

介绍文档——幻享PPT自动生成器&供需平台

1. 概要

1.1 小程序名称——幻享PPT

1.2 小程序简介

1.3 创新点与技术难点

2. 项目背景与需求分析

2.1 领域现状

2.2 市场分析

2.3 具体解决的问题

2.4 用户需求

2.5 系统需求

2.5.1 功能性需求

2.5.2 非功能性需求

3. 产品设计

3.1 术语表

3.2 产品结构

3.3 业务流程图

3.4 UI理念与引导设计

4. 技术实现方案

4.1 开发模式

4.2 开发环境与使用工具

4.3 系统架构

4.4 技术实现

4.3.1 幻灯片自动生成模块

开发过程中遇到的困难

技术实现总结

4.3.2 消息模块

聊天

系统消息

4.3.3 悬赏模块

4.3.3 PPT制作者信息与用户模块

5. 系统测试

5.1 自动生成模块测试

5.2 大厅测试

5.3 发布测试

5.4 其他测试

6. 应用运营

6.1 上线发布

6.2 可持续运营和维护方案

6.2.1 推广

6.2.2 运营策略

6.2.3 维护方案

7. 阶段性需求满足情况与下一步开发计划

1. 概要

在本部分中我们从小程序名称与小程序简介两方面，介绍我们小程序的愿景与小程序的简要信息，帮助读者快速了解我们所开发的小程序。

1.1 小程序名称——幻享PPT

我们团队开发的微信小程序名字为幻享PPT自动生成器&供需平台，幻为幻灯片，享为享受，旨在减少用户在制作和获取幻灯片环节中感受到的繁琐，让制作，获取幻灯片的过程变得流畅舒适，甚至是一种享受。谐音幻想则是考虑到我们的PPT生成功能，目前并未普及大众，而我们做的是将一种次世代的幻想拉进现实，让用户感受到这堪比幻想的体验。

1.2 小程序简介

我们团队开发的幻享PPT，是为了解决因幻灯片制作繁琐而给用户带来的问题，基于微信开发者工具，借助微信云存储，云函数，云调用等功能开发的，具有幻灯片自动生成，幻灯片悬赏平台两大主功能的微信小程序。

幻灯片自动生成功能可以允许用户仅输入标题与文案，即可完成自动生成幻灯片。在此基础上，我们还提供数十种幻灯片模板供用户选择，并且还支持对用户指定的幻灯片页进行智能配图。

幻灯片悬赏平台功能可以方便需求高质量定制幻灯片的用户，去发布悬赏，以求得掌握技术的用户的帮助。同时允许有幻灯片制作技术的用户，去接取悬赏，赚取赏金。

我们通过精心设计的UI界面与交互逻辑，借助微信云开发提供的丰富功能，在微信小程序平台上，最大程度的发挥以上两项主功能带给用户的便利体验。

1.3 创新点与技术难点

我们团队该项目的创新点在于，目前没有任何一个成熟的移动端APP或小程序可以去帮助用户自动生成PPT，我们所做的工作是在移动端PPT生成领域开启新的尝试。

我们的技术难点在于，小程序的开发环境与web端并不相同，web端实现的功能的开源代码移植到小程序上会出现各种兼容性问题，而且讨论都比较少，遇到问题可以借鉴的资源稀少，我们必须自己解决遇到的大部分问题，这也是导致相关竞品稀少的原因。。

2. 项目背景与需求分析

在本部分中我从幻灯片应用的领域现状和对用户的使用感受进行调研，分析用户的潜在需求，以及我们的系统要完成哪些目标去满足用户的需求。

2.1 领域现状

幻灯片是目前广泛应用于演讲的辅助工具，几乎任何一场演讲都离不开幻灯片。幻灯片可以更加形象地体现想法、构思，可以令展示内容更加简单和条理化，可以补充演讲者口头不好展示的内容，以便于更好的表达演讲者想去表达的内容。

幻灯片目前最常用的幻灯片制作放映软件是Power Point（PPT），PPT通过突出重点,简化内容,理顺思路,从而增强与受众的互动,提高沟通效率。提供各种各样的功能，包括丰富的样式可插入内容，多样的流畅动画效果，让展示内容变得生动有趣。

但是PPT同时存在着一些问题，高质量PPT制作难度大需要学习成本；而日常汇报中的低要求PPT，想制作一个简易的大纲PPT，需要打开加载软件，又需要大量令人烦躁的机械操作，例如频繁的选中插入文本框，调节字体样式；扩展性不够强，设备必须安装office，而且版本的不断更新导致一些低配电脑用起来会很卡顿。

以上问题催生了一些其他的幻灯片工具，

第一类是例如reveal.js, slidev, DZSlides, Slidy, S5, Slideous等基于html的幻灯片展示工具，非常的轻便，只需要按照markdown格式输入文案即可生成一个包含动画的精致幻灯片，但缺点也很明显，不是熟悉前端的人很难上手；

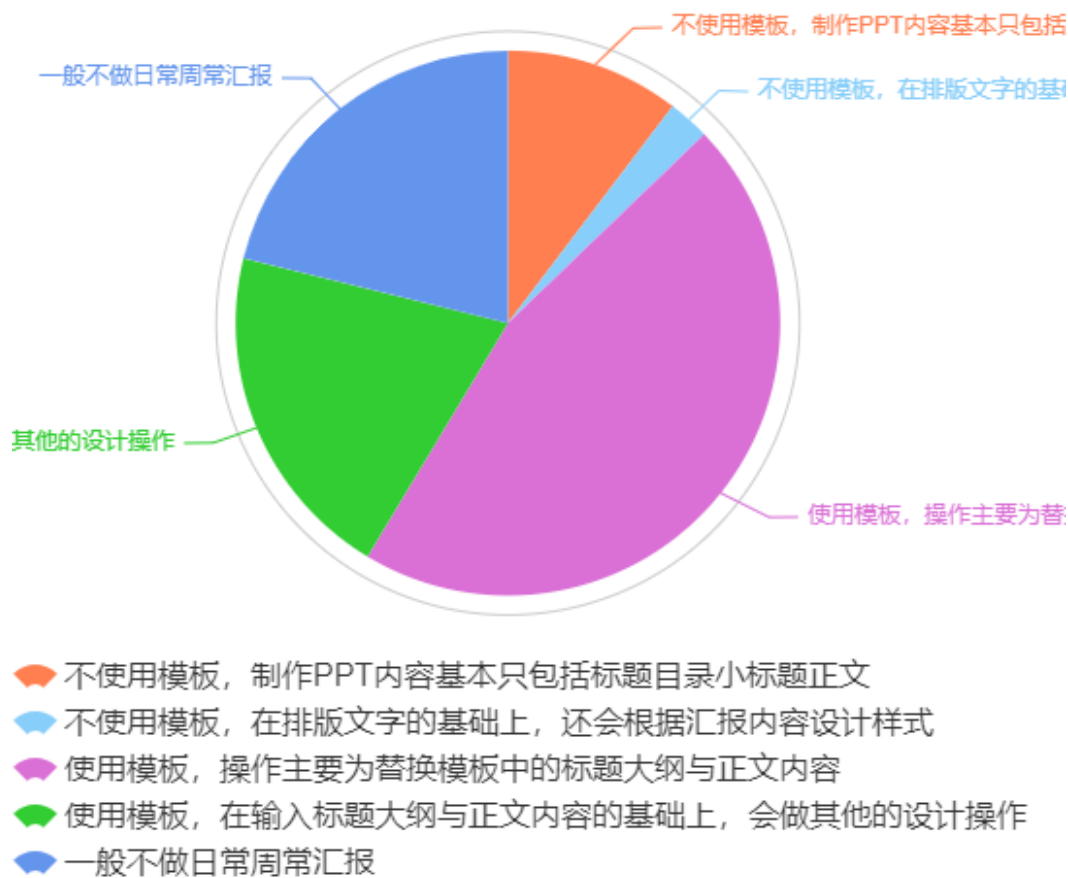
第二类是例如坚果云提供将文档一键转换成PPT的接口的方式，这种方式往往功能与商家功能高度捆绑，如果使用则必须将文档上传到商家软件上才能使用服务，并且也不够便捷，需要使用商家的格式生成文档，再进一步完成转换。

