

咔记小程序

每天上传一张照片，记录你的此刻。

队伍编号	NO.328198
队伍名称	做不完不睡觉
指导老师	李本富
队长信息	马勇杰 南方医科大学 生物医学工程学院
队员信息	林培奇 南方医科大学 生物医学工程学院 李泽臣 南方医科大学 生物医学工程学院

设计理念

随着人们的社交的泛化，越来越多的年轻人选择在微信朋友圈记录自己的生活，而在其他较私人化的地方抒发自己的情绪情感。这类情感有些用户只希望自己能够看到，有些用户则希望将这些情绪动态在特定的时间和特定的人分享。于是，“咔记”这款小程序孕育而生了。

产品定位

“咔记”以每日最多上传一张照片以记录生活和心情为基础。当用户上传的照片达到三张时，即可使用生成视频的功能，通过选择特定的照片集生成有字幕、时间和地点的视频，并且可以选择性的分享给自己的朋友。使用户在微信中也可以拥有释放真正自我，表达自我的一片天地。

产品适宜人群

全年龄段

技术开发方案 - 前端

选用框架 - **mpvue, vuex, weui**

因为我们队伍中的小伙伴都有 Vue.js 的开发背景，所以我们使用了美团点评最近开源的 mpvue 框架。mpvue 框架提供了完整的 Vue.js 的开发体验，大大的提供了开发效率。同时 mpvue 使用了 webpack 进行构建，使我们可以很方便的使用 async/await 关键字, scss 预处理器和其他 npm 的依赖。

同时，我们还用 vue 的状态管理器 vuex 来管理数据，简化了数据的操作流程。

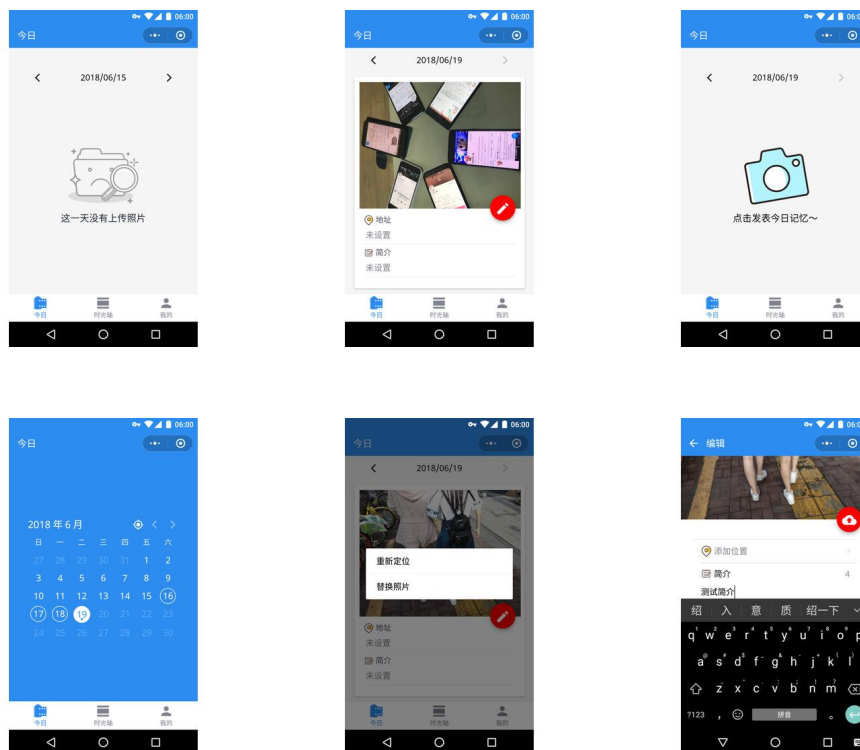
在 UI 上我们使用了腾讯的 WeUI 的 UI 库，以保证视觉效果和微信整体统一。

前端页面数目 - 10 个

包含：教程页面（pages/tutorial）、视频列表页面（pages/video-list）、视频详情页面（pages/video-detail）、视频编辑页面（pages/video-editor）、照片编辑页面（pages/phto-edit）、展示今日照片页面（pages/today）、时光轴页面（pages/timeline）、用户个人页面（pages/user）、生成视频页面（pages/selector）、查看分享视频页面（pages/share）。

核心功能点 - 6 点

1. **每日拍照**：用户每次进入“咔记”都会进入“今日”页面。若用户今日尚未拍照，则会显示提醒；反之，则展示用户今日已上传的照片。同时用户也可以通过上面的日历组件跳转至任意时间查看之前上传过的照片。



