

目录

Quantime需求文档	3
文档说明.....	3
输出环境	3
产品概述.....	3
产品介绍	3
产品定位	3
产品特点	3
产品Logo.....	3
产品背景.....	4
背景概述	4
用户画像	8
需求小结	8
产品结构.....	10
功能说明.....	10
页面逻辑.....	12
交互设计的原则与思路.....	13
小程序后端云开发文档.....	14
数据库设计.....	14
实体集.....	14
关系集.....	16
云函数API开发.....	16
云开发过程中的重点.....	18
lookup联表查询.....	18
getUserProfile	18
npm module: wl-pinyin	18
前后端联调.....	19
产品运营、推广与未来发展方向.....	20
运营方向	20
推广方向	21
未来开发方向.....	21
Quantime前端文档	22
文档说明.....	22
单页面说明.....	22
时间统计&表单信息.....	22

我的信息&类管理	24
新建表单	25
填写表单	27
统计分析	28
其他页面展示	29
页面交互逻辑	30
统计者	30
参与者	30
前端核心构思和实现	32

Quantime需求文档

文档说明

输出环境

小程序版本	V1.3.5
开发者工具版本	Stable v1.05.2103200
撰写人	汪奕晨、黄余飞、徐瑞柏、王青蕊
撰写时间	2021.5.29

产品概述

产品介绍

Quantime是由**浙江大学 卓月团队**在2021年开放的一款时间统计小程序，支持自动生成可供选择的时间区间选项，给出可视化的快捷统计结果。

Quantime通过 Quantum time（时间量子）的物理学概念打造核心产品逻辑，塑造全新的移动交互时间统计哲学。

产品定位

Quantime是一款简单易用、轻量高效的时间统计微信小程序，帮助用户快速发布时间统计表单，即时查看准确清晰的统计结果，满足不同场景下用户协同开展时间安排的需求。

产品特点

- 程序自动生成时间量子概念（按小时粒度）的可选时间选项，减少表单发布负担。
- 时间量子颗粒选项匹配时间轴，简单直观，减少表单填写负担。
- 时间统计结果支持可视化热图（colorbar）显示，整体情况一眼纵览。
- 统计结果详情支持查看带首字母的具体时间参与者列表展示，交互方便。

产品Logo



Logo中的小圆圈代表时间量子。

由外向内，圆圈逐渐聚拢、扩大、相接，与小程序功能相呼应，意为成功找到不同用户碎片时间的交集，即时间统计功能。

产品背景

背景概述

微信10年，依托微信的社交私域流量，微信小程序已经成为一种新的应用形态，表单统计分发场景时常见于微信之中，其中，有一类统计需求：时间统计，在当今疫情加速高效线上协同管理办公，微信小程序应用场景进一步拓宽的背景下，却没有得到应有的重视，旧有方案难以适应移动端交互形态，拖累协作效率。

调研分析

聚焦时间统计这一需求，我们在朴素的“打造移动端交互方便的时间统计小程序”这一想法基础上，开展前期用户访谈以进行需求挖掘，得到需求列表后制作访谈问卷分析核心需求，分析结果如下：

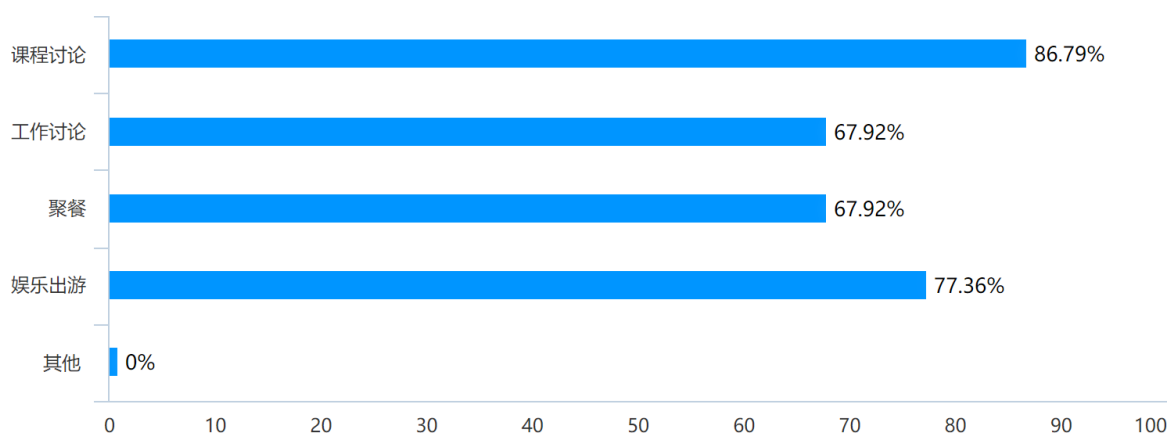
问卷基本情况说明

本次问卷调查共收集到有效问卷106份，年龄段集中在18-22岁（100%），职业以全日制学生为主（98.11%）。

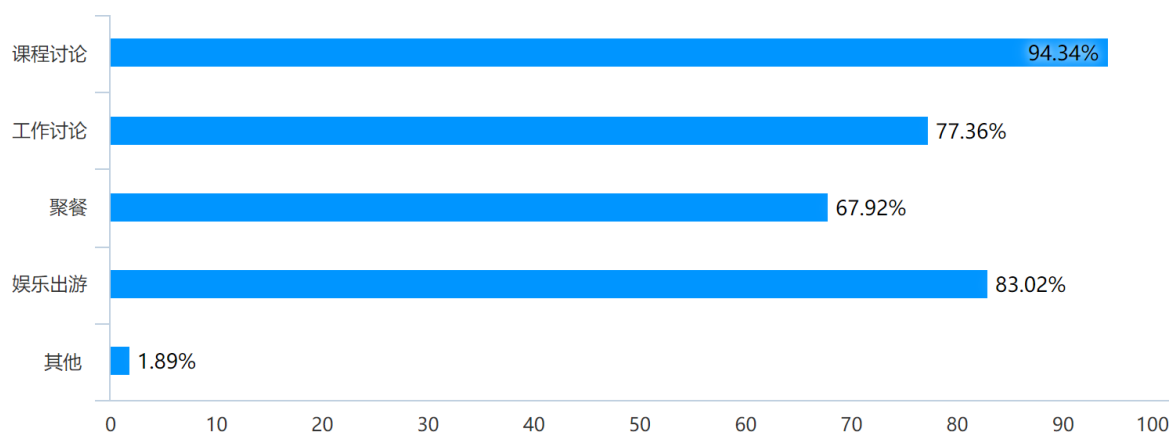
时间统计需求验证

1. 时间统计需求场景广泛

时间统计需求场景涉及广泛，包含课程讨论、工作讨论、聚餐、娱乐出游等，且场景覆盖率极高，上述场景发起时间统计的需求覆盖率均达到**60%**以上，其中，课程讨论的发起场景覆盖率达**86%**以上，参与覆盖率达**94%**以上。



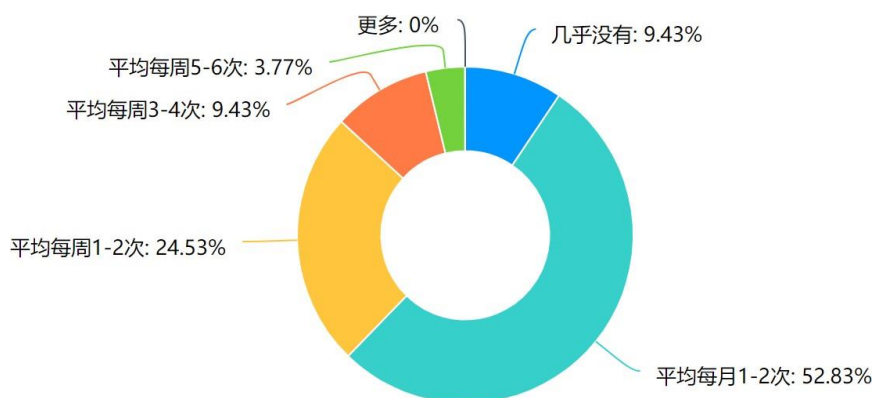
您日常有哪些 **发起** 时间统计的场景？



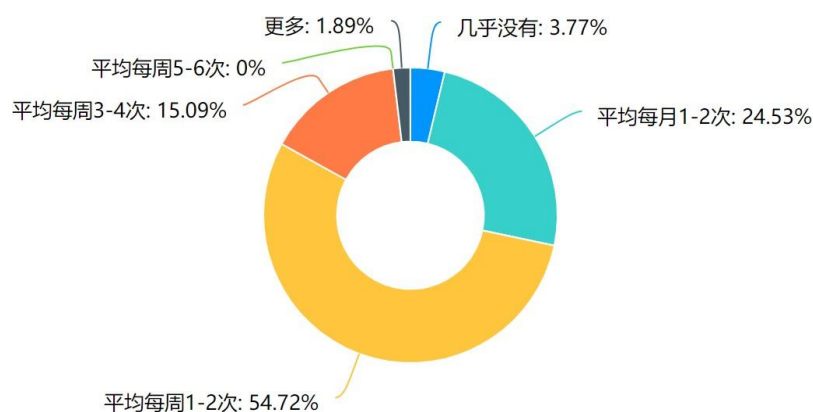
您日常有哪些 参与 时间统计的场景？

2. 时间统计需求频率可观

时间统计需求具备相当可观的频率，过半数填写者表示平均每月发起1-2次时间统计，平均每周参与1-2次时间统计，更有一部分填写者具有平均每周发起1-2次或更多时间统计，平均每周参与3-4次时间统计的需求，有望转化为产品的铁杆用户。