所以我们思考，能不能借助小程序这个平台，去开发一个尽可能解决以上所有痛点的工具呢？

2.2 市场分析

我们制作了问卷，对该领域用户的体验进行调查，并进行数据可视化分析（详细问卷可见最后附录），我们借助任课老师对研究生进行调查，以及队伍成员通过社交网络与家长在群里转载，最终获得了203份结果。

我们首先对直觉上PPT用的最多的用途，日常/周常汇报进行调查，



调查结果发现，在收到的203份数据之中，剔除掉不做汇报的43份数据，在剩下的160份数据中，最常用的使用方式为，使用模板，操作主要为替换模板中的标题大纲与正文内容，不做复杂的设计操作（93份，占比58.1%）。还有21个人选了完全不用模板，基本只输入标题与正文（13.1%），我们调研的做日常周常汇报的用户中有71.2%的用户几乎是不会碰除了输入替换标题与文案的操作的。

我们接下来对这114份数据问下一个问题，是否觉得PPT制作过程操作繁琐。



结果的两端差距非常显著，觉得麻烦的有82人，觉得不麻烦的人有32人。

调研结果表明对日常频繁使用的低要求PPT制作感觉繁琐的人为多数，如果我们可以开发一个，在制作简单PPT更加易用的，并且不像reveal.js而是普通人也可以使用的，减少这方面操作量的工具，是有价值的。

然后通过一个问题，锁定在制作高质量PPT时的，学习与收获不成比例的痛点。

*** 3. 您是否需求过超出自己制作能力的高质量PPT（例如比赛用或者项目用）**

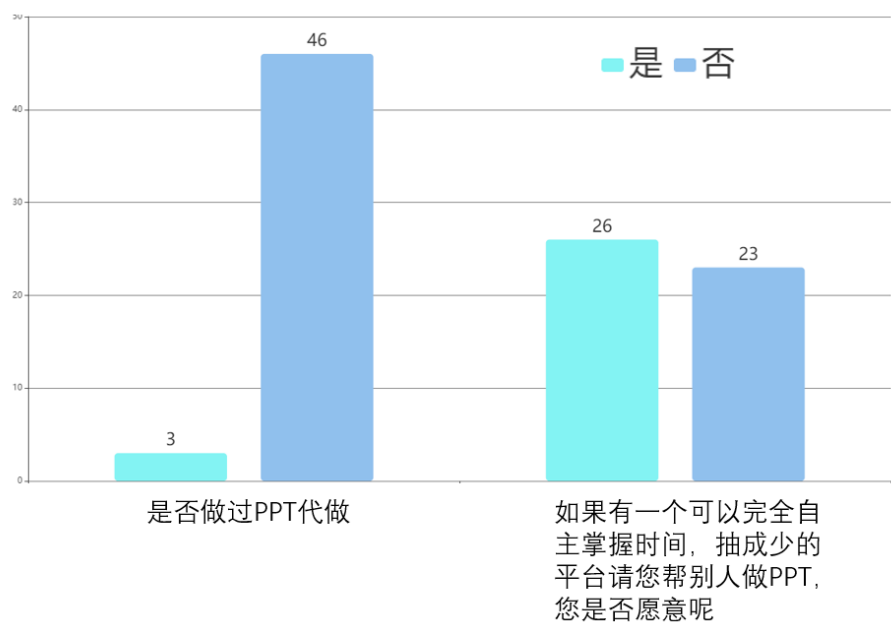
- ☐ 是
- ☐ 否

在上述回答是的67份回答中，有49人回答选择自己系统学习自己做，而有18人选择买。

我们继续询问了这49人：

1. 是否做过PPT代做？
我们发现，绝大多数自己系统学习过PPT制作的都没有做过PPT代做，我自己也有亲身的体会，一般我们不会去找淘宝店或者设计网站，是因为时间成本和未知让我们抵触，这是普遍的很容易理解的心情。
2. 而当我继续询问，如果有一个可以完全自主掌握时间，抽成少的P2P悬赏平台，是否愿意做呢？

非常出乎我意料的是，竟然有超过半数的人，选择了愿意尝试，也就是说如果给与一个低时间成本，且熟悉的平台，有很多人是愿意让自己的技能变现的。



如果是一个自由的面向个人的少抽成的平台的话，做PPT代做的意愿会显著上升，可以看出目前这一块仍有可以进军去扩展市场的空间。

同时为了进一步分析问题，我们假装PPT制作者，对淘宝店进行了对抽成情况的询问，发现对于时间灵活的非全职PPT制作者是会收到20%~50%的抽成。

根据我们的调查，以我们认为目前缺乏一个P2P的，可以让零散的幻灯片需求者与零散的幻灯片供应者对接的，减少中间商赚差价的平台。

2.3 具体解决的问题

经过以上需求调研，我们进一步分析问题

幻灯片目前存在的问题：

1. 制作高质量PPT难度大：制作一个高质量，有契合内容的页面设计，恰到好处的动画效果的PPT是需要比较多的时间去系统学习的，有一定的学习成本，而需求者往往是在一些少数特定场合（参加比赛或者重要项目介绍）才会去用到高质量PPT。这带来了两方面的问题。
 - a. 如果需求者去学习了PPT制作，结果发现后续很少再有需要制作高技术要求的PPT了，则学习与收获不成正比，想用业余时间变现，去找设计网站或淘宝店，往往都会有时间的约束和高抽成，导致收益不高。
 - b. 如果选择购买PPT，购买PPT的途径一般为设计服务网站与淘宝店，设计服务网站可以对比价格，但是一般需要店家和网站双重抽成，导致价格虚高，而淘宝店价格不透明，往往需要货比三家才能避免被宰客。

2. 制作低要求PPT操作多：PPT软件体量大，操作多，哪怕是我们需要一个很简单的，只需要标题与正文不需要图片与动画的PPT，也需要频繁的创建页面，选中挑选功能，创建文本框，设置字体等操作，软件操作往往实际远大于输入文案大纲的时间。

为了解决以上问题，我们要做的：

1. 减少制作低要求PPT的操作。
2. 提供一个幻灯片P2P的供需平台。

而1可以帮助2的推广，让2被更多的用户使用成为可能。

2.4 用户需求

基于我们对市场进行的调研分析，提取了三类主要用户。

1. 简单PPT需求者

对于简单的PPT的需求者，用户的需求为减少PPT制作的操作量，使用尽可能少的操作（例如仅输入文案）去获得满足自己简单需求的PPT。

能够在一定程度上对PPT进行简单的定制，选择风格或模板。

2. 复杂PPT需求者

对于复杂的PPT的需求者，用户的需求在于尽快，尽可能低成本的去获取满足自己需求的PPT。

3. 优秀PPT制作者

对于有能力制作优秀PPT的用户，用户的需求在于，绕过淘宝中间商赚差价，能够直接与需求方沟通，同时可以使用零散的时间去为相应人员提供服务，让自己学会的技能变现，提高学习PPT制作的收益。

2.5 系统需求

我们开始构思我们的系统，首先是平台选取，我们想照顾电脑不熟练的用户，打破必须坐在电脑前制作PPT的痛点，所以我们选择了移动端，并且为了兼顾开发难度的同时，不抛弃安卓和ios的任意一方用户，我们选取了微信小程序作为平台。

然后针对三类用户的具体需求，我们对系统的功能提出了以下要求

2.5.1 功能性需求

编号	需求	优先级
1	输入文案，可以生成幻灯片	高
2	输入文案，可以选择生成的幻灯片的模板与样式	中
3	输入文案，用户可以在线修改生成的PPT的组件与文字	中
4	输入文案，可以获得多种文件格式的幻灯片（pptx或者html）	低
5	输入文案，系统对文案自动进行扩展	低
6	用户可以自定义模板	低
7	用户可以发布自己的悬赏（输入描述，悬赏金额，补充图片）	高
8	用户可以处理自己的悬赏，采纳其他用户提交的文件，或者对接取悬赏者进行沟通需求。	高
9	用户可以与其他用户进行实时聊天交流。	高
10	用户可以接取悬赏，提交文件，领取悬赏。	高
11	用户可以发布自己的个人信息（自己擅长PPT风格，预期价格，做过的作品）	高
12	用户可以在大厅看到PPT制作者信息，主动去联系	高