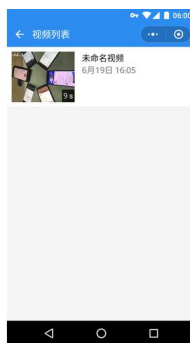
2. **时光轴**：在时光轴页面，用户可以浏览之前已上传的照片，同时也可以通过右下角的按钮进入视频生成页面。



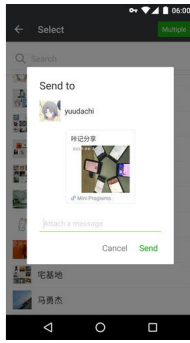
3. **用户设置**：用户可以在个人设置中设置是否开启每日拍照的提醒，也可以选择生成的照片的显示时长等。



4. **生成视频**：用户可以在生成页面中通过选择需要的照片进而生成视频，用户已生成的视频也可以在“我的”页面中随时查看或者下载至本地。



5. **视频分享**：用户可以将生成好的视频随时分享给自己的好友查看。



6. **每日提醒**：“咔记”还可以通过设置，在每日的特定时刻对用户发出拍照提醒。当用户在 7 日之内没有在小程序中进行操作的话，也会自动取消提醒。



存在的问题/可以优化地方 - 5 点

1. 视频在播放时显示的时长有误。有可能是因为我们生成的视频的一般比较短（6s 左右），video 组件不能很好的显示时长。
2. 我们使用的一个 mpvue 的组件 mpvue-ripple，有时会发生报错。主要的问题是其中经过了一次 timeout 的延时，但是没有对结果进行判断所以产生了错误。
3. 后续可以使用 TypeScript 进行编写。因为使用了 webpack，理论上我们只需要添加一个 TypeScript 的 loader 就可以直接使用 TypeScript 进行开发。但是因为部分前端开发的队员不熟悉 TypeScript 所以一开始定型时没有采用。TypeScript 拥有对语法和类型的检查，可以使整个项目更加地健壮。
4. mpvue 目前对一些 Slot 中的自定义组件渲染有问题，所以部分地方组件的复用率并不高，需要等后续上游进行更新。
5. 在时光轴（timeline）的页面中的 v-for 循环目前还没有进行优化。（虽然一般来说需要一百多组数据以上才会出现性能瓶颈）后续可以使用分页加载，无限滚动等进行优化。

技术开发方案 - 后端

技术选型 - Node.js, TypeScript, PostgreSQL, Redis

为了满足快速开发的需求，我们选用了 Node.js 的作为后端的编程语言。同时，为了保证

程序的鲁棒性和可扩展性，我们还使用了 TypeScript。

我们使用了 PostgreSQL 作为主要数据库；同时因为发送模板信息的 access token 和 form id 具有有效期的设定，不适合存在 SQL 数据库中。所以我们选用了 Redis 作为缓存数据库。

最佳实践 - systemd, cos, cdn, RESTful, JWT

为了保证服务器上的 HTTP 服务能一直保持运行，我们使用了 systemd 对其进行管理。保证了 HTTP 服务即使出错之后也能立即重启。同时，定时提醒也是通过 systemd-timer 而非 cron 实现，这样能更好对日志进行管理。

一开始我们将用户的照片和视频都存在服务器上，但是因为服务器的带宽所限，照片和视频的加载都受到影响。视频在播放的时候甚至会出现有绿边的现象。于是我们使用了腾讯云所提供的对象存储服务（COS）来存储照片，解决了服务器容量和带宽有限的问题。同时对象存储服务也提供了 CDN 的功能大大地加快了用户加载照片和视频的速度。

我们所有的后端 API 都完全遵守了 RESTful API 的原则；同时我们还使用了 JWT（JSON Web Token）对用户进行验证和判断。

存在的问题/可以优化地方 - 3 点

1. 在给用户发送模板信息时，可以先对用户今天是否已经上传了照片进行判断。这样可以节省对 form id 的使用。
2. 在使用了 JWT 进行用户验证之后，对于重复运算居多的非对称加密来说，Node.js 的性能就比较吃紧了。后续可以使用 Rust 或者 C++ 等高效语言来编写验证服务器，而其他功能可以继续使用 Node.js 服务器。
3. 我们使用开源软件 ffmpeg 来生成视频。ffmpeg 支持非常多的滤镜（filter），但是经过我们测试发现，在我们使用的腾讯云服务器上，一些的滤镜（例如 fade 和 blur）使用起来生成速度非常慢，10s 的视频需要大概 85s 左右的生成时间，用户体验非常不友好，所以我们最后只使用了很简单的滤镜。

团队情况

队员都为南方医科大学生物医学工程在读大三学生。

团队分工

代码：林培奇（主力）李泽臣 马勇杰（协力）

界面设计：李泽臣

文档编写：协力完成

测试：协力完成