您平常 发起 时间统计的频率是？



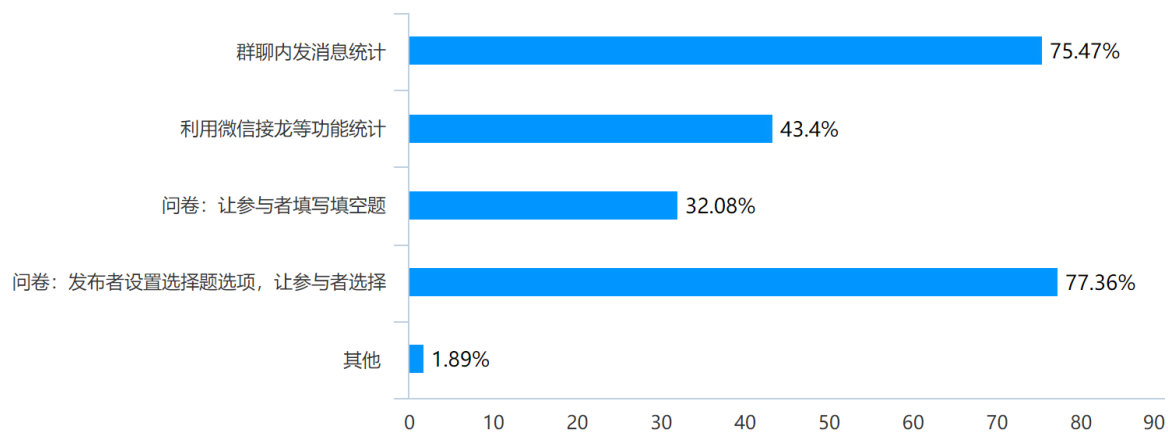
您平常 参与 时间统计的频率是？

时间统计需求挖掘

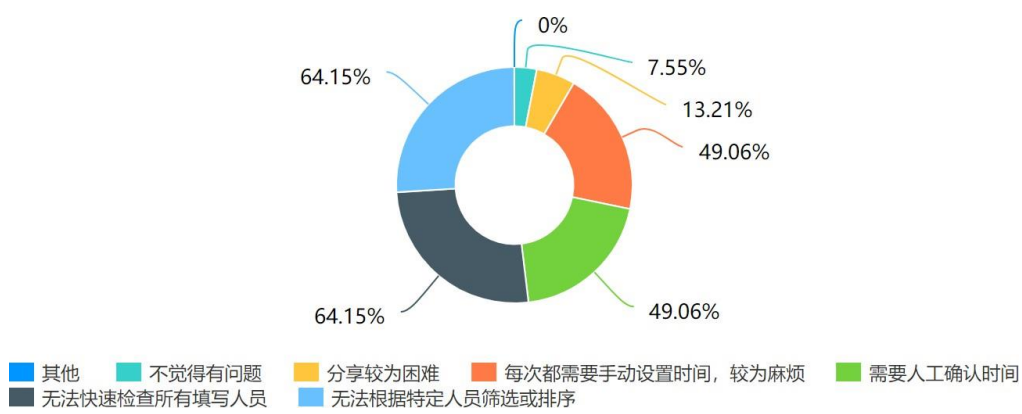
我们首先调查受访者的之前使用时间统计形式和他们觉得可能存在的问题。

时间统计形式以“群聊内发消息统计”和“问卷：发布者设置选择题选项，让参与者选择”为主，均占到75%以上的比例。

而在调查现有的时间统计形式可能存在的问题时，仅有不到8%的受访者表示“不觉得有问题”，而“无法快速检查所有填写人员”、“无法根据特定人员筛选或排序”、“每次都需要手动设置时间，较为麻烦”、“需要人工确认时间”等都是现有方式的显著痛点。



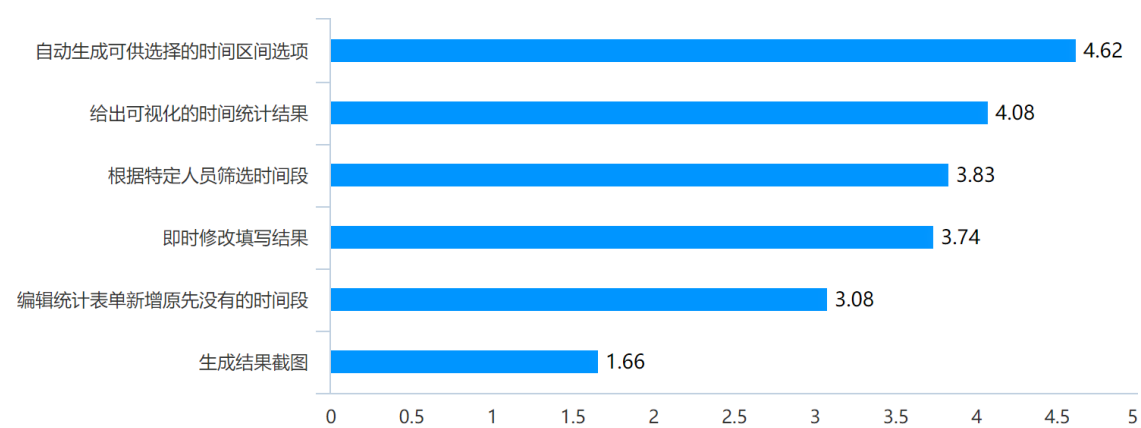
平常用过的时间统计形式有？



您觉得现有的时间统计形式可能有的问题是？

时间统计需求分析

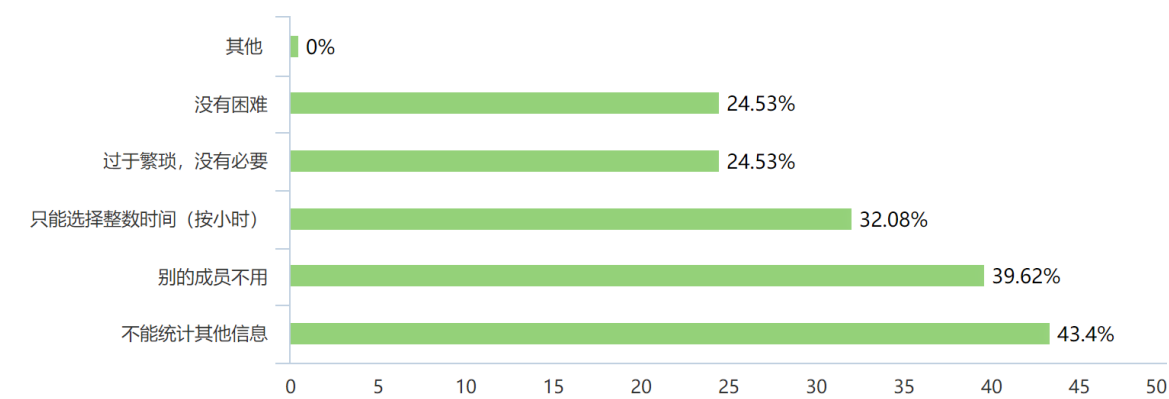
最后，我们根据筛选出的需求列表，邀请问卷填写者为需求进行排序：



请您为下列时间统计的需求排序
(计算得分越高越优先)

其他需求分析

在我们所列的核心需求列表外，我们还邀请受访者填写可能遇到的困难和为一些需求的必要性打分，来 帮助我们筛选需求列表，确保没有重要需求遗漏在外：



如果有一款小程序可以：.....您觉得在使用它的过程中可能会遇到什么困难？

请您为下列功能的必要性打分

功能	打分平均值（满分5分）
通过我的界面查看填写过的表单	4.43
拥有设置不可用时间段，和新增确定统计时间段的日程功能	4.11
允许在填写时为具体的时间区间点增加标签	4.02
允许全天的时间段选择	3.96
为常用的人员组建立群组，每次问卷可以直接在群组内发布	3.87
精确到分钟的连续化时间区间选项	3.4

用户画像

基于前期用户访谈和调查问卷的结果，我们总结提炼了Quantime的典型用户画像如下：

- 班团干部勇当组长 —— 小林
 - ◆ **形象：**具备领导力，勇于站出来的小林，经常主动担任各种课程小组的组长，也是班长和某组织的管理者。
 - ◆ **场景：**每次需要统计组员，部门成员时间手动设置时间选项都非常麻烦，并且需要人工下载结果，肉眼分析。
 - ◆ **需求：**希望能减少发布和开展时间统计的精力和时间成本，发布时避免手动设置时间区间选项，统计结果时能够一眼看到直观结果，拯救选择困难症。
- 聚餐出游娱乐达人 —— 小黄
 - ◆ **形象：**喜欢和朋友聚餐出游的小黄，平时想见缝插针和同学室友聚餐娱乐，每逢周末节假日也要和朋友们或近或远的出行游玩。
 - ◆ **场景：**希望能够快速统计和朋友的共有空闲时间，充分利用朋友之间的碎片化时间，找到大家空闲时间意想不到的交集，而不用再发送很多条聊天消息反复确认，减少约时间的负担，让聚餐出游更尽兴。
 - ◆ **需求：**方便快捷的时间统计产品，节省发送多条聊天消息的时间和精力，同时能够找到足够精细共有时间，提高找到共同时间的概率。
- 学习工作四处奔忙 —— 小杨
 - ◆ **形象：**学习勤奋，工作认真的小杨，奔波于浙江大学紫金港、玉泉两个校区之间，经常需要参加各种课程小组讨论、科研组会、组织工作小会等。
 - ◆ **场景：**各种活动，每次组长或组织者统计时间时都是问卷，问卷时间区间太大日程安排困难，填写填空题又担心让组织者为难，并且一个个看文字选时间非常麻烦。
 - ◆ **需求：**希望有符合直观感受的时间选择的交互，能够方便地填写问卷，同时可供选择的时间粒度精细，提高时间利用率。