2.5.2 非功能性需求

编号	需求	说明
1	界面需求	UI设计需要简约美观，能直观的表现功能。让UI能够起到让用户使用舒适，并且在无形中引导用户的作用。
2	易用性需求	小程序的使用方式应符合用户习惯，并且能满足用户其对使用的期望，尽力做到更多的使用者能够轻松上手使用小程序的功能。
3	安全性需求	努力兼顾程序安全、系统安全、数据安全三个方面，保证系统以及用户的数据安全。
4	系统的可扩展性以及可维护性需求	系统开发时应注重代码规范，写好注释，保证代码可维护。并且开发时要保证代码高内聚低耦合，保证具有可扩展性。

3. 产品设计

在本部分中，我们会介绍我们在开始开发之前所做的准备，包括预先对产品结构的设计，用户使用的流程图的预先构建，UI设计理念与潜在指引的设计。

3.1 术语表

为了在后续介绍之中减少篇幅，帮助读者理解我们的内容，减少歧义，我们构建后文常用名词的术语表，供读者参考。

术语	说明
PPT	考虑到很多用户的理解中幻灯片等于PPT，所以PPT前文以及后文中，并不单指微软office套件中的power point，而表示泛幻灯片类型的演示工具，包含PPT与html幻灯片等多种结构。
生成用户	在名词所在用例下，目的为使用幻灯片自动生成功能的用户
悬赏者	在名词所在用例下，需求高要求幻灯片，然后在系统中发布了悬赏的用户
PPT制作者	在名词所在用例下，在系统的个人页面中填写信息注册，可以在大厅中的PPT制作者信息列表中查到的，表示愿意接取需求为他人制作PPT的用户。

3.2 产品结构

我们以微信小程序的普遍界面设计作为我们设计本产品结构时的初始导向，首先将我们的小程序以页面为分割标准，分为四个部分：首页，大厅，消息，我（即个人信息）。

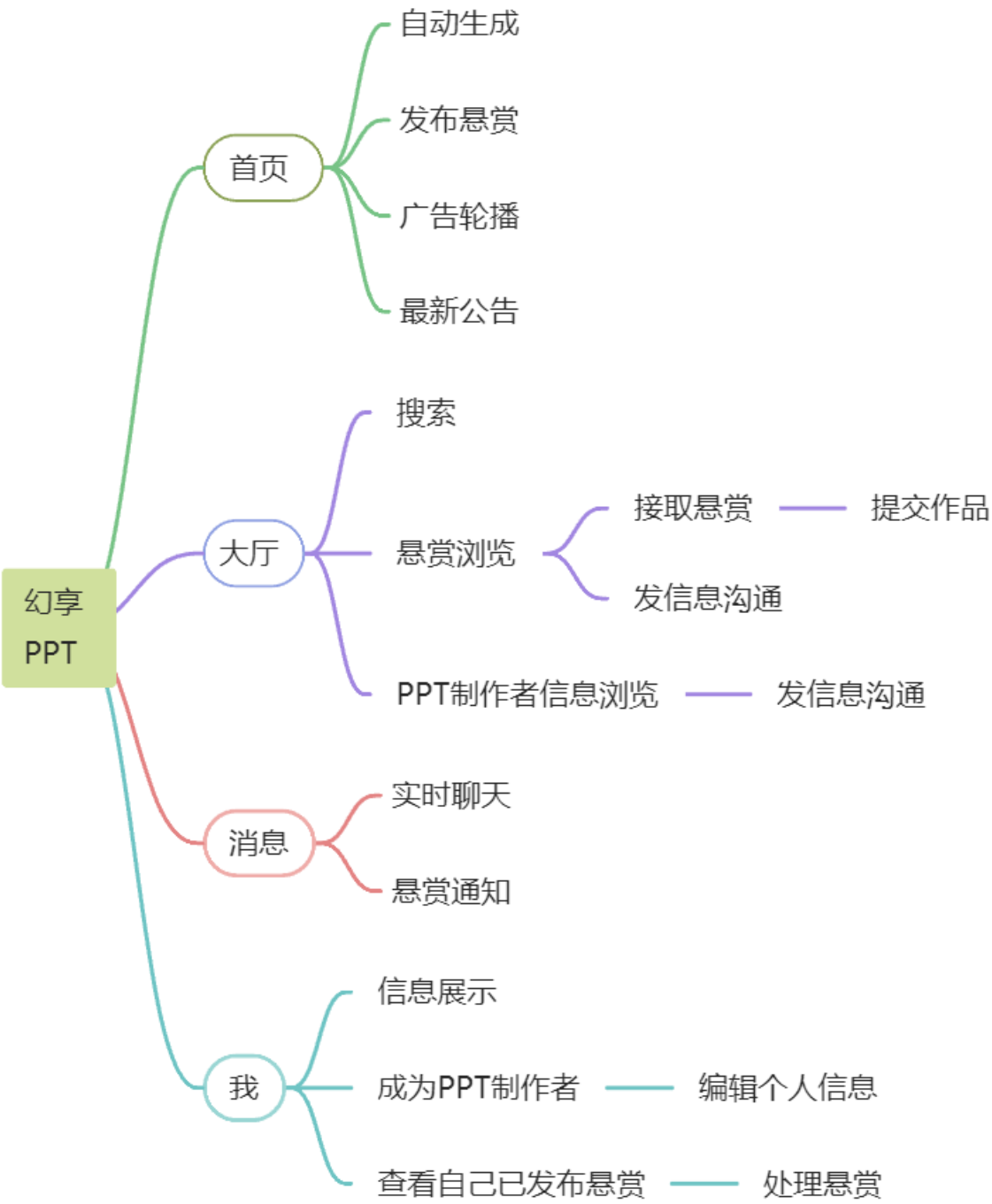
在首页中我们应该包含自动生成模块，发布悬赏的入口，同时可以接一些推广，在首页放置广告轮播图来提升我们的商业价值，在没有商业广告时我们可以放置一些风景图片来给用户舒适体验或者放我们生成过的优秀PPT。而最新公告用以通知用户我们所新增的新功能，提醒用户尝试。

大厅页面，是我们悬赏供需平台的主战场，在此用户可以浏览系统中发布的悬赏，无论是作为PPT制作者还是悬赏者都可以获取有用信息，同时也可以浏览在平台注册的PPT制作者信息，用户可以根据需要使用平台内置的消息功能去与PPT制作者实时聊天。

消息页面，在此页面之中，用户可以浏览其他用户的聊天，并可以点击开启聊天窗口。而系统消息（作为提醒用户发布的悬赏的进度，例如被接取，被提交作品等）也将被实现消息页面。

