

Technical Solution for Implementing Song Request in Traditional Broadcast Systems Based on E-Learning Tablet

[Yuzhou Tan]

Abstract

By connecting the e-learning system of the school with the system on the cloud and changing the traditional pattern of playing songs on the broadcast, the e-learning system of the school, the broadcast control system and the third-party music platform on the Internet are connected, and the online campus radio station is integrated to build the online song ordering website, so as to implement the interactive pattern of users selecting songs, nominating songs and ranking website on the e-learning tablets and playing the songs on the campus broadcast at regular schedule.



Keyword: E-learning Tablet, Broadcast System, Interactive Song Request

Yuzhou Tan

Senior 3 AP student of Guangzhou Foreign Language School, who has strong passion towards computer programming and internet technologies, and has several project experiences upon video processing, smart home and other applications.

技术
交流

1 Introduction

校园广播是每所学校不可缺少的基础设施之一，通常由学校职工或学生广播站负责管理和运营，做为学校师生获取信息和通知的重要渠道，主要用于多种活动，如播报新闻、发布通知、升国旗、课间操等；同时，校园广播还可以播放音乐、广播节目等，为学生们提供学习之余的娱乐和休闲。

目前传统的校园广播系统通常由管理人员手动操控广播系统的方式播放固定的音乐，操作繁琐，容易出现误时误点的播放。本文以广州外国语学校广播系统为开发对象，基于原生的HTML+CSS+JS，将安装学生平板或电脑上的数字化教学系统、校园广播系统以及互联网的第三方音乐平台集成，搭建用户互动的线上点歌台，使得同学们通过学习平板、电脑等设备进行热门歌曲搜索和歌曲提名，点

歌台根据歌曲提名排行榜的情况定时通过校园广播系统播放歌曲，一方面减少了广播系统的日常人工介入的操作运维，同时使点歌管理系統化、自动化和互动化，也使同学们紧张学习之余的娱乐活动变得更加丰富多彩。

2 Overview

通常校园广播台播放是由学校广播站管理人员选择具体内容播放，通过人工控制音乐或歌曲的播放次序和时间，播放内容相对固定和有限，而本方案中线上点歌台通过使用RESTful API技术，实现了学生使用在学校数字化教学系统终端上开发的前端应用，连接互联网第三方音乐平台，挑选感兴趣或者热门的歌曲进行试听和点歌提名。线上点歌台通过对提名的音乐和歌曲建立排行榜，对排行榜中的音乐和歌曲按点歌量等算法选择排序，在预设规定的时间段内自动通过校园广播系统进行播放，满足学生对歌曲选

择多样化和互动化的需求。

3 Planning

3.1 Overall Structure

点歌台总共由六部分组成，包括线上点歌网页前端，线上点歌数据中心，学校数字化教学系统，校园广播控制台，校园广播服务器以及第三方音乐平台，如图 1 所示。

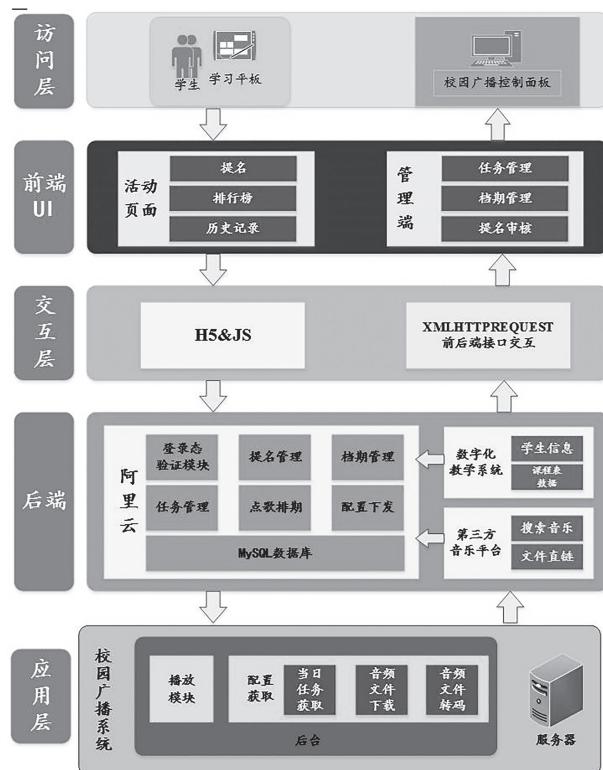


图 1 线上点歌台总体架构示意图

线上点歌网页前端主要实现面向学生及管理人员的服务开放；线上点歌数据中心主要向点歌前端及控制台提供数据库服务；学校数字化教学系统主要实现为点歌网页开通入口，提供学生用户信息开放能力；校园广播控制台主要实现动态配置更新及并定时控制能力；校园广播服务器主要提供音频播放能力；第三方音乐平台主要提供音乐搜索及文件下载能力。