需求小结

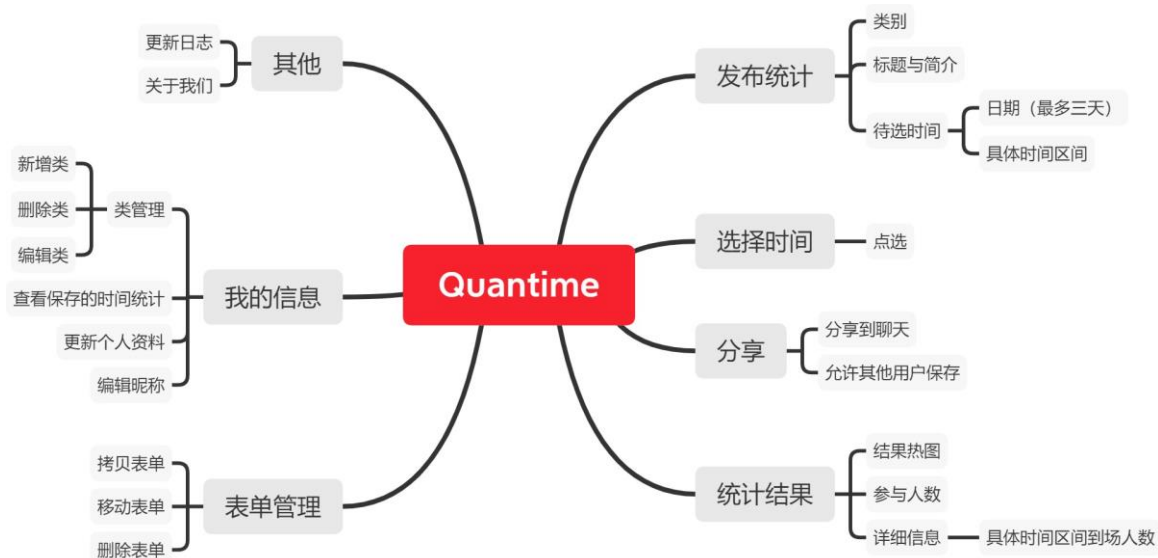
基于对时间统计的需求挖掘和痛点分析，并结合排序等方式进行需求强度验证，我们概括了**2**大核心需求：**希望能够降低发布和填写时间统计的成本，避免手动设置的时间区间选项，希望能够有直观清晰、一眼总览的统计结果。**

具体剖析2大核心需求，我们发现它们具体可以分为很多小的需求点，对应地，我们在小程序设计中对症下药打造了相应的功能。

以下是具体的需求列表，序号从前往后表示优先级从高到低。

序号	优先级	问题需求	对应功能
1	五星	希望减少发布时间统计时手动设置时间区间选项的精力和时间成本	程序自动提供时间量子概念（按小时粒度）的可选时间选项
2	五星	希望能够一眼总览整体的时间统计情况，迅速找出参与统计者可用时间的重叠	时间统计总览页面的可视化 colorbar 显示（时间块颜色越深表示在该时间段有空的人越多）
3	五星	希望能够快速确定具体时间段的参与人员	时间统计详情页面：带 colorbar 的滑动时间条和带索引的具体时间参与者列表展示
4	五星	希望能够快捷分享表单，方便填写	支持将时间统计表单快速分享到微信聊天
5	四星	希望能够直观、清晰地表达自己的空余时间，充分利用碎片时间	匹配时间轴的时间量子概念（按小时粒度）的可选时间选项，支持允许参与者保存时间统计查看统计结果
6	四星	希望能够对时间统计的要求和情况等信息进行定制化	新建表单时要求用户填写标题和类别，供统计发布者说明情况和表达约束
7	三星	希望能够对时间统计的具体区间进行限制	支持时间量子锁定不可选
8	三星	希望能够对日历上不连续的日期进行时间统计	新建表单时允许用户随机选择日期开展统计（目前最多允许统计 3 天）
9	三星	希望能够有序地管理不同的时间统计	包含统计分类的二级设计，支持分类的新建和分类名编辑，支持表单新建时分类和对移动已有的表单
10	三星	希望能够快速根据昵称辨识参与统计的人员	参与人员展示和时间统计详情页面均支持带首字母的具体时间参与者列表展示，同时支持昵称的修改
11	二星	希望能够快速复用先前的时间统计	支持时间统计表单的拷贝

产品结构



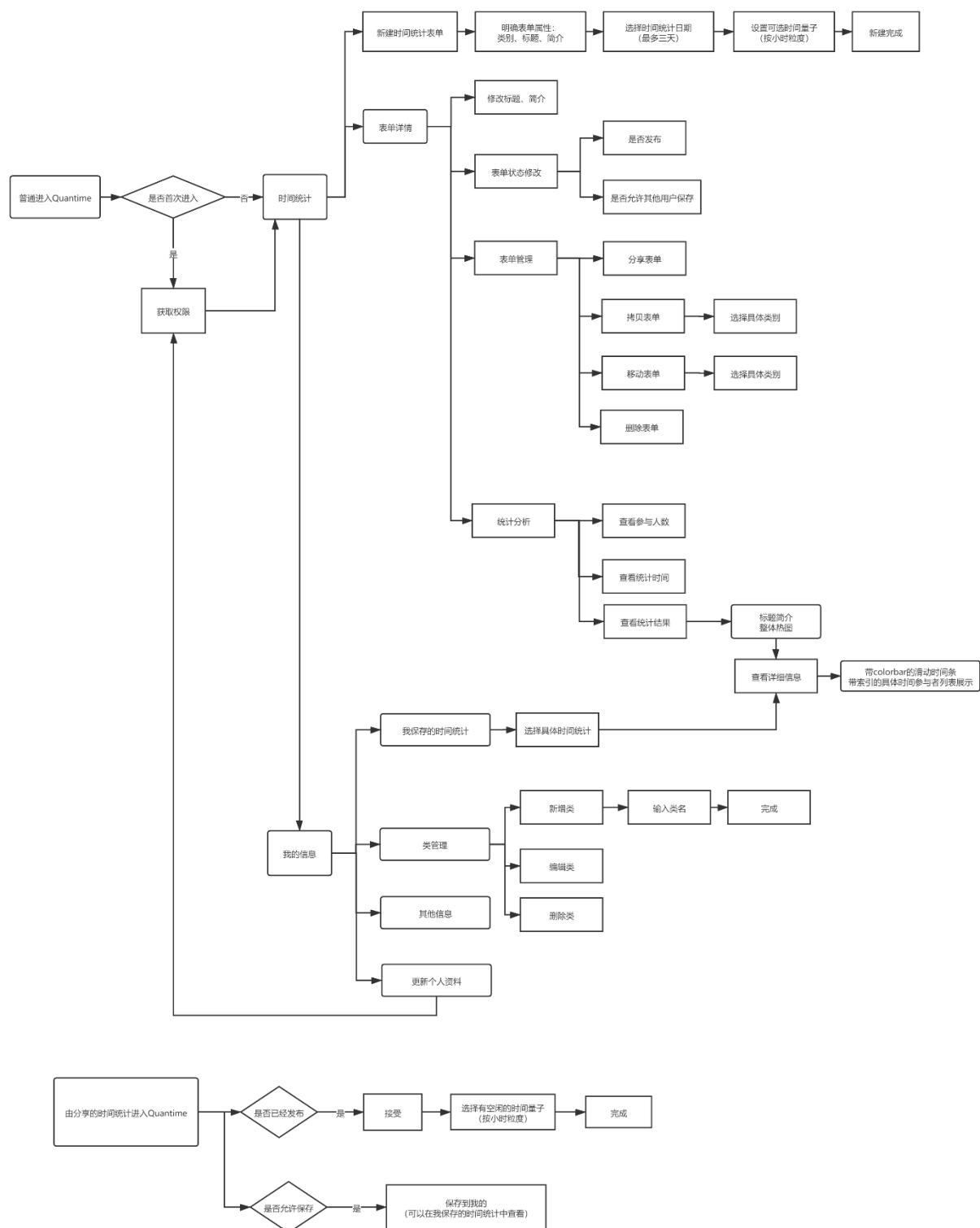
功能说明

时间统计：

- 发布时间统计表单
 - 明确时间统计表单属性
 - 类别
 - 标题
 - 简介
 - 明确统计时间段
 - 明确统计日期（最多三天）
 - 明确可选时间量子（按小时粒度）
- 操作时间统计表单
 - 修改标题简介
 - 修改发布状态（允许/暂停填写）
 - 修改是否允许保存（允许/禁止其他用户查看统计结果）
 - 表单管理
 - 分享统计
 - 分享到群聊
 - 拷贝表单
 - 移动表单
 - 删除表单
- 填写者选择有空闲的时间量子（按小时粒度）
 - 点选