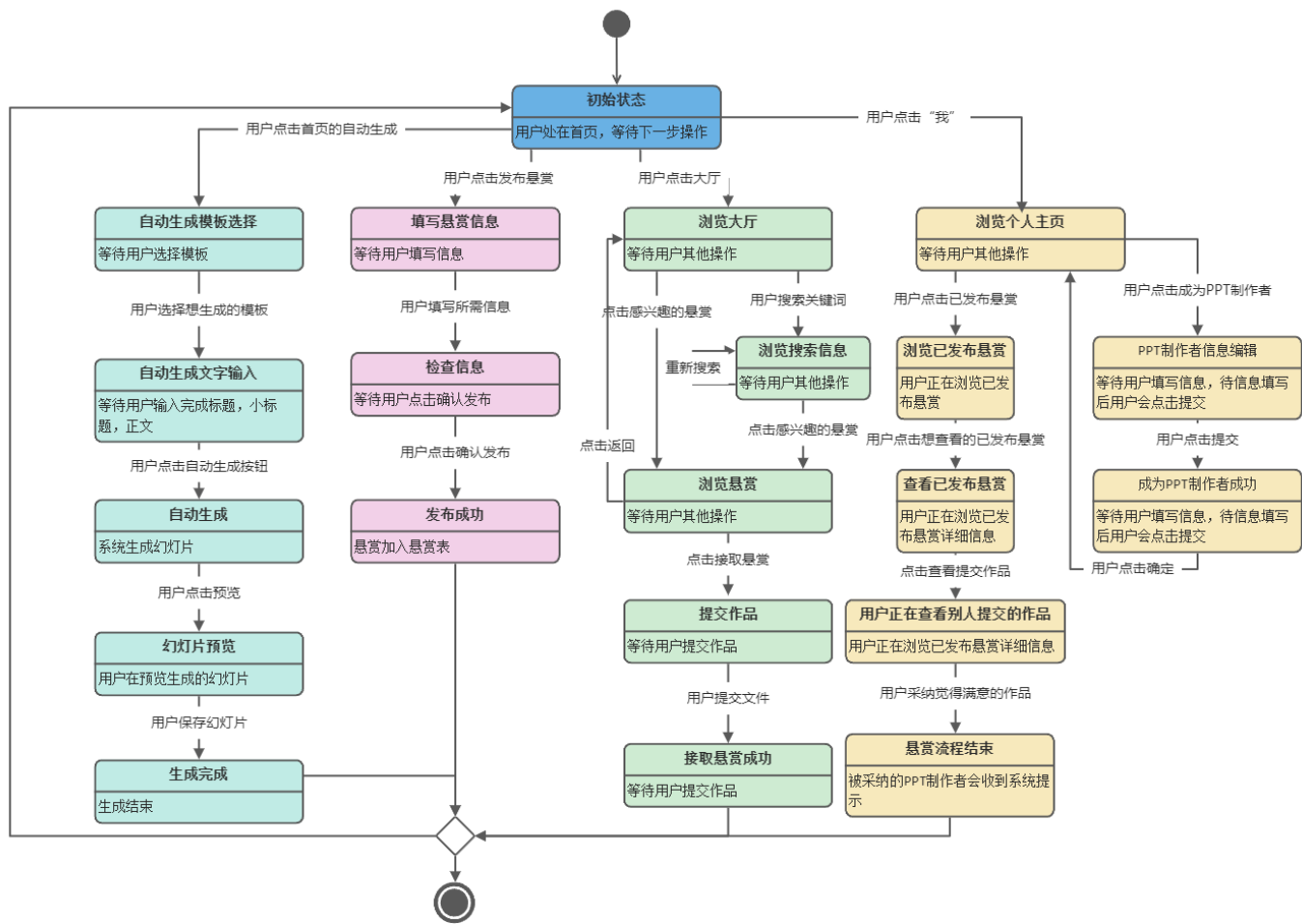
在“我”页面之中，用户可以看到自己的信息，可以点击成为PPT制作者，注册为PPT制作者让大家看到，也可以在页面中查看自己已经发布的所有悬赏，去查看已发布悬赏的处理状态，并可以进入对应界

面进行处理。



3.3 业务流程图

为了更好细致的梳理清楚我们所提供的功能，我们对我们的主题功能（除私信）进行了流程图的构建，对幻灯片自动生成，悬赏，PPT制作者三个模块进行了细致的流程分析，保证功能实现时可以清晰的完成代码编写，同时能够满足系统的要求。



3.4 UI理念与引导设计

我们的UI设计应该严格的遵循我们的软件的理念，一定要针对我们的用户的特点去分析我们应该如何进行设计。

1. 清晰直接的传达，开门见山

首先，我们考虑到使用我们软件的，是需求PPT的用户，而为什么选我们的小程序而不是PPT软件，我们的竞争力在哪里，就是因为我们更加方便。所以在界面以及交互上，我们一定不能做的太冗余缓慢，能减少哪怕0.1s用户耽误的时间，我们都应该努力去做。

所以我们为了用户考量，没有做任何花哨的进入小程序的动画效果，而是选择了开门见山的传达信息方式，我们按照优先级，进行用户功能的排列，将优先级最高的功能，越放在用户使用最少的操作就能解除到的地方。

举个例子：用户一进入我们的小程序，映入眼帘的直接就是两个大大的我们的主功能按钮。

我们还通过层次构图，去引导用户可以迅速的定位到自己想要的，而不是需要点击几层菜单才能进入功能，也不需要浪费任何的精力在功能中寻找自己所需的功能。

细节决定成败。



2. 舒缓不激烈的配色以及样式设计，为用户解压提供舒适体验

考虑到现在大家普遍是ddl战士，往往卡着最后的时间点才去准备，而这种时候，往往是心情焦虑的，我们最初在浅蓝，浅红，浅绿等多种颜色中进行纠结。最终还是从用户的考量，去寻找一个最能舒缓用户心情的配色，让用户一看到我们的界面，就会缓解心中的焦虑，变得心情安定，我们尽力让使用我们软件的过程，多一分享受。

所以我们调研了相关颜色心理学科普文章，在大部分文章中，都提到了类似

黄色给人轻快、透明、充满希望的心理暗示，堪称健康阳光色。在色彩心理学角度讲，黄色是最能改善心情的颜色之一。淡黄色、芥末黄和暗黄色，都会让你的房间亮起来、醒过来！

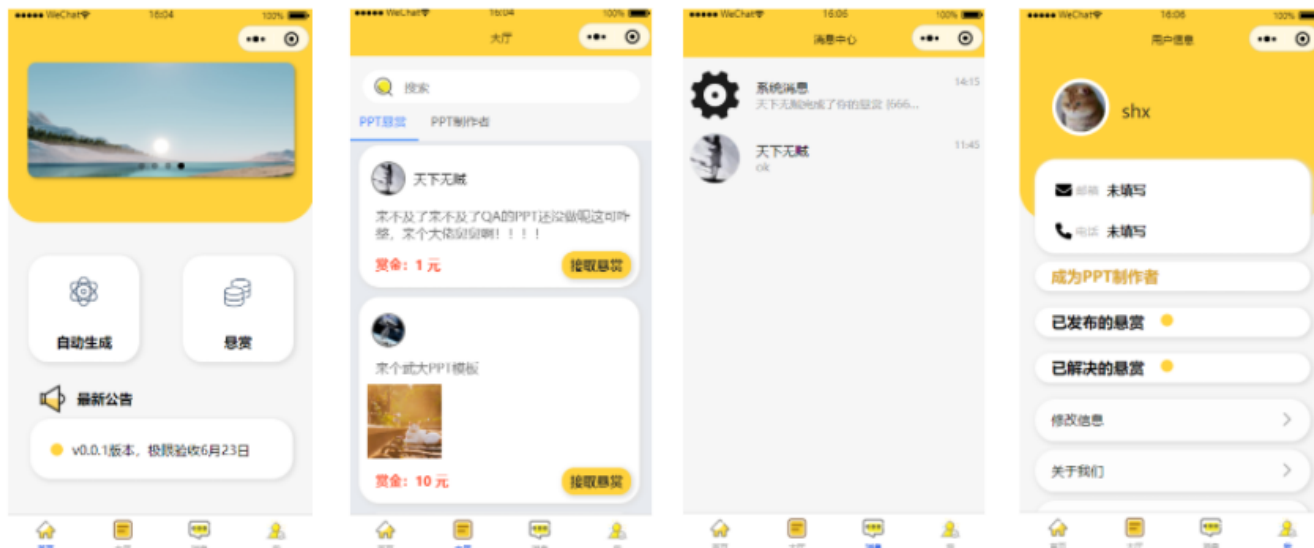
的文字，所以我们最终决定，使用浅黄色，黄白交替作为我们的主色调。而为了某些按钮的醒目，我们考虑使用第二舒缓的浅绿色作为醒目引导。

而在样式设计上，我们也遵循着这条原则。圆角在我们的设计的样式中，都有体现。在我们查阅的相关心理学资料之中，用户在焦虑的时候，看到尖锐的物体，会加重自己的焦虑，所以我们刻意的避免这一点，整个小程序，我们几乎规避了所有尖角样式，尽力让温柔的圆角遍布在小程序的每一个角落，让用户有着极致舒适解压的体验。

3. 整体设计的一致性

设计原则要贯穿整个产品，这包括视觉、交互等不同方面。从交互层面上来说，表现为页面切换方式、导航设计的一致性。视觉上，是指色彩、字体、图标等元素的一致性。没有任何一个用户在使用一款产品时，发现所在新页面的导航居然变了，或者整体风格都变了，从而陷入懵逼状态。

