组）显示界面 | 6月17日 | 张尧 |
| 我的主页显示界面 | 6月23日 | 张尧 |
|  |  |  |
|  |  |  |
| 整体调试 | 6月25日 |  |
| 部署 | 6月27日 |  |

# 9 产品交付

## 9.1 产出

程序代码

• 微信前端代码

• 数据库表和字段SQL语句

• 后端Java代码

• 数据库模式图

项目过程资源

会议记录

项目过程文件等

文档

– 项目方案书

– 需求分析

– 系统设计说明书

– 编程规范

– 开发计划

– 测试计划

– 测试报告

– 操作说明书

## 9.2 运行环境

1. 服务器端

• 操作系统：CentOS 7

• Java 8版本

2. 小程序客户端

• 操作系统：所有支持微信最新版的操作系统，比如Android，iOS，Windows或者MacOS

根据实测，需要将微信客户端手机版升级至7.0.2以上，微信桌面客户端升级至2.9.2以上。

# 10 验收标准

经用户和开发小组双方确认的“需求规格说明书”。

重点确认软件的可靠性、易使用性和功能完整性。