- 统计结果分析
 - 可视化整体热图（colorbar）显示（时间块颜色越深表示在该时间段有空的人越多）
 - 具体时间段统计分析
 - 带热图（colorbar）的滑动时间条
 - 带昵称索引的具体时间参与者列表展示
- 用户交互功能
 - 搜索类别、表单
 - 管理类别
 - 新建类别
 - 编辑类别
 - 删除类别
 - 更新信息
 - 更改昵称
 - 重新获取权限
 - 查看保存的时间统计结果
 - 其他信息
 - 更新日志
 - 用户反馈
 - 关于我们

页面逻辑



交互设计的原则与思路

- 基于产品核心逻辑设计UI和交互

在进行产品设计和开发时，我们始终不忘从需求出发，牢记Quantime的产品核心逻辑是**帮助用户快速发布时间统计表单，即时查看准确清晰的统计结果。**

- 在易学性和交互效率之间寻找平衡

对于Quantime采用的移动端全新的时间统计交互，我们力求在易学性和交互效率之间寻找平衡

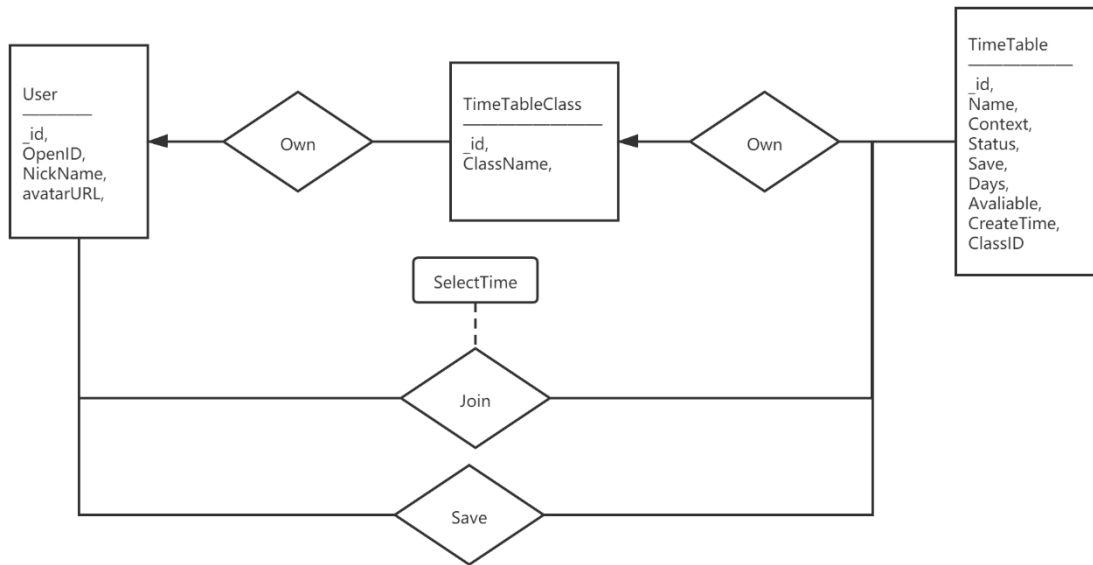
- 采用符合直觉的类时间轴形式展示时间量子，并辅以时间轴标注，减低学习成本
- 通过合适的颜色传达寓意，灰色代表不可选，橙色代表可选，蓝色代表选中
- 提供合适的信息，给到用户清晰准确的反馈
 - 加载完成前的预设值：考虑到小程序加载的情况，我们为组件和页面设置合适的预设值，避免造成用户不必要的困惑
 - Toast
 - 在提交、确认性质的按钮点击后使用Toast报告状态
 - 对于一些校验不通过的失败提交提供相应的Toast说明
 - 异常提醒
 - 新建类提示：对于没有类的情况，显示新增类提示引导
- 增加可读性，降低信息获取成本，提升使用体验
 - 利用卡片等进行信息展示，将表单状态，简介预览等信息放到卡片上
 - 在时间统计详情页的滑动条上增加热图（colorbar）
 - 在合适的区域进行留白，增加文本可读性
- 降低操作成本
 - 默认类：对于新用户，提供默认类，减少新建类操作成本
 - 输入框：自动聚焦和弹起键盘，降低手动操作成本
 - 降低误触成本
 - 二次确认：在表单删除、类别删除等不可撤回的结果处弹出对话框二次确认

小程序后端云开发文档

数据库设计

使用小程序的云数据库，基于其自动生成主键 `_id`，可以较为方便的存储对象，列表等特性，我们可以简化我们的数据库设计。

数据库设计的ER图如下：



可以将其转化为以下五张表

实体集

User

```
create table User(  
  _id varchar(20) NOT NULL COMMENT '使用小程序自动生成的_id',  
  OpenID varchar(20) NOT NULL COMMENT 'vx OpenID',  
  NickName varchar(20) NOT NULL COMMENT 'Name displayed',  
  avatarURL varchar(100) COMMENT 'user avatar from vx itself',  
  primary key(_id)  
)
```

TimeTableClass

```
create table TimeTableLClass(  
  _id varchar(20) NOT NULL COMMENT '使用小程序自动生成的Class的_id, aka, class',  
  ClassName varchar(20) NOT NULL COMMENT 'Name of Class'  
  OpenID varchar(20) NOT NULL COMMENT 'User OpenID belonged to'  
  primary key(_id)  
)
```

TimeTable

```
create table TimeTable(  
  _id varchar(20) NOT NULL COMMENT '使用小程序自动生成的Table的_id, aka, TableID',  
  Name varchar(20) NOT NULL COMMENT 'Name of Table',  
  Context varchar(100) NOT NULL comment 'Context',  
  Status int NOT NULL COMMENT 'Status of the table listed in the following',  
  Save int NOT NULL COMMENT 'whether the table can be saved by others'  
  Days List NOT NULL comment 'up to 3 days selected stored in string',  
  Availiable List NOT NUMM comment 'The quantum time situation marked for each  
Day, -1 means disabled, 0 means abled',  
  CreateTime datetime not null comment 'Create Time',  
  ClassID varchar(20) not null comment 'Class ID belonged to'  
  primary key(TableID)  
)
```

status code	状态
0	未发布，禁止用户填写
1	发布，允许用户填写

save code	状态
0	禁止用户保存
1	允许用户保存

关系集

TimeTable_Member_Relation: 用户填写表单的关系记录

```
create table TimeTable_Member_Relation(  
  _id varchar(20) NOT NULL COMMENT '使用小程序自动生成的关系的_id',  
  TableID int not null comment 'Time Table ID',  
  UserID varchar(20) not null comment 'User open id',  
  SelectTime List not null comment 'Selected time in List Form',  
  primary key(_id)  
)
```

TimeTable_Save_Relation: 用户保存表单到本地的关系记录

```
create table TimeTable_Save_Relation(  
  _id varchar(20) NOT NULL COMMENT '使用小程序自动生成的关系的_id',  
  TableID int not null comment 'Time Table ID',  
  UserID varchar(20) not null comment 'User open id',  
  primary key(_id)  
)
```

云函数API开发

通过云函数的方式，在小程序端提供调用数据库及进行简单数据处理的接口云函数与参数一栏

类型	名称	参数	功能
Get - 用户信息	Login	/	app.js 中判断用户注册状态
Post - 用户信息	UserRegister	avatarURL NickName	注册/更新用户的头像与昵称
Get - 用户信息	getUserInfo	/	获取用户的头像与昵称
Post - add - 类信息	AddTimeTableClass	ClassName	为用户新增类
Post - add - 表单	AddTimeTable	Name/Context/BelongClassID Days/Status/Save	为某个类新增一个表单

Post - add - 关系	JoinTimeTable	TableID	用户加入填写某一表单
Post - alter - 表单	AlterTimeTableInfo	TableID 可选： NewStatus/NewSave... 等更新信息	更新表单内容
Post - alter - 类	AlterTimeSelected	TableID / SelectTime	修改已选择的日期
Get - 类与表信息	GetTableInfo	/	获取用户所有的类与表的信息
Get - 仅类信息	GetClassList	/	仅获取用户所有的类的信息
Post - delete - 表单	DeleteTimeTable	TableID	删除表单与其相关的关系中的信息
Post - delete - 类	DeleteTimeTableClass	ClassID	删除类并递归删除所有属于该类的表单的信息
Post - CopyTable	CopyTable	ClassID / TableID	将表单拷贝到对应的类下
Post - 保存表单	SaveToMine	TableID	保存对应的表单
Get - 保存表信息	GetSavedTable	/	获取用户所有保存的表单列表
Get - 统计信息	GetStat	TableID	获取表单的统计结果
Get - SelectTime	GetSelectTime	TableID	获取用户在某个表单的选择时间结果