3.2 Modules

(1) 线上点歌网页前端：采用 HTML5 技术，主要包括供学生使用的点歌活动页面和供校园广播系统管理人员使用的管理页面。点歌活动页面分为三大板块，分别为

音乐提名板块、提名排行榜板块和历史提名板块。管理页面分为三大板块，分别为广播任务管理板块、点歌排期管理板块和点歌提名审核板块，如图 2 所示。

用户交互活动页面是通过 iframe 嵌入至学校数字化教学系统内，使用纯原生 HTML+CSS+JS 搭建网页，通过 XMLHTTPREQUEST 实现 RESTful API 接口。



图 2 线上点歌网页前端内容

(2) 线上点歌数据中心：数据中心部署在阿里云服务器上，主要包括网页后端、控制台配置管理、周期性自动化排行榜结果排期。使用 WAMP (Windows+Apache+MySQL+PHP)。控制台配置管理负责通知校园控制台当天的播放内容、音频文件数据。

(3) 学校数字化教学系统：装载在学生平板电脑上的学校数字化教学系统，通过 WebView 向学生提供点歌页面入口，使学生能够在学习平板电脑上访问线上点歌网页，并通过 JSBridge 打通网页前端获取学生信息的接口。

(4) 校园广播后台：主要实现动态从线上点歌数据中心获取当天广播任务，并通过校园广播系统服务开放能力，定时控制学校广播服务器播放音频文件。

广播控制：由于因历史原因校内共设有两套广播控制系统，一套负责校园一期工程建筑内的旧广播终端，另一套负责二期工程建筑内的新广播终端，需采用不同技术对

接后再实现对两套系统的同步控制。

根据调研，一期工程的广播系统提供一套基于 Java 的广播控制程序，在使用 JADX 对其 jar 软件包进行逆向工程分析后，得知其使用 java.io.net 和序列化传输实现 TCP 控制协议和 UDP 音频流式传输协议，故依照其代码重写一个支持命令行执行播放控制的 Java 程序，以备后期调用；

二期工程的广播系统并未提供客户端，而是转而使用网页端进行控制，在使用 Fiddler 抓取到后端 RESTful API 接口后，使用 Python 的 requests 库重新封装，同时编写一个支持命令行执行上传音频文件、播放控制的 Python 程序；

通过 Python 实现配置更新获取，同时调用 ffmpeg 对音频文件进行码率压缩、响度均衡处理；使用 Windows 任务计划及批处理实现定时执行广播任务。

(5) 校园广播服务器：主要提供音频播放开放能力。

(6) 第三方音乐平台：提供音乐搜索及文件下载开放能力。

4 Implementation

4.1 Main Page

用户在领启公司开发的学生平板领启系统上点击进入活动页面入口，WebView 加载文章详情页，请求学生平板从领启系统的后端获取文章数据，领启系统后端返回带有 XSS 注入代码的文章数据，领启系统前端开始渲染文章内容并注入 JS 代码生效，清空 HTML body 内容，插入 iFrame 元素，WebView 加载活动页面，活动页面触发 onLoad 事件，开始页面初始化，请求领启系统后端检查登录态，领启系统后端返回用户信息，如果未登录或登录

态失效返回 403，活动页面请求领启系统后端获取领启课表，后端再返回课表数据，如果未登录或登录态失效返回 403，活动页面使用 localStorage 存储课表数据，执行 intro 动画，最后 WebView 进入活动页面，展示给用户交互，如图 3 所示。