所以我们每一个页面，都贯穿了上面两条设计原则，保证无论用户进入到那一个页面，都不会有任何的割裂感。



一以贯之的黄白配色与圆角样式，给您无比舒适的体验

4. 技术实现方案

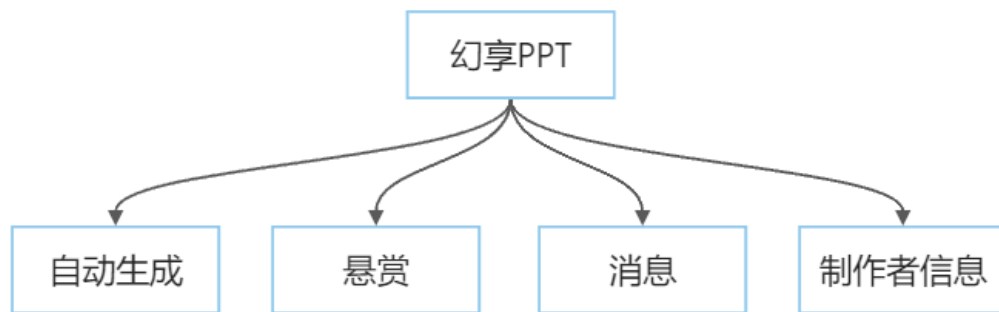
在本部分中，我们将从起到初期规划与宏观把控作用的开发模式开始，到开发环境，再到具体实现，自顶向下的讲述我们是如何实现我们预期的功能的。

4.1 开发模式

在讲具体技术实现之前，先讲述一下我们的开发模式，考虑到开发人员为四位，开发规模不大，同时项目有着需要严格满足用户需求以及系统需求的特点。所以我们团队使用瀑布开发模型。经历需求，分析，设计，实现四大部分。

中间使用了传统的结构化分析模型，构建了数据流图，流程图，ER图，帮助我们理清思路，严格执行功能。

完整文档可见github项目链接（附录）。



4.4 技术实现

在本部分中，我们按照我们实际开发体验中从难到易的顺序，讲述我们在开发各个模块的过程中，我们认为值得在该介绍文档中表述的内容。

4.3.1 幻灯片自动生成模块

此部分我们的开发过程比较曲折，所以出于私心，我想把该部分的开发过程在文档中呈现，如果读者不感兴趣，可以直接跳过到下面的技术实现总结。

开发过程中遇到的困难

· 使用pandoc+reveal.js将功能大致实现

1. 我们一开始对如何使用这一部分的功能，仅有一个模糊大体的认知，就是借助自己写的文本解析器，或者使用pandoc去解析用户输入的文本，将解析好的文本，生成html文本，借助reveal.js之类幻灯片展示框架，因为一开始我们对小程序经验不多，尤其是这类似乎没有人试过的东西，更是知之甚少，所以我们一开始根据网上的教程，搜索微信小程序安装npm包，试图把这几个工具直接安装到小程序根目录。
2. 根据网上教程构建package.json，使用npm install --save将github上能尝试的包全都试了一遍，结果在微信开发者工具内无一能够不报错的运行，报某些函数或变量undefined。经过一番查找，发现缺失的都是nodejs组件，才明白小程序并不是基于nodejs的，并没有nodejs的环境，为了解决这个问题，我们发现了云函数。
3. 然后我们尝试了微信云开发中的云函数功能，我们将pandoc，reveal.js以及pptxgenjs的nodejs依赖使用npm安装到云端。

然后在微信云开发环境下定义一个云函数，在云函数内执行开源库提供的接口，实现我们需要的功能。

这样可以使用小程序，调用部署在云端的云函数，调用开源库。

我们首先尝试了本机调试云函数，功能正常运行了。

但是当部署到云端时，发现功能完全无法正常调用，云端测试也没能提供有用信息，我们再度不知所措。

4. 我们经过翻阅文档，询问与尝试后，定位到了问题。其中pandoc的nodejs并非只依赖于nodejs，我们下载的pandoc的npm包，仅仅是调用本地环境下pandoc的一个wrapper，也就是说调用模块中的函数，需要服务器端部署好pandoc的环境。
5. 我们尝试了一些trick的方法，修改了云函数的内存空间，下载了一份符合云函数操作系统与架构的release版pandoc的包，放到云函数代码同目录下，修改了npm包中的源码，将获取系统环境路径的部分改成了当前目录，将这编译好的pandoc和代码一并送入了云函数空间下，心想者总行了吧。
6. 但是还是不行，查看云端测试的代码，发现pandoc的执行过程是通过child_process创建子进程来实现的，但是报错显示，上传的云函数无权生成子进程，导致云函数根本无法执行。
7. 然后我们购买了华为云服务器，在自己的服务器上搭建了pandoc环境，小程序通过发送request，调用我们自己搭建的服务器部署的服务，实现了功能。

· 使用PptxGenJs扩展模板

在经历了上述困难，简单完成了基础功能后，我们发现了一个问题，html框架下转换出来的幻灯片是好看，排版也不错，也有比较花哨的切页动画，但是pandoc直出的ppt文件实在是太过简陋，纯粹的白底黑字，虽然排版别有一番风格，身边也确实有人有人用这种风格，但根据我们在需求分析中得出的结论，更多的人还是更喜欢用模板的，再加上html幻灯片可能不会被大众接受，所以我们还是选择，再加一套框架，实现更多的ppt文件下的模板。

所以我们继续搜索github，发现了救命的唯一一个可以用代码生成ppt的nodejs框架pptxgenjs，我们学习了pptxgenjs的接口，单纯的调用接口函数，激情加班，造了几套模板，填补了ppt模板的空白。

我们成功开发模板之后，分析了一下可行性，觉得我们完全有能力再加一个，让用户在小程序内自定义模板，这样可以对一些企业收费自定义模板，员工以后再需要ppt，直接使用我们的自动生成功能，就可以生成公司自定义模板的ppt了。

可惜，完成上面功能的时间已经大大超出我们预期了，自定义模板功能到开始审核前，应该实在是没可能做完了，预计未来会将这个功能加进去，届时会在首页的最新公告中声明。

· 修改文本输入方式，对用户更加友好

我们实现的初版功能，是相当于把slidev的功能，从网页端搬到小程序端，用户使用#区分大标题小标题和正文（markdown格式文本），虽然我们写了文档，但是在给指导老师看的时候，老师认为不满意，大

部分用户不应该喜欢输入大段文本，还要自己去分结构。还举了几个低代码项目的例子，我们听完受益匪浅，决定优化用户交互方式。

我们将原本的一整个文本输入框，需要用户手动输入#去给ppt分页，而我们根据老师的建议，拆分成了模块化的输入框，用户可以新增模块来新增页面，每一个模块（页面）我们都单独给出了标题与正文的输入框，避免了用户一次性输入过多文字，增加一定的点击操作，增强了交互性。

在用户输入完成，点击生成按钮后，我们将所有的模块文字，按照markdown的规则，加入控制符，串联成一个整段的一个markdown，将串联好的markdown文本与分块文本送入我们设计的云函数。