云开发过程中的重点

lookup联表查询

类似于sql中表的连接查询，这时我们可以使用小程序云数据库中的aggregate lookup实现。

以下是API: GetSavedTable 的实现过程的一部分。将 TimeTable_Save_Relation 表中的 TableID 与 TimeTableInfo 表中的 _id 字段做自然连接。

```
const wxContext = cloud.getWXContext()
const db = cloud.database();
var $ = db.command.aggregate;
var TableList = [];
await db.collection('TimeTable_Save_Relation').aggregate().match({
  UserID: wxContext.OPENID
}).lookup({
  from: 'TimeTable',
  localField: 'TableID',
  foreignField: '_id',
  as: 'TableInfo',
}).replaceRoot({
  newRoot: $.mergeObjects([ $.arrayElemAt(['$TableInfo', 0]), '$$ROOT' ])
}).project({
  TableInfo: 0
}).end().then(res => {
  console.log('MySavedTable lookup here', res)
  TableList = res.list
})
.catch(err => console.error(err))
```

getUserProfile

获取用户的信息是各小程序开发过程中不可避免的问题。

首先，根据小程序开发的最新规范，获取用户信息的接口改为getUserProfile。我们将这个模块放在Login页中，当用户第一次登录时，会唤醒Login从用户处得到授权，获得用户的头像url，昵称等信息并上传到我们的云数据库中。

处于安全性的考虑与要求，我们不在app前端中存储任何用户有关的信息，如OpenID，头像Url，昵称。而是通过调用云函数的方式，在需要使用这些信息的时候云函数中会通过wxContext 获取用户OpenID，通过OpenID在我们的程序中查询得到头像Url，昵称等其他需要的信息。

在整个过程中，OpenID等敏感隐私信息对前端不可见，这保证了我们小程序的安全性。

npm module: wl-pinyin

在我们的程序中，需要对用户昵称进行按首字母分类与排序。

为了获取中文对应的拼音首字母，我们使用了wl-pinyin的npm包

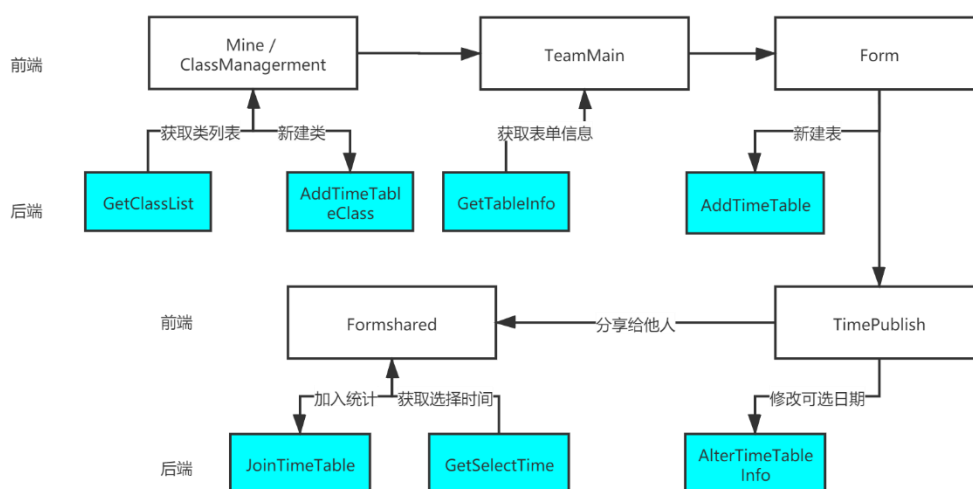
使用 `npm install wl-pinyin` 安装 node modules，然后在小程序开发工具中构建 npm 模块。之后我们就可以通过该模块获取中文字符串的首字母。

```
import pinyin from 'wl-pinyin';
pinyin.getFirstLetter( 'test string' ) //这里会返回字符串的首字母
```

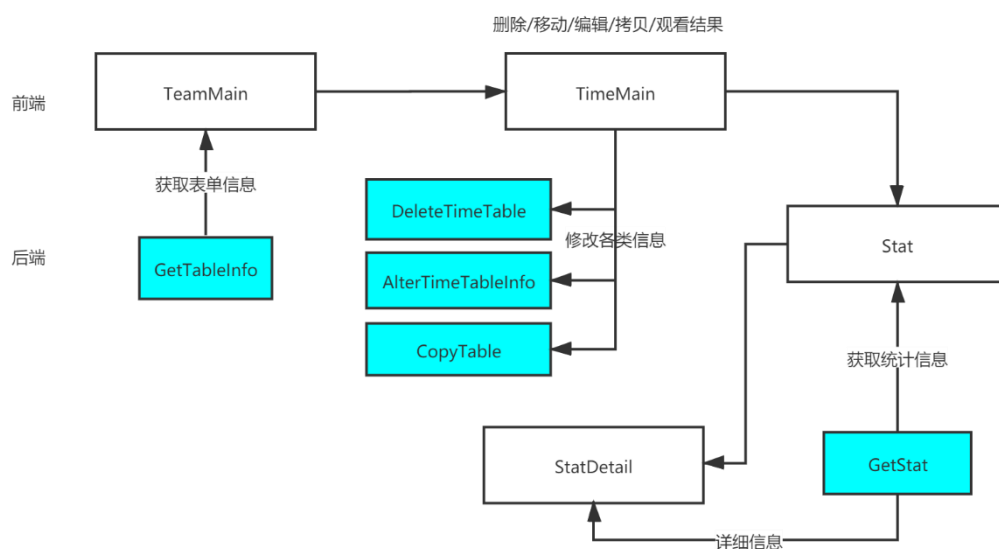
前后端联调

这里列出几条比较重要的前后端数据通路

新建表单并分享的流程



观察统计结果的通路



产品运营、推广与未来发展方向

运营方向

迭代优化

在未来的运营过程中，我们会尽力听取用户的反馈意见。目前我们已经通过**用户反馈群**尝试收集用户的意见，作出相应的调整。同时，我们计划在易出错的地方设置更便捷的反馈入口和更针对性的反馈提示，方便我们更好地采集用户意见。

埋点与数据分析

我们会在产品的一些核心功能上**埋点**，收集数据到我们的后台，使得我们可以在整体上对这些数据进行统计与分析。通过数据了解用户行为，掌握产品动向，更精准地指导我们的优化方向，包括但不限于

- 在用户高频访问的页面优化运行速度
- 将高频访问的功能交互上优先，如调整交互按钮的位置，调整交互的方式，尽量减少用户交互的成本与难度

用户粘性分析

由于本小程序是具有实际应用场景，可以解决需求分析中的某些痛点难点的工具类产品，故对有实际需求的用户，会对本产品具有较高的依赖性。同时，在对市场上的类似竞品进行分析后，我们的核心前端功能具有结果呈现上的可视化优势，在核心功能不改变的前提下，用户会具有较高的粘性。因此我们在推广过程中更加侧重于新用户的引流，当然，实际场景下，是否具有旧用户流失的现象可以通过我们前面提到的埋点方式结合相应的统计分析后得出。

盈利方式

该项目未来可选的盈利方式有

1. 提供更精细化的付费功能

- 如在统计结果呈现上做出更智能，更精准的展现
- 如在用户可保存的表单数量，类别数量上加以相应的付费等级限制

2. 提供广告位

- 参考问卷星等类似工具平台，我们可以在产品页面中，创建表单完毕后等处添加相应的广告页，由于我们小程序广泛的传播属性，会为广告带来较高的点击率，从中获取盈利

3. 流量转移

- 我们可以结合该小程序的应用场景，在同一产品线中继续开发其他产品，在应用场景上形成互补。通过该小程序的流量反哺产品线上的其他产品，而这些产品可能有其自身的盈利方式。

推广方向

我们的小程序定位为小工具，依托微信本身的日常社交属性，我们首先要在功能和逻辑上做到尽善尽美，让使用它的用户能实质上优化 workflow，切实感受到工作效率的提升。

作为一款消费级小程序，**Quantime 现已上线**，当前最重要的工作之一就是用户增长。

在产品本身功能和逻辑完善的基础上，Quantime 借助产品本身的分享特性在社交群体中开展推广。

其中，积累原始用户可以说是至关重要，由于我们的目标群体主要是有较多统计需求的在校学生，以及其他一些具有固定此类需求的场景，我们规划有以下推广方案：

推广方案

1. 在高校内主动地举办相应的校园推广活动，来让更多的人知道我们的产品，邀请同学们体验 Quantime，出于其对痛点的完美把握，再自然形成二次传播。
2. 联系各高校的权益部门，借此推广项目，可以提供相应的校园服务。如校园账号认证等。
3. 寻找具有固定此类需求的场景，如小型公司等。主动联系并进行推广，可以针对开发定制功能。
4. 举办相应的宣传活动，如
 - 专注产品本身的特性进行推广活动，如在相应的比赛中进行推荐
 - 结合开发过程进行推广，如举办有奖 Logo 设计大赛、前端优化方案大赛等

未来开发方向

在已经满足用户的主要交互与功能的前提下，未来我们计划在小程序的统计交互上和统计结果呈现上做更多的优化。

1. 通过筛选，排序，让发布统计者迅速决定合适的时间。
2. 通过智能推荐的算法，一键生成推荐结果，省去组织者烦心的抉择过程。
3. 通过更加人性化的交互方式，让组织者和参与者可以更加便捷的选择对应的时间区块。