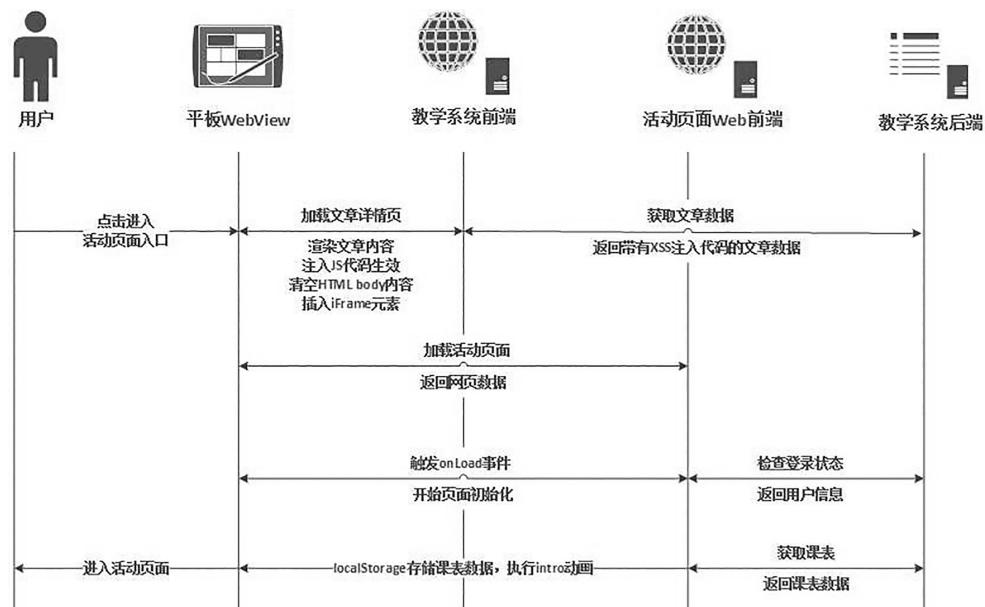


图 3 用户进入活动页面的流程

4.2 Nomination

用户通过 WebView 点击“我要提名”卡片进入活动页面，在活动页面中触发 click 事件，进入提名页面。提名页面会触发 onLoad 事件，进行内部初始化并获取点歌时间段，然后解析任务列表并插入下拉框，等待用户选择。用户输入歌曲关键词搜索后，会触发输入框 input、compositionstart、compositionend 事件，系统会将搜索关键词和平台 ID 传入阿里云后端，请求音乐平台接口，获取搜索结果，并格式化数据展示在 WebView 上。用户选择搜索结果项后，音乐会播放。最后，用户双重确认提交后，系统会将数据 PUT 到阿里云后端。阿里云后端会根据用户 ID、当前周期时间段、提名状态为正在审核或已通过审核，检索该用户提名记录数量，并判断是否超过当前周期可提名次数。如果没有超过，则会插入提名数据到 MySQL 中。返回执行结果后，WebView 会检查执行结果，如果成功，则跳回主界面，返回给用户，如图 4 所示。