技术实现总结

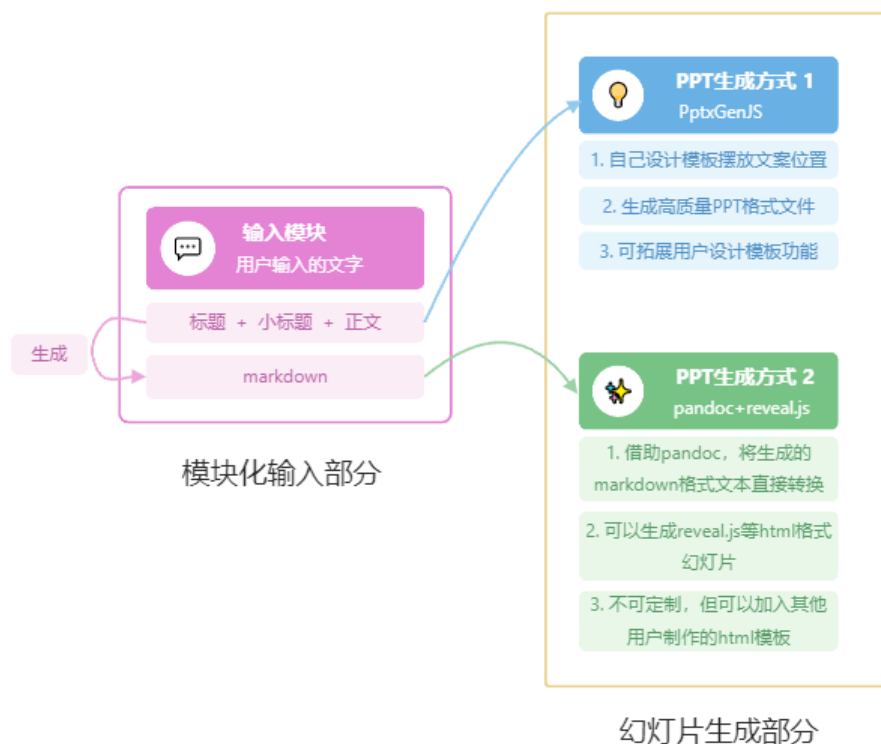
功能逻辑展示：



我们的PPT生成模块，由两个部分组成

1. 模块化输入部分
2. 幻灯片生成部分

实现流程图如下：



模块化输入部分：

在本部分中，我们使用可扩展输入模块来捕获用户想要用以生成PPT的文案。初始模块需要用户输入标题等常用信息，用户可以点击新建幻灯片页，来新建新的输入模块，每一个模块都有该页标题与正文两个输入窗口。输入窗口通过 `wxml` 中的 `textarea` 控件，实时捕获用户输入在模块中的内容，并且写入 `data` 中相应变量。在用户输入完成，点击生成按钮后，我们将所有的模块文字，按照markdown的规则，加入控制符，串联成一个整段的一个markdown，将串联好的markdown文本与分块文本送入我们设计的云函数。

幻灯片生成部分：

在本部分中，我们使用云函数，调用基于nodejs的幻灯片生成框架的两套方法，输入通过模块化输入部分获取的整理好的用户文案，输出我们想要的PPT，通过返回url让用户下载的方式将生成的PPT发送到用户手里。

第一套方法是使用 `pandoc+reveal.js/slidev/DZSlides/Slidy/S5/Slideous`，调用云函数后，云函数解析用户选用的模板，发现是该方法生成的，则走该条路径，云函数将文本，使用 `request`，访问部署在华为云的服务器，服务器调用部署好的 `pandoc`，直接生成对应样式的PPT，并生成一个临时下载链接，传回云函数，云函数传回小程序，用户可以通过传回的链接下载生成的PPT。

第二套方法是使用 `pptxgenjs`，`pptxgenjs` 是可以完全通过微信云函数进行调用的，所以云函数解析用户选用的模板发现是该方法生成的，则走这条路径，通过我们用代码实现的模板，对用户模块化分装好的文案，文案输入我们的函数，函数将生成的文件写到云函数本地，然后我们使用云调用，调用

cloud.uploadFile函数将生成的文件上传到**云存储**之中，用户通过FileID打开文件（可以通过右上角菜单进行保存）（为什么这么做：因为上传到云存储之后，使用wx.openDocument时可以自动识别打开文件类型）。

同时为了避免大量用户同时使用该功能，生成的文件都以用户的openid进行命名，防止冲突。

本模块中使用的云开发技术：云函数，云存储，云调用

4.3.2 消息模块

消息模块包含**聊天**和**系统消息**两部分，我们想考虑兼容悬赏模块，所以由完全自己开发。

聊天

我们考虑到方便用户使用，将消息放置在小程序内，可以避免用户出去加微信联系，并且也方便我们将交易限定在小程序内，方便后续拓展商业内容。所以我们的小程序的开发计划中的实时聊天功能是一个高优先级项。

代码逻辑：

我们首先在云开发的数据库之中新建了 `message` 表，以表属性userAInfo和userBInfo作为数据表的键，用表数组属性message_list来储存userA与userB之间的所有消息。

message

- 含义: 消息记录
- 权限: 所有用户可读,仅创建者可写
- 数据项:

名称	数据类型	含义
userInfo	"openid":openid, "userName": userName, "avatarUrl": avatarUrl	用户A的基本信息
userInfo	"openid":openid, "userName": userName, "avatarUrl": avatarUrl	用户B的基本信息
message_type	bool	消息类型,false为系统消息,true为用户消息
message_list	"_openid" : string, "text" : string, "time" : date	[消息记录]: 发送者的_openid, 发送内容, 发送时间
last_send_time	date	上一次发送消息的时间

用户进入消息界面，后我们使用where对数据库进行查询，使用or条件，查询当前用户是A或者是B即可，压缩表空间。在 `wxml` 显示查询到的所有联系人。

点击联系人，进入到chat+_id界面，每一对聊天者都有自己的聊天界面，识别当前用户，放在右边，即可实现。

特别技术点：

实现实时聊天以及已进入聊天页面时，页面始终会在最下方显示最新的消息

使用微信组件 scroll-view, 在每一次进入页面之后使用scroll-into-view="{{scrollLast}}"将页面滚动到最下方,实现最新消息的查看, 设置scroll-with-animation="true" 实现滚动效果 .

实现多行文本自动换行

过长的消息使用wxss样式属性

▼ CSS 复制代码

```
1 word-wrap: break-word;
2 word-break: break-all;
3 white-space: pre-line;
```

在开发聊天过程中，我们掌握了页面防抖的实现，

防抖就是在一段时间内重复触发事件，延迟事件处理函数的执行,它并没有减少事件的触发,只是限制了处理函数的执行。

常见的搜索输入框都有联想功能,但它们不会在用户不断输入搜索条件过程中重复发送请求,这样会浪费不必要的性能,比较合理的是在用户停止输入的间隙发送请求,这样大大减少了请求

然后在小程序中 所有所有所有所有所有所有 用户输入的位置均实现了页面防抖

JavaScript | 复制代码

```
1  handleInput(e) {
2      clearTimeout(this.data.time)
3      var that = this;
4      this.data.time = setTimeout(() => {
5          that.getInputValue(e.detail.value)
6      }, 300)
7  },
```

系统消息

最核心的部分技术部分使用微信云开发数据库的监听

(<https://developers.weixin.qq.com/miniprogram/dev/wxcloud/guide/database/realtime.html>),云开发数据库支持实时推送变更数据的能力，给定查询条件，每当数据库更新而导致查询条件对应的查询结果发生变更时，小程序可收到一个更新事件，其中可获取更新内容和更新后的查询结果快照

监听数据库变化并且针对变化的数据将内容实时渲染到小程序页面上,同时自定义排序算法,按最近发送时间排序,实现最新接受的消息置顶

外部消息内容显示使用微信的wxss样式属性

CSS | 复制代码

```
1  white-space : nowrap;
2  text-overflow:ellipsis;
```

使用到的云开发技术：云数据库

4.3.3 悬赏模块

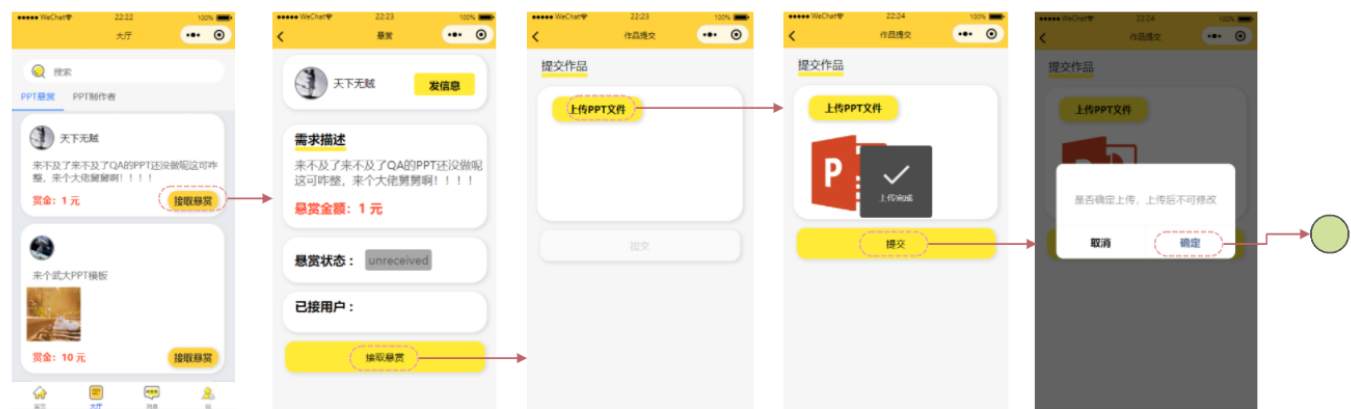
悬赏模块的主要技术点是借助微信开发文档以及教程，使用bindtap设置按钮，使用navigate函数跳转到对应的界面，并且使用setData，查询数据库等函数，执行流程图中相应功能，其中主要实现依赖的内容