Quantime前端文档

文档说明

感谢微信提供的宝贵机会！我们将会在本文档中自豪地介绍本团队开发的时间统计小程序Quantime。这个名字取自一个物理学概念Quantumtime，时间量子，意味着时间是量子化的，有着最小单位，这与我们的时间统计哲学不谋而合。下面将会从单页面说明，页面逻辑，前端核心构思及实现等四个方面介绍我们的小程序。海德格尔说人类在根本上是时间性的存在，希望我们的小程序能够让大家的时间交错，让大家的存在共鸣，一起诗意地栖居。

单页面说明

时间统计&表单信息

以下两个页面是用户与我们小程序的初见。



页面1

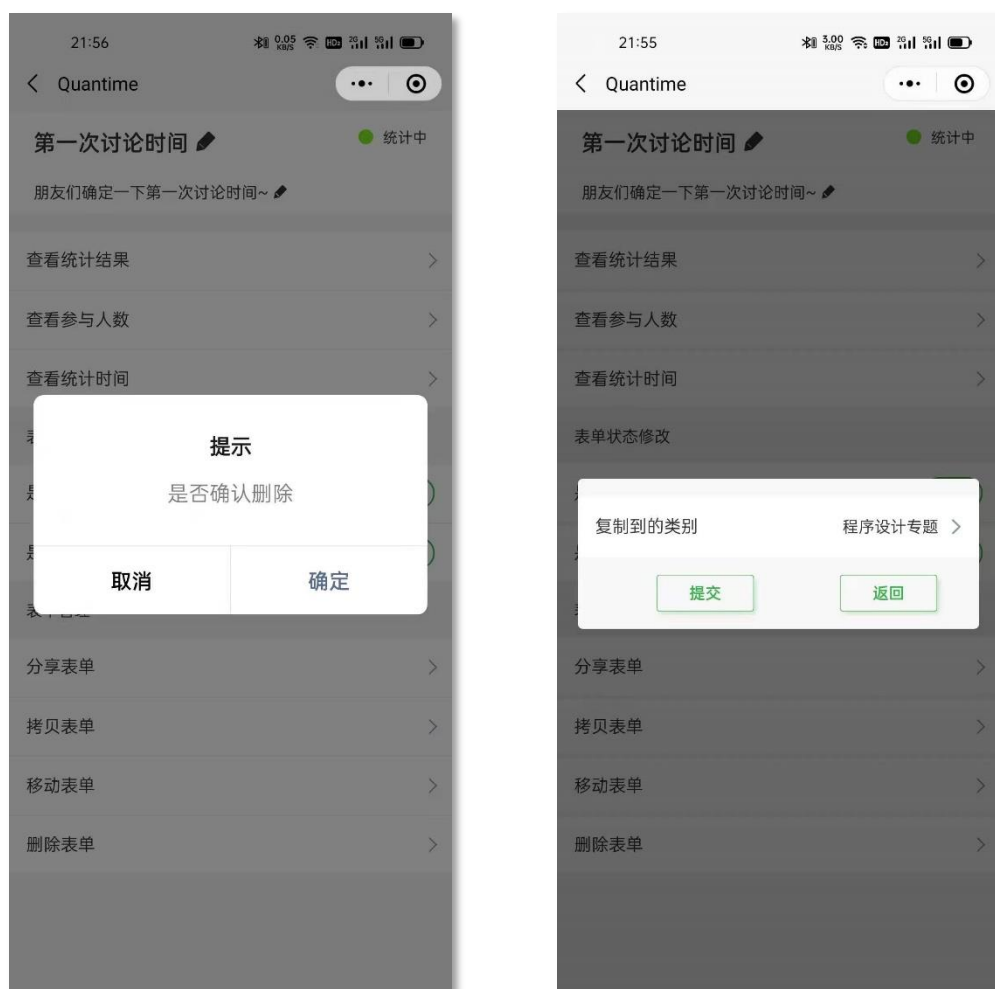


页面2

页面1展示用户创建的统计分类和具体的表单（正如真机截图显示的那样，统计分类可以是一门课程，具体表单是某一次讨论），这样子的二级设计有助于用户整理归纳自己创建的统计表单。同时，对于小组事务繁多的用户，我们提供了类和表单搜索功能，方便用户直达所需。

页面2展示表单的属性和方法，第一部分三个分栏跳转到不同的统计结果分析界面，展示不同视角的时间统计结果；第二部分两个分栏更改表单的状态，功能如其名，设置分布状态为“是”，在页面1中会有绿色“统计中”的标识，否则会有红色“已停止”的标识。第三个部分四个分栏展示表单的四个方法，分享表单会触发小程序自带的分享函数，跳转到分享页面，邀请微信好友或者群聊成员填写自己创建的表单，统计大家的时间。拷贝表单方法允许表单的复用，提高使用效率，移动表单和删除表单允许用户对表单进行改删。

后三个方法均会触发模态框，辅助用户进行准确选择(如下图所示)。



我的信息&类管理



新建表单

新建表单包括两个关键页面，分别明确表单的属性和统计的时间段。

明确表单属性

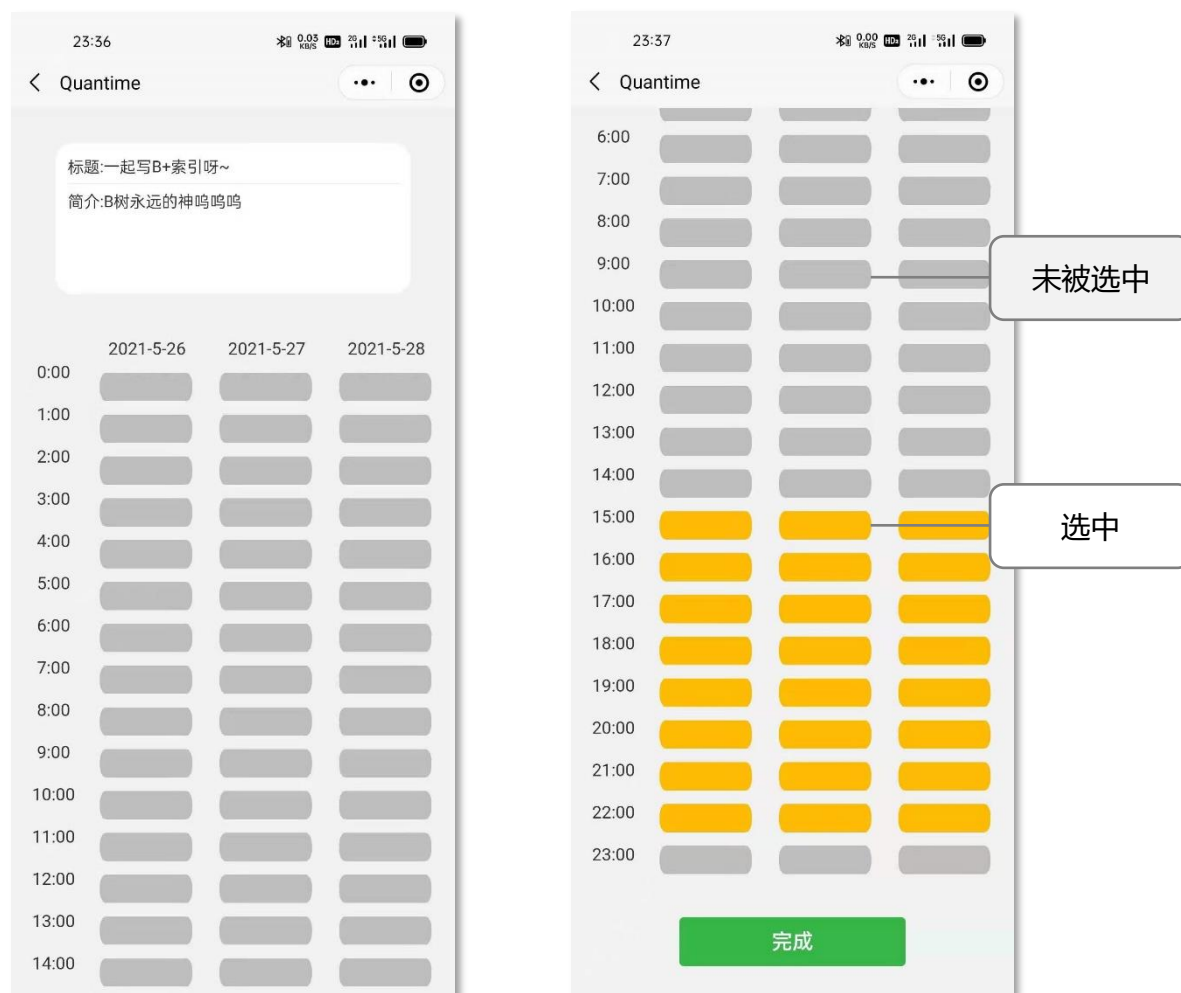
The image displays three sequential screenshots of a mobile application interface for creating a new form. The app is titled 'Quantime'.