» Technological Communication

技术交流

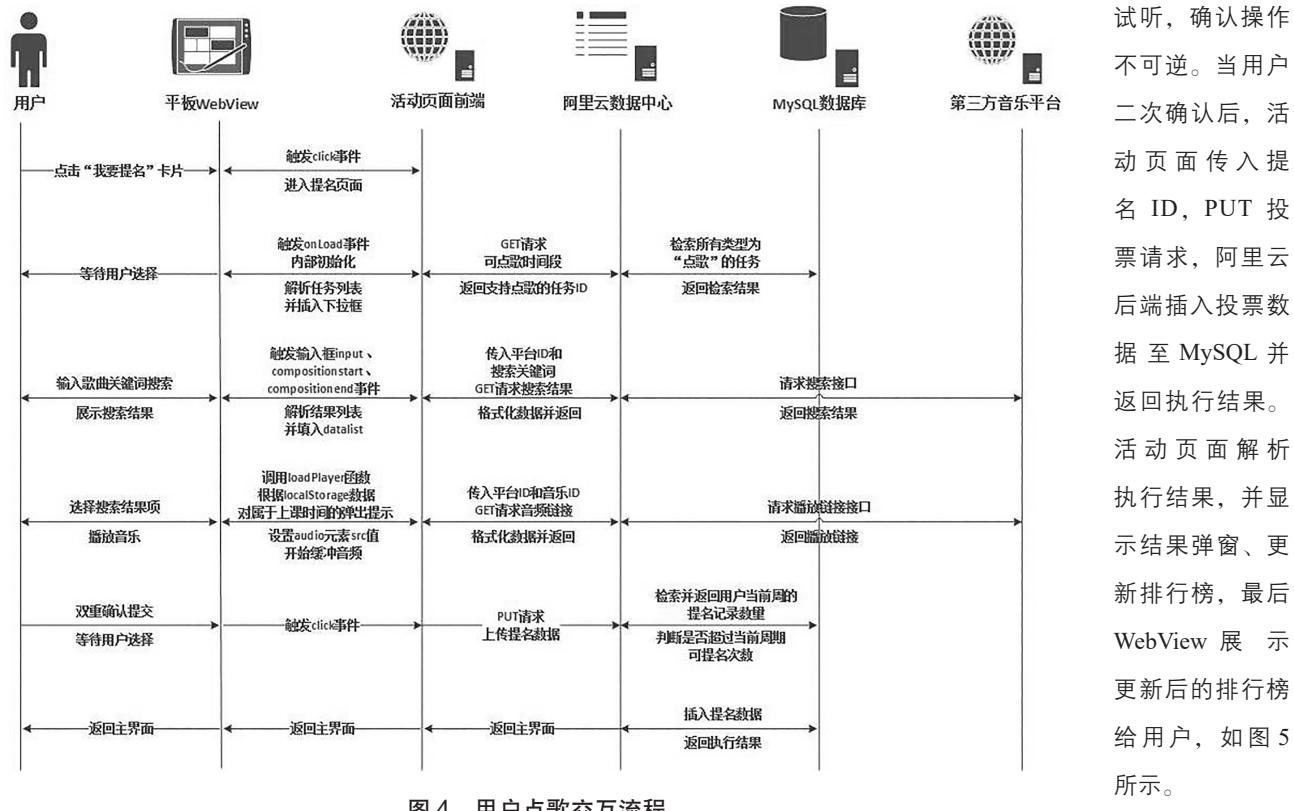


图 4 用户点歌交互流程

4.3 Ranking

用户进入 WebView，点击“歌曲榜单”卡片，触发 click 事件进入活动页面。在活动页面内部初始化，获取点歌时间段，解析任务列表并插入选项卡至页面，默认选择第一个任务。当用户划到页面最底部时，触发 Intersection-Observer 加载提名

榜单。当用户点击试听时，触发 click 事件，音乐播放。

用户点击投票时，触发 click 事件，活动页面检查是否已

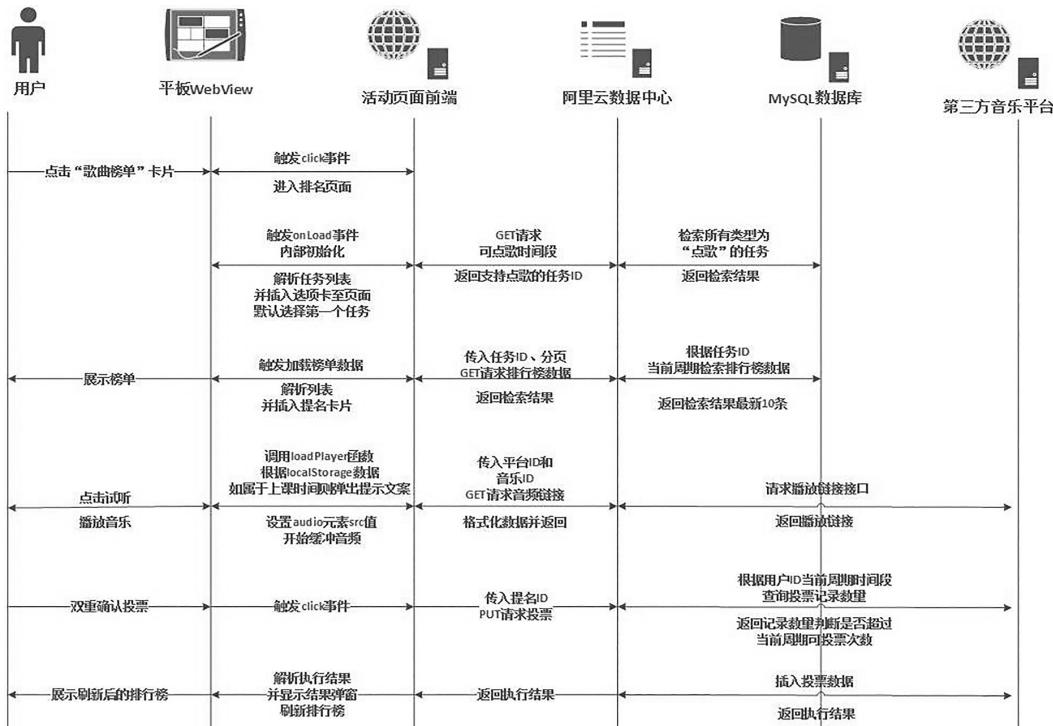


图 5 提名排行榜交互流程

4.4 Loading Time Slot for Playing

活动页面向阿里云后端发送 GET 请求，获取可点歌

时间段。阿里云后端通过 MySQL 检索所有类型为“点歌”的任务，并返回检索结果给阿里云后端。最后，阿里云后端将支持点歌时段的任务 ID 返回给活动页面。

4.5 Loading Ranking List

活动页面向阿里云后端发送 GET 请求，传入任务 ID

和分页，获取排行榜数据。阿里云后端使用 MySQL 根据任务 ID 和当前周期的开始结束时间检索排行榜数据，并返回检索结果的最新十条数据。阿里云后端将检索结果返回给活动页面，活动页面使用 WebView 解析列表并插入提名卡片，如图 6 所示。