为微信小程序基础操作，云数据库和云存储。

以下是我们实现的悬赏发布功能



如下图展示我们实现的接取悬赏的功能



如下图展示我们实现的处理悬赏的功能



在本功能模块之中，我们在云开发-数据库建立了requirement表，用来存储发布的悬赏。

requirement

- 含义: 发布的悬赏
- 权限: 所有用户可读,仅创建者可写
- 数据项:

名称	数据类型	含义
avatarUrl	string	悬赏者头像地址
userName	string	悬赏者昵称
describe	string	悬赏者的悬赏描述
picList	array	悬赏者上传的补充图片的链接
status	string	悬赏的状态
acceptedUserList	array	接受悬赏人的列表
submittedUserList	array	提交作品人的列表
acceptedWorkID	string	解决悬赏的用户的openid
money	string	悬赏金额
uploadTime	date	上传时间

其中认为值得写的技术点：

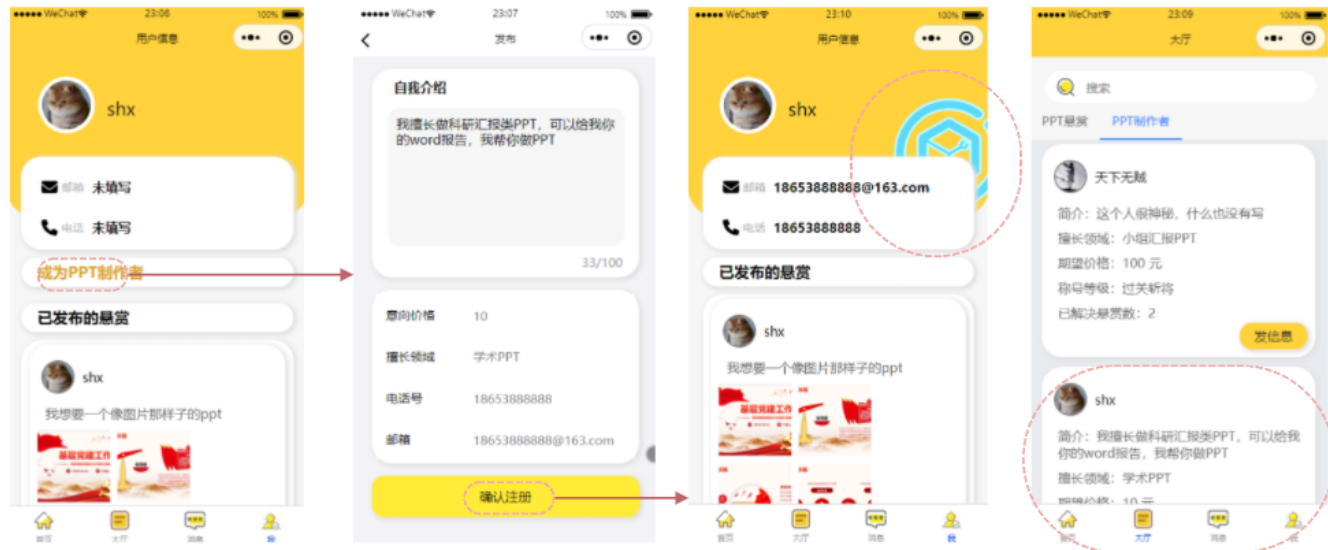
1. 多图片上传以及预览
用户点击上传的+号，通过云调用，调用wx.cloud.uploadFile，将图片暂存到云存储中，并且在页面中使用数组保存云存储中的FileID，然后实时刷新页面，通过FileID可以直接显示上传的图片，点击上传后，直接将保存的fileID数组，作为表项属性插入数据库中。
2. 只预览图片但不提供下载，以防悬赏者不采纳直接下载
最初我们思考，可以读取用户上传的PPT，解析成html格式在页面中渲染出来，或者解析PPT，在每一页加上水印再给悬赏者，在做了一半之后发现wx.openDocument可以给菜单按钮，让用户无法保存，功能直接实现了。
3. 使用了微信官方文档中的NLP接口实现了敏感词检测

使用到的云开发技术：云数据库，云存储

4.3.3 PPT制作者信息与用户模块

PPT制作者模块实现起来用到的技术点，与悬赏模块的类似。

在主页点击成为PPT制作者后，就会在个人主页出有蓝色标识，并且在大厅的PPT制作者栏中出现用户所填写的信息。



值得注意的是，我们另外设置了PPT制作者的声望称号系统，根据用户完成的PPT悬赏量来进行等级认证，等级越高，在PPT制作者栏中，就会排的越靠前，越容易被联系。

我们设计了user表，用来记录个人信息

- 含义: 用户记录
- 权限: 所有用户可读,仅创建者可写
- 数据项:

名称	数据类型	含义
avatarUrl	string	用户头像地址
call	string	用户电话
email	string	用户邮箱
expertise_areas	string	用户擅长的领域
intentional_price	string	用户期望价格
introduction	string	用户介绍
person_title	string	用户称号等级
solved_requirement_num	string	用户解决的悬赏数
userName	string	用户昵称
user_type	boolean	用户是否为PPT制作者

- 相关说明和注意事项:
 - 登录后user_type默认为 `false`，在完成成为PPT制作者后user_type置为 `true`

值得说明的技术点：

1. 我们在用户输入个人信息时，使用了正则表达式进行判断，当用户输入的电话号码或邮箱不符合要求时，会有报错，让用户不能成功提交。
2. 使用了微信官方文档中的NLP接口实现了敏感词检测



所用到的云开发技术：云数据库

5. 系统测试

5.1 自动生成模块测试

在我们完成了pandoc+reveal.js非模块化输入（输入markdown文本）时，请其他的同学设计了测试用例，并完成了第一轮测试，测试结果如下。

编号	对象	标题	目的	测试步骤	预期结果	实际结果
1	PPT生成界面	空输入	容错测试	1.不输入任何内容 2.按确认键	系统提示尚未输入，界面保持输入界面	没有报错，生成空PPT
2	PPT生成界面	错误输入	容错测试	1.不按照格式要求输入内容 2.按确认键	系统提示输入内容格式错误，界面保持输入界面，并要求重新输入	没有报错，以错误文本生成PPT
3	PPT生成界面	正确输入	功能测试	1.用户在提供的模板中选择其中一个模板 2.输入正确文案 3.按确认	系统提示PPT生成成功，并向用户提供PPT文件，文件符合用户预期	符合预期
4-1	输入界面	正确的纯文本输入	可靠性测试	1按照格式要求正确输入纯文本内容 2.按确认键	系统提示PPT文稿生成成功，并进入模板选择界面	符合预期
4-2	输入界面	正确的文本和图片输入	可靠性测试	1.按照格式要求正确输入内容，含有文字和按要求输入的图片链接 2.按确认键	系统提示PPT文稿生成成功，并进入模板选择界面	符合预期
4-3	输入界面	不完全正确的输入内容	可靠性测试	1.不完全按照格式要求输入内容，如##代表的一级标题后直接跟####代表的三级标题 2.按确认键	1.系统根据用户输入标题位置自动重新排版标题，并提示PPT文稿生成成功，进入模板选择界面 2.系统提示输入内容格式错误，并要求重新输入	系统未报错，下至5_3
5	模板选择界	承接	可靠性	1. 用户在提供的模板中选	系统提示PPT生成成功，	符合预期