- First Screenshot (23:24):** Shows the 'New Form' screen. It has three input fields: '类别' (Category) with a dropdown menu showing '数据库原理' (Database Principles); '标题' (Title) with the text '一起写B+索引呀~'; and '简介' (Intro) with the text 'B树永远的神呜呜呜'. Below these is a date picker set to May 2021. A blue '下一步' (Next Step) button is at the bottom.
- Second Screenshot (23:22):** Shows a modal dialog for selecting a topic. The options are: '程序设计专题' (Program Design Special), '微信小程序开发' (WeChat Mini-program Development), '面向对象程序设计' (Object-oriented Programming), and '数据库原理' (Database Principles). The '确定' (Confirm) button is green, and the '取消' (Cancel) button is grey.
- Third Screenshot (23:23):** Shows the 'New Form' screen again, but with a date picker modal open. The date picker shows May 2021. A dark grey box with the text '最多选择三天' (Maximum selection 3 days) is overlaid on the date picker. The '下一步' (Next Step) button is at the bottom.

根据页面提示自然填写相关信息即可，类别选择使用了微信小程序自带API，让用户可以在窗口滑动选择。为了简便起见，允许用户统计的最大天数为3天，用户选择天数超过三天时会有提示消息，并让溢出的选择失败。简介处还可以填写额外的定制化约束，比如希望参与者选择连续的三个小时等，这样方可应对多样化的统计需求

明确统计时间段

这一页面体现了我们站在统计者(如活动组织者, 小组长等) 角度, 对时间统计的思考。正如我们的小程序名字Quantime, 我们认为时间是量子的, 有着最小的粒度(在这里是一小时), 所以我们提供了按小时点选的用户界面, 统计者可以点击设置参与者们的可选时间, 设置之后, 参与者只需在设定的时间范围内提供自己的空余时间信息。



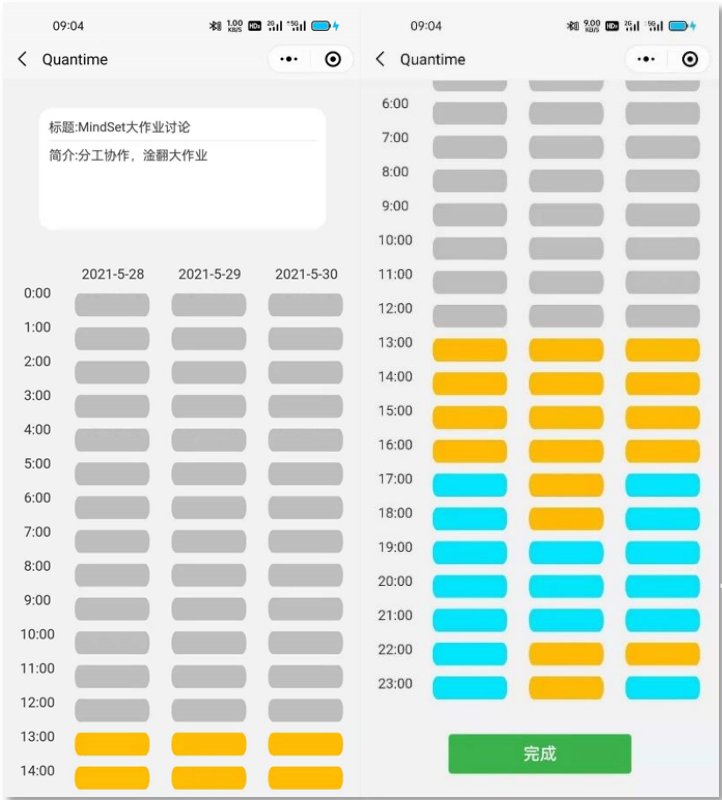
灰色模块表示未被选中, 黄色模块表示被选中。用户可以在**3天, 3*24h**内, 自由组合, 最大化挖掘可选时间, 提高找到大家共同时间的概率。

与以往用户直接指定可选时间段的模式相比, 粒度化展示的时间能够让统计者思考更加具象化, 能够考虑更全面; 与以往人为提供多个时间段选项的统计方式相比, 用户能够用合多为一, 更加直观, 也不用顾虑因为选项太多让参加者为难; **更重要的是, 量子时间能够让碎片化时间得到充分利用, 找到大家空闲时间意想不到的交集**, 这是依靠组织者先验确定可选时间段的传统方法无法做到的。这也是我们选择Quantime这种统计方式的原因。随后点击完成即可成功创建表单。

填写表单

填写表单包括两个关键页面，分别告知参与者统计的目的和可选的时间段，与之前一致，我们采用时间量子的统计方式，**让参与者能够更精细地表达自己的空余时间，充分利用碎片时间。**

参与者点击组织发布在微信聊天中的小程序链接后，会进入邀请界面；在知晓统计目的之后，选择接受；在组织者同意的情况下（在表单属性中设置），如果参与者希望了解统计结果，可以点击保存到我的，在我的信息->我保存的时间统计里面查看统计结果(但无权对于表单进行任何修改)。

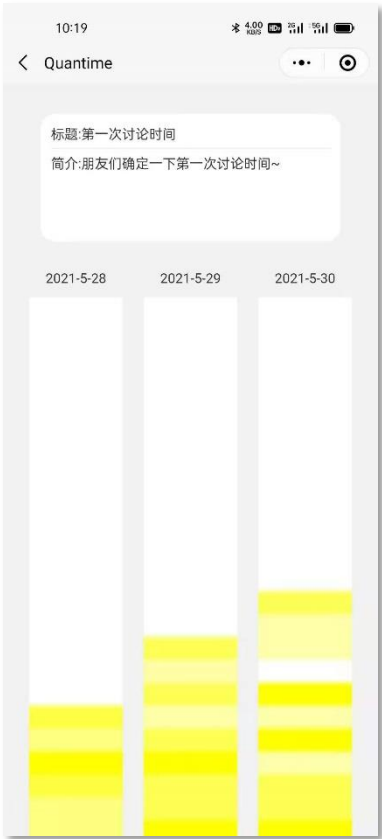


选择接受之后便进入时间统计页面，灰色的时间块表示无需填写，橙色的时间块表示统计者设置的可选时间，被点击时会切换为蓝色表示选中，最后点击完成即可，非常便捷。

统计分析

在收集大家的可用时间后，Quantime会进行自动统计分析，并基于我们的时间统计理念，提供多样的可视化，帮助使用者一眼找出大家可用时间的重叠。

受数据可视化中的热力图概念启发，我们提供了时间统计总览页面，时间块颜色越深表示在该时间段有空的人越多，统计者可以根据该页面快速总览时间统计结果。



此外，受到通讯录的启发，我们提供了带索引的参与者列表展示。统计者可以根据微信名拼写快速搜索相关参与者，定位还未参与统计的组织人员。



最后我们提供了时间统计详情页面，滑动时间条可以查看箭头所指时间段有空的人员，统计者可以根据与会人员信息最后确定合适的讨论开展时间。之后我们将会详细介绍这个页面的设计理念和实现方法。