操作	状态	审核意见	详情	档期
通过 试听 拒绝	未审核		2025-3 郑真 Death Cab For Cutie - Summer Skin 原因: yeeeeee~	GWITC.2.午间.2.点歌
通过 试听 拒绝	通过		2025-融创1 王睿 贾年糕 - homeless (无家可归) 原因: 小贾: 支持	GWITC.2.午间.2.点歌
通过 试听 拒绝	通过		2025-融创1 庄子越 邦宇Tristan - 【原神】日间登录曲 (竖琴+长笛简单翻奏片段) 原因: 求求了优悠别查了	GWITC.1.早间.3.36分
通过 试听 拒绝	通过		2025-融创1 庄子越 武青圣 - 原神登录BGM (Genshin Impact Main Theme) 原因: 求求了别查了	GWITC.1.早间.3.36分
通过 试听 拒绝	通过		2025-3 洪嘉颖 KeShi - more 原因: keshi! ! !	GWITC.3.下午.1.点歌
			2025-3 洪嘉颖	

图 6 提名歌曲审核页面

4.6 Song Playback

WebView 调用 loadPlayer 函数，根据 localStorage 数据判断是否属于上课时间，如果是，弹出提示文案，活动页面向阿里云后端传入平台 ID 和音乐 ID 并发起 GET 请求获取音频链接，阿里云后端向音乐平台接口请求播放链

接，音乐平台接口返回播放链接，阿里云后端格式化数据并返回给活动页面，活动页面将数据传给 WebView，WebView 设置 audio 元素的 src 值，开始缓冲音频，最后播放音乐，如图 7 所示。



图 7 点歌计划安排页面

5 Summary and Insights

通过在学校内教学系统与互联网系统的连通，改变传统广播歌曲播放模式，需要将教学系统、广播系统、第三方音乐平台集成搭建校园线上点歌台。在此基础上才能实

现用户在学校数字化教学系统终端上挑选歌曲、点歌提名、展现点歌台排行榜和校园广播定时播放的互动模式，增加了广播系统播放内容的新颖、热门内容，提高同学们互动参与度，也能反映同学们对校园生活的热爱。

(Received Date: 2023-02-01)

基于教学终端实现传统广播系统的互动点歌技术方案

[谭禹舟]

摘要

通过学校内教学系统与互联网系统的连通，改变传统广播歌曲播放模式，将学校数字化教学系统、广播系统以及互联网第三方音乐平台连通，集成搭建校园线上点歌台，实现用户在学校数字化教学系统终端上挑选歌曲、点歌提名和点歌台排行榜，再通过校园广播定时播放歌曲的用户互动模式。



关键词：教学终端 广播系统 互动点歌

谭禹舟

广州市外国语学校高三 AP 班学生，对计算机编程和互联网网络技术有深度的喜爱，在自媒体视频处理、智能生活等应用有多个项目研发经验。

技术
交流

1 前言

校园广播是每所学校不可缺少的基础设施之一，通常由学校职工或学生广播站负责管理和运营，做为学校师生获取信息和通知的重要渠道，主要用于多种活动，如播报新闻、发布通知、升国旗、课间操等；同时，校园广播还可以播放音乐、广播节目等，为学生们提供学习之余的娱乐和休闲。

目前传统的校园广播系统通常由管理人员手动操控广播系统的方式播放固定的音乐，操作繁琐，容易出现误时误点的播放。本文以广州外国语学校广播系统为开发对象，基于原生的 HTML+CSS+JS，将安装学生平板或电脑上的数字化教学系统、校园广播系统以及互联网的第三方音乐平台集成，搭建用户互动的线上点歌台，使得同学们通过学习平板、电脑等设备进行热门歌曲搜索和歌曲提名，点

歌台根据歌曲提名排行榜的情况定时通过校园广播系统播放歌曲，一方面减少了广播系统的日常人工介入的操作运维，同时使点歌管理系统化、自动化和互动化，也使同学们紧张学习之余的娱乐活动变得更加丰富多彩。