— 1	面	4_1	测试	择其中一个模板 2. 输入正确文案 3. 按确认	并向用户提供PPT文件， PPT文件符合用户预期， 即纯文本排列正确	
5 — 2	模板选择界面	承接 4_2	可靠性 测试	1. 用户在提供的模板中选择其中一个模板 2. 输入含图片链接的文案 3. 按确认	系统提示PPT生成成功， 并向用户提供PPT文件， PPT文件符合用户预期， 图片被插入到合适的PPT 位置	符合预期
5 — 3	模板选择界面	承接 4_3	可靠性 测试	1. 用户在提供的模板中选择其中一个模板 2. 输入标题级别错误的文案 3. 按确认	系统提示PPT生成成功， 并向用户提供PPT文件， 文件符合正常逻辑，如三级标题被自动转化为二级标题，保证标题结构的正确性	直接按照用户输入内容生成ppt，没有考虑标题的问题
6	输入界面和模板选择界面	多个用户同时操作	压力测试	多个用户同时对系统进行操作	系统针对各个用户做到正确处理	符合预期
7	输入界面和模板选择界面	正确输入	性能测试	正确输入	系统在规定时间限制（如10秒内）做出正确处理	符合预期

我们对上述问题进行了改正，通过将整块输入改编为模块化的方式，解决了输入错误格式信息的问题。

并且我们针对空输入，进行单独处理，现在可以对空输入进行报错，避免浪费资源。

修改代码后，以上可用测试用例依然通过，我们认为无需再增加测试用例。

5.2 大厅测试

编号	对象	标题	目的	测试步骤	预期结果	实际结果	说明
1	大厅	无悬赏	容错测试	清除数据库中悬赏	不显示任何悬赏，显示下拉到底的图片	符合预期	无悬赏时显示应与其他平台类似
2	大厅	超过一页悬赏	容错测试	往数据库中插入超过10条悬赏	下拉可以继续显示剩余悬赏	符合预期	测试悬赏多时的情况
3	大厅	带图悬赏测试	容错测试	往数据库中插入带图片的悬赏	正常显示	符合预期	可以显示带图的悬赏

5.3 发布测试

我们在本部分中，对悬赏发布和个人信息发布进行测试

编号	对象	标题	目的	测试步骤	预期结果	实际结果
1	悬赏发布与个人信息发布	正常测试	容错测试	输入正常合法内容	可以正常发布	符合预期
2	悬赏发布与个人信息发布	敏感词测试	容错测试	在内容中输入不文明词汇或句子	提示含有敏感词	符合预期
3	悬赏发布与个人信息发布	特殊结构测试	容错测试	金额，手机号，邮箱分别不按格式输入	都会提示请按正常格式输入	符合预期

5.4 其他测试

考虑到其他功能大都使用了规范的插件，或者是一些简单的代码逻辑。

我们请了数位同学与老师进行上手测试，并没有发现明显问题。

6. 应用运营

6.1 上线发布

我们的小程序在本地模拟器调试，与真机调试中，均能提供功能正确的流畅体验，已提交审核，预期会在近日发布。

6.2 可持续运营和维护方案

根据需求调研以及目前开发进度以及功能测试体验，用户的积极反馈给了我们对该项目可用性的充足的自信。

我们将从，推广，运营策略，维护方案三方面展示我们未来的规划。

6.2.1 推广

1. 从校园内民间组织进行合作，聚拢初始用户

利用社交圈子，与同学所在的学校学生会和技术社团等民间组织建立合作关系，通过表白墙进行打广告，进一步可以通过学校官方账号扩大影响力。

2. 建立官方线上平台

建立微信公众号，微博官微，b站官方账号，抖音官方账号从传统社交平台到新兴短视频平台，通过文章配图，短视频，长视频等多种媒介，利用结合热点积极发声等公关技巧，尽可能低成本高效的向尽可能多且广的用户，介绍“幻享PPT”小程序，并且及时的进行用户分析，积极的用户调查，定位到更细化的用户类型，精准有策略的设计下一步推广计划。

3. 在对应论坛或讨论区针对性推广

百度贴吧，知乎等平台往往有PPT相关的讨论与问题，可以在问题下，利用官方建立账号，分享使用我们软件带来的便利，以吸引更多的愿意尝试的用户。

6.2.2 运营策略

我们的运营策略简而言之即为，品质为先，用户至上。

作为一个目前市场，没有在同平台有竞争对手的工具类程序作品。我们认为，我们仅使用常规推广，在社交平台官号，及时发布更新，与用户进行沟通，不搞歪门邪道花里胡哨的宣传足矣。

我们目光的侧重点，应该如何保持软件稳定的基础上，丰富软件的功能，优化用户的体验。

经常性的去查看用户在各大平台上的反馈，收集用户的反馈，针对用户在使用中的痛点，进行优化。

做到以品质和态度赢取用户的喜欢。

6.2.3 维护方案

- 1. 纠错性维护：我们通过开启的反馈渠道，及时查看用户的建议，收集用户使用中收到的问题，采用每周固定时间查看用户反馈，并根据bug类型及时制定纠错方针，bug不存到下一周。
- 2. 适应性维护：同时兼顾了解友商，查看市场上是否有更新的竞品，我们应该学习，不断地适应新环境。同时订阅我们使用的开源库，如果开源库更新，我们也应该积极更新以适应新功能。
- 3. 完善性维护：我们会及时制定下一步的完善计划与里程碑，完善计划会结合用户的体验，争取与用户一起让软件变得更好。

7. 阶段性需求满足情况与下一步开发计划

我们对我们在需求分析阶段，分析所得的需求进行阶段性验收，需求完成情况如下（截至文档撰写完毕的6/30/2022）。

需求	优先级	完成情况
输入文案，可以生成幻灯片	高	已完成
输入文案，可以选择生成的幻灯片的模板与样式	中	已完成
输入文案，用户可以在线修改生成的PPT的组件与文字	中	未完成
输入文案，可以获得多种文件格式的幻灯片（pptx或者html）	低	已完成
输入文案，系统对文案自动进行扩展	低	未完成
用户可以自定义模板	低	未完成
用户可以发布自己的悬赏（输入描述，悬赏金额，补充图片）	高	已完成
用户可以处理自己的悬赏，采纳其他用户提交的文件，或者对接取悬赏者进行沟通需求。	高	已完成
用户可以与其他用户进行实时聊天交流。	高	已完成
用户可以接取悬赏，提交文件，领取悬赏。	高	已完成
用户可以发布自己的个人信息（自己擅长PPT风格，预期价格，做过的作品）	高	已完成
用户可以在大厅看到PPT制作者信息，主动去联系	高	已完成

针对未完成的需求进行说明：

1. 输入文案，用户可以在线修改生成的PPT的组件与文字：

我们认为该技术很难在移动端实现，技术力不允许，在实际开发后我们认为此条不可行。

2. 输入文案，系统对文案自动进行扩展

我们对NLP领域发展做了调研，目前文本自动扩展的研究还没有智能到，可以生成完美符合期待，又能匹配格式的PPT内适用的文字，目前不可行，但我们会继续跟进学习，如果领域有新进展，我们会及时的将功能突破更新。

3. 用户可以自定义模板

我们认为可行，未完成的原因在4.3.1中讲述了，因为开发基础功能所化的时间超出了团队计划，所有暂且搁置了该功能。

下一步的开发计划：

1. 用户可以自定义模板

我们观察学习了一众PPT模板，认为其实组成PPT的组件其实并不算多，尤其是我们的功能要求仅是能够美观实用的契合文案，所以我们认为将模板抽象成我们能操作的组件其实并不复杂，难度应该不会高于开发一个小游戏，我们有相关的开发经验，我们认为有很高的可行性。我们预计会在8月31之前完成该功能

2. 更多的模板选择

目前我们开发的模板数量虽然数量上很多，但实际可以用的大部分是大同小异，只有pptxgenjs产生的模板是我们自己亲手打造的，具有一定独特性的模板，html模板大都只是pandoc直出的简单换色。我们认为reveal.js等html幻灯片展示框架，还有继续可以挖掘的空间，可以生成包含更加多样样式与特效的模板。

会随着学习进度，在小程序生命周期内随时逐步添加。

3. 智能配图

我们在开发过程中调研过这个模块，使用NLP模型，对文本和图片进行匹配，搜索，但是可用的开源代码，都是基于给定的网络图片数据集，而这些数据集往往不具备艺术性，简单的提取用户输入的关键词，然后对关键词进行文本和图片匹配，搜到的图片往往不够美观。

我们考虑尝试自己构建数据集，和使用网络搜图，可能会涉及版权等诸多问题。如果我们发现能够解决遇到问题的方案，我们的软件又将增添一笔色彩。但是目前还没有进行足够的调研，无法确定可能性，故该功能发布时间待定