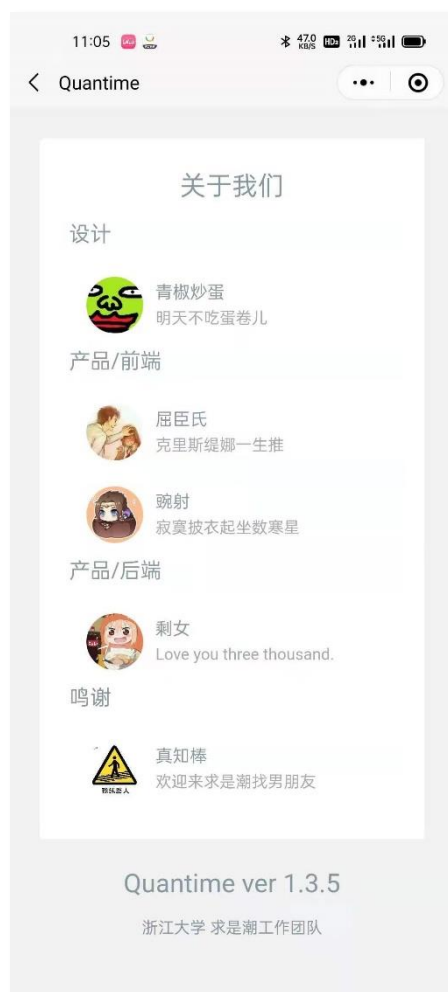


其他页面展示



第一次进入小程序的用户将会看到登录页面，点击获取权限授权小程序获取用户头像昵称等信息

最后可以在关于我们页面看到开发者和版本信息，我们是浙江大学求是潮工作团队，请多指教，欢迎来[Github](#)找我们玩耍。



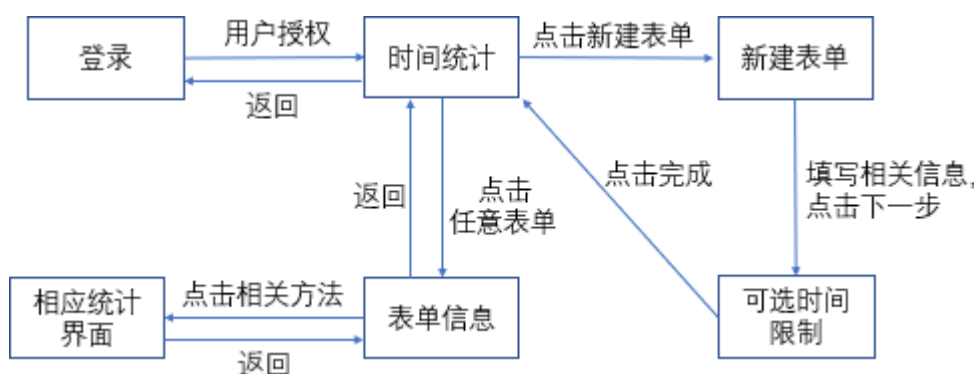
页面交互逻辑

Quantime的页面逻辑和交互设计都追求自然，无需额外学习成本，使用者能够顺畅迅速上手。接下来将从统计者和参与者两个视角介绍页面逻辑。

统计者

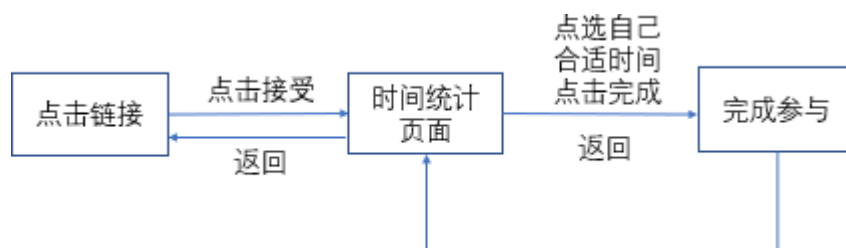
第一次登录的统计者将会见到上面所展示的登录界面，授权后进入时间统计页面。点击任意表单即可跳转表单信息界面。如果想要了解该表单的统计结果，点击表单信息页面的相关方法即可跳转相应统计界面。

点击新建表单按钮，进入新建表单界面，按照引导一步步填写相关信息即可跳转到下一个页面，最后点击完成会返回主页，可以点击新建的表单设置相关的状态并选择发送给微信好友或群聊。



参与者

参与者点击统计者分享的链接即可进入小程序，点击保存到我的收藏夹能获得统计结果的一个拷贝，了解时间统计情况；点击接受即可参与发布者的时间统计，跳转到时间统计页面，点选自己的合适时间，点击完成即可。随后会跳转到自己的主页(如果是第一次登录会跳转到授权页面)，参与者可以在我的信息->我保存的时间统计看到自己之前拷贝的统计结果，了解时间统计情况。



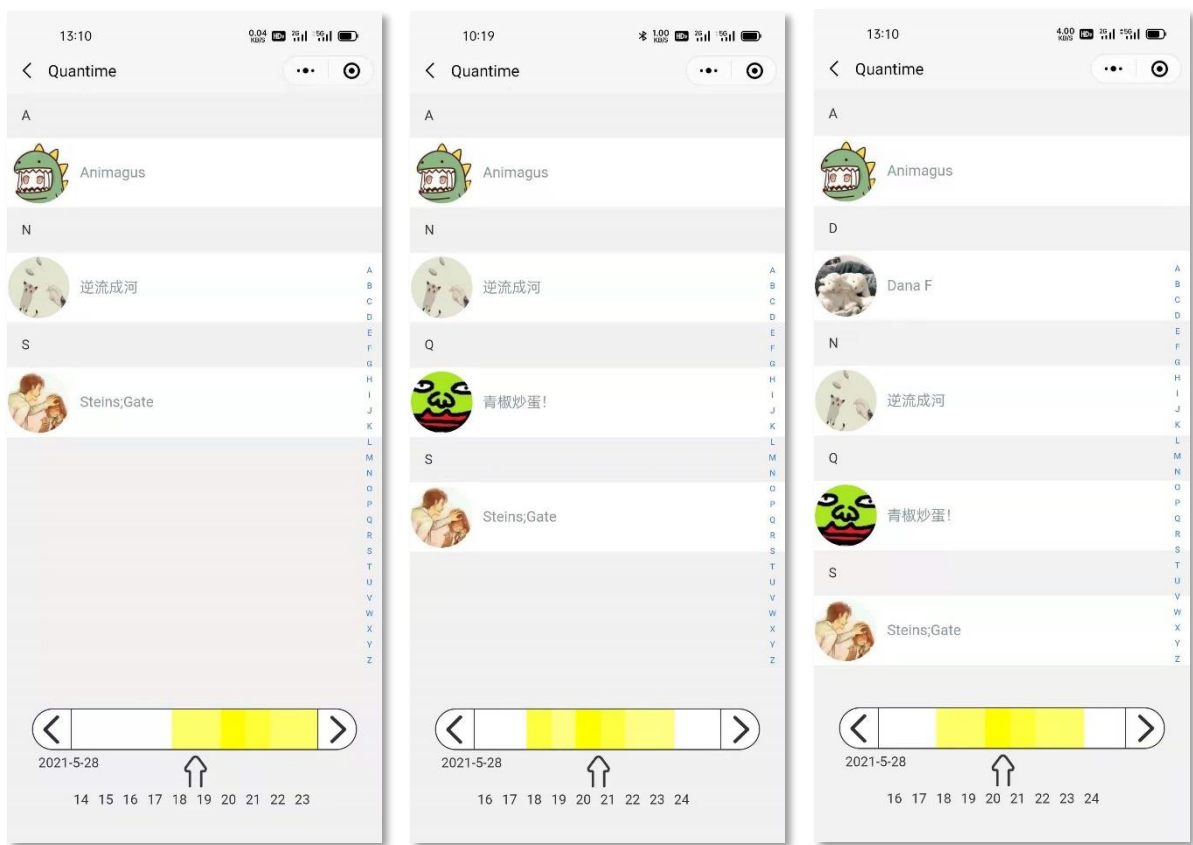
作为用户来说，可以在“我的信息”页面更新用户信息，管理自己定义的统计类，查看已保存的统计表单，看到小程序的版本信息和更新日志，并能向我们反馈宝贵的意见。下图从左到右依次是“查看已保存的统计表单”，“用户反馈”，“更新日志”页面。



前端核心构思和实现

Quantime最核心的前端是时间块组件，下面将分享我们的想法和实现方法。

时间块由三部分组成：最上面的时间传送带(可滑动)，固定在屏幕中央的箭头，最下面的时间轴。正如之前所提到，用户能够滑动时间块(第一张图绿色框内部分) 查看箭头所指的时间段有空的具體人員(第三张图绿色框出部分的时间轴帮助辨认目前箭头所在时间段)。时间块本身也具有热度图显示，人数多的时间块具有更深的颜色，在把握总体和细节之间取得平衡。点击时间块两侧的箭头可以切换查看的日期。



这样设计的初心是，想要提供更直观自然的统计分析结果展示。用户不用输入具体日期和时间段，不用面对统计柱状图，什么都不用输入，触屏滑动即可尽得俯仰之妙，获取各种角度的统计结果展示；同时热度图和名单索引列表，让用户对于统计结果有更直接的感知，也能更准确得找到关键的参与者。通过非常直观自然的界面，没有障碍地获取丰富的信息，从而辅助统计者快速定夺活动举办或者会议开展的时间。

在实现上，将时间传送带和时间轴放在一个Scroll-view中，利用wx-for单独渲染每一个时间段和时间结点，利用fixed布局固定屏幕中央的箭头和传送带两边的箭头，整体包装成一个组件，利用属性以数组形式接受一天内的各个时间段的人数。在JS中利用时间段-人数数组计算

每个时间块的颜色rgb值，并利用微信小程序框架的动态属性绑定机制，让每一个时间块根据该时间拥有的人数呈现不同深浅的色彩。

该组件最大的难点在于如何获取中央箭头目前所指的时间段，从而能够显示在该时间段有空的人。经过我们的思考，采用了一个巧妙的方法，下面介绍大致的想法：

1. 首先利用Wxss方便的rpx机制，结合百分比(%)设置时间块各部分的宽度，做到自适应布局，这样才能保证，不同大小的屏幕，时间块总是在正中间，而且时间传送带一次恰好只能显示十个时间段，这样箭头一开始总是指向固定的时间段(这一点非常重要)
2. 引入空白的时间轴和时间传送带item，实现padding的效果，从而箭头一开始一定指向0点到1点这一时间段。
3. 利用scroll-view原生的scroll方法，获取传送带已经滑动的绝对距离，利用getSystemInfoSync()方法获取当前设备屏幕宽度，计算每一个时间传送带中每一个时间段item的宽度，传送带滑动的距离除以每一个item的宽度即可知道此时已经划过了多少个时间段item；结合一定是从0点-1点时间段item开始，可以计算出出现在中央箭头指向的时间段item。
4. 利用observers机制监听中央箭头指向的时间段item以及日期是否变化，若变化则向父页面传值，父页面获取目前中央箭头指向的时间段信息后，即可准备好并更新相应数据，渲染出当前时间段有空闲时间的人。

核心代码如下：

```
//数据监视，向父页面传值
observers:{
  'Current,date':function(Current,date){va
    rMyDetails={
      'Day':date,'Hour':Current
    }
    this.triggerEvent('ChangePeople',MyDetails)
  },
},
//利用滑动距离，计算获得当前中央箭头指向的时间段
scroll:function(e){
  var dist=e.detail.scrollLeft
  var Cur=(parseInt(dist/this.data.width+this.data.Current/24))%24;if
  (Cur!=this.data.Current)//有变化时才更新，因为scroll方法会被频繁触发
  {
    this.setData({Current:
      Cur
    })
  }
},
```

参考资料：本小程序部分页面基于color ui 和 wx_calendar：第三方“小历同学”日历组件。