2 线上点歌台概述

通常校园广播台播放是由学校广播站管理人员选择具体内容播放，通过人工控制音乐或歌曲的播放次序和时间，播放内容相对固定和有限，而本方案中线上点歌台通过使用 RESTful API 技术，实现了学生使用在学校数字化教学系统终端上开发的前端应用，连接互联网第三方音乐平台，挑选感兴趣或者热门的歌曲进行试听和点歌提名。线上点歌台通过对提名的音乐和歌曲建立排行榜，对排行榜中的音乐和歌曲按点歌量等算法选择排序，在预设规定的时间段内自动通过校园广播系统进行播放，满足学生对歌曲选

择多样化和互动化的需求。

3 线上点歌台设计方案

3.1 总体架构

点歌台总共由六部分组成，包括线上点歌网页前端，线上点歌数据中心，学校数字化教学系统，校园广播控制台，校园广播服务器以及第三方音乐平台，如图 1 所示。

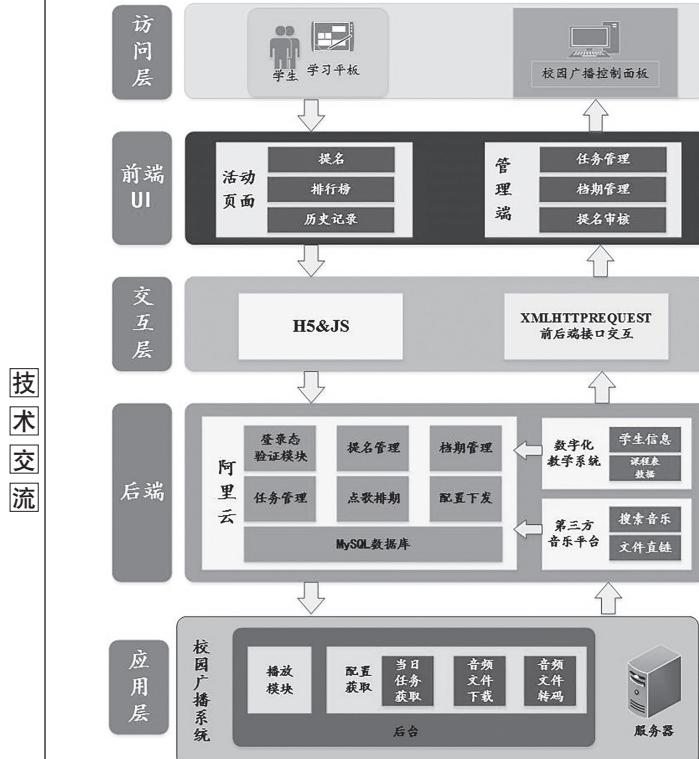


图 1 线上点歌台总体架构示意图

线上点歌网页前端主要实现面向学生及管理人员的服务开放；线上点歌数据中心主要向点歌前端及控制台提供数据库服务；学校数字化教学系统主要实现为点歌网页开通入口，提供学生用户信息开放能力；校园广播控制台主要实现动态配置更新及并定时控制能力；校园广播服务器主要提供音频播放能力；第三方音乐平台主要提供音乐搜索及文件下载能力。

3.2 功能模块

(1) 线上点歌网页前端：采用 HTML5 技术，主要包括供学生使用的点歌活动页面和供校园广播系统管理人员使用的管理页面。点歌活动页面分为三大板块，分别为

音乐提名板块、提名排行榜板块和历史提名板块。管理页面分为三大板块，分别为广播任务管理板块、点歌排期管理板块和点歌提名审核板块，如图 2 所示。

用户交互活动页面是通过 iframe 嵌入至学校数字化教学系统内，使用纯原生 HTML+CSS+JS 搭建网页，通过 XMLHTTPREQUEST 实现 RESTful API 接口。



图 2 线上点歌网页前端内容

(2) 线上点歌数据中心：数据中心部署在阿里云服务器上，主要包括网页后端、控制台配置管理、周期性自动化排行榜结果排期。使用 WAMP (Windows+Apache+MySQL+PHP)。控制台配置管理负责通知校园控制台当天的播放内容、音频文件数据。

(3) 学校数字化教学系统：装载在学生平板电脑上的学校数字化教学系统，通过 WebView 向学生提供点歌页面入口，使学生能够在学习平板电脑上访问线上点歌网页，并通过 JSBridge 打通网页前端获取学生信息的接口。

(4) 校园广播后台：主要实现动态从线上点歌数据中心获取当天广播任务，并通过校园广播系统服务开放能力，定时控制校园广播服务器播放音频文件。

广播控制：由于因历史原因校内共设有两套广播控制系统，一套负责校园一期工程建筑内的旧广播终端，另一套负责二期工程建筑内的新广播终端，需采用不同技术对

接后再实现对两套系统的同步控制。

根据调研，一期工程的广播系统提供一套基于 Java 的广播控制程序，在使用 JADX 对其 jar 软件包进行逆向工程分析后，得知其使用 java.io.net 和序列化传输实现 TCP 控制协议和 UDP 音频流式传输协议，故依照其代码重写一个支持命令行执行播放控制的 Java 程序，以备后期调用；

二期工程的广播系统并未提供客户端，而是转而使用网页端进行控制，在使用 Fiddler 抓取到后端 RESTful API 接口后，使用 Python 的 requests 库重新封装，同时编写一个支持命令行执行上传音频文件、播放控制的 Python 程序；

通过 Python 实现配置更新获取，同时调用 ffmpeg 对音频文件进行码率压缩、响度均衡处理；使用 Windows 任务计划及批处理实现定时执行广播任务。

(5) 校园广播服务器：主要提供音频播放开放能力。

(6) 第三方音乐平台：提供音乐搜索及文件下载开放能力。

4 核心功能技术实现方案

4.1 点歌台主界面

用户在领启公司开发的学生平板领启系统上点击进入活动页面入口，WebView 加载文章详情页，请求学生平板从领启系统的后端获取文章数据，领启系统后端返回带有 XSS 注入代码的文章数据，领启系统前端开始渲染文章内容并注入 JS 代码生效，清空 HTML body 内容，插入 iFrame 元素，WebView 加载活动页面，活动页面触发 onLoad 事件，开始页面初始化，请求领启系统后端检查登录态，领启系统后端返回用户信息，如果未登录或登录

态失效返回 403，活动页面请求领启系统后端获取领启课表，后端再返回课表数据，如果未登录或登录态失效返回 403，活动页面使用 localStorage 存储课表数据，执行 intro 动画，最后 WebView 进入活动页面，展示给用户交互，如图 3 所示。

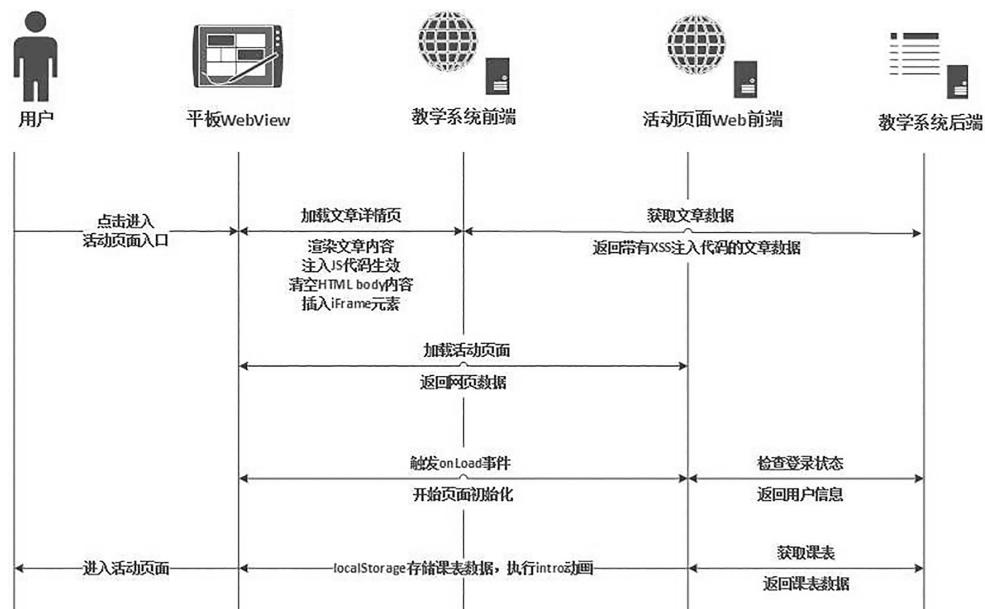


图 3 用户进入活动页面的流程

4.2 音乐提名

用户通过 WebView 点击“我要提名”卡片进入活动页面，在活动页面中触发 click 事件，进入提名页面。提名页面会触发 onLoad 事件，进行内部初始化并获取点歌时间段，然后解析任务列表并插入下拉框，等待用户选择。用户输入歌曲关键词搜索后，会触发输入框 input、compositionstart、compositionend 事件，系统会将搜索关键词和平台 ID 传入阿里云后端，请求音乐平台接口，获取搜索结果，并格式化数据展示在 WebView 上。用户选择搜索结果项后，音乐会播放。最后，用户双重确认提交后，系统会将数据 PUT 到阿里云后端。阿里云后端会根据用户 ID、当前周期时间段、提名状态为正在审核或已通过审核，检索该用户提名记录数量，并判断是否超过当前周期可提名次数。如果没有超过，则会插入提名数据到 MySQL 中。返回执行结果后，WebView 会检查执行结果，如果成功，则跳回主界面，返回给用户，如图 4 所示。

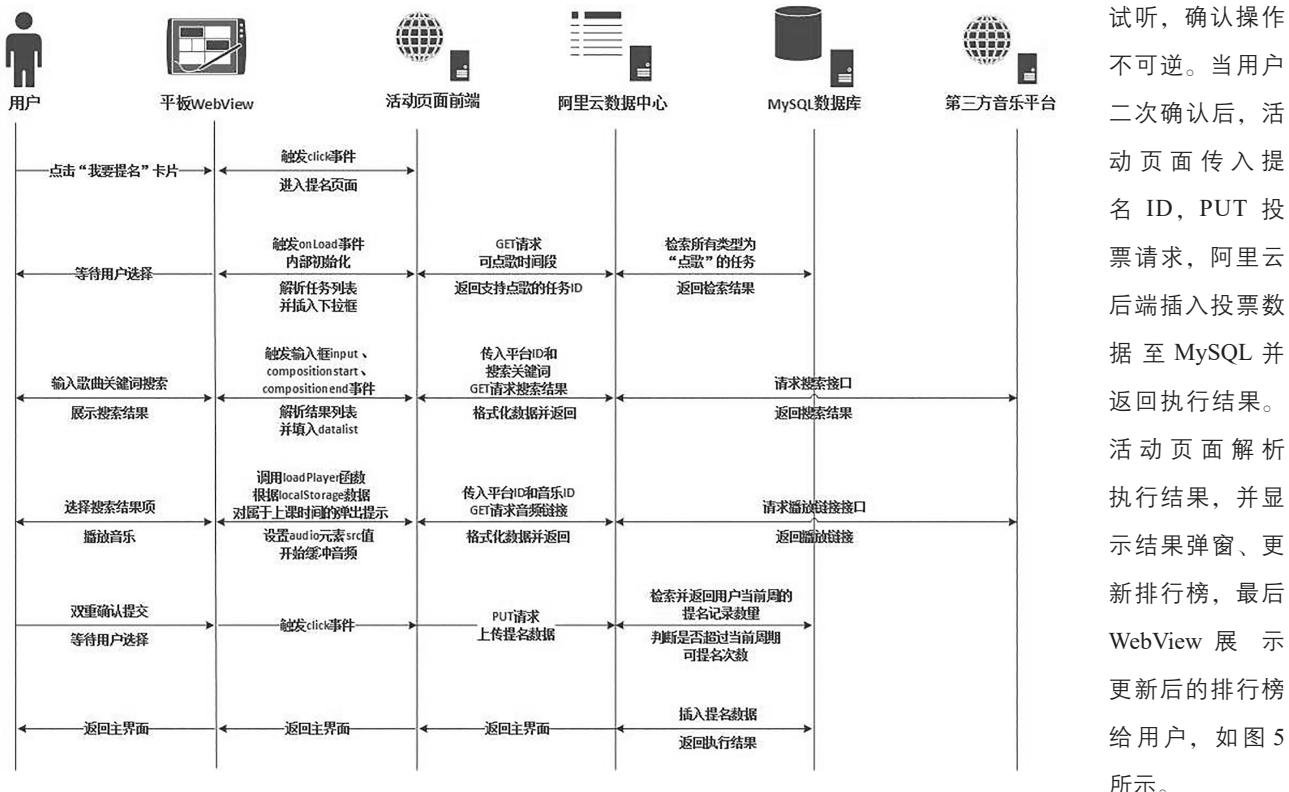


图 4 用户点歌交互流程

4.3 提名排行榜

用户进入 WebView，点击“歌曲榜单”卡片，触发 click 事件进入活动页面。在活动页面内部初始化，获取点歌时间段，解析任务列表并插入选项卡至页面，默认选择第一个任务。当用户划到页面最底部时，触发 Intersection-Observer 加载提名

榜单。当用户点击试听时，触发 click 事件，音乐播放。

用户点击投票时，触发 click 事件，活动页面检查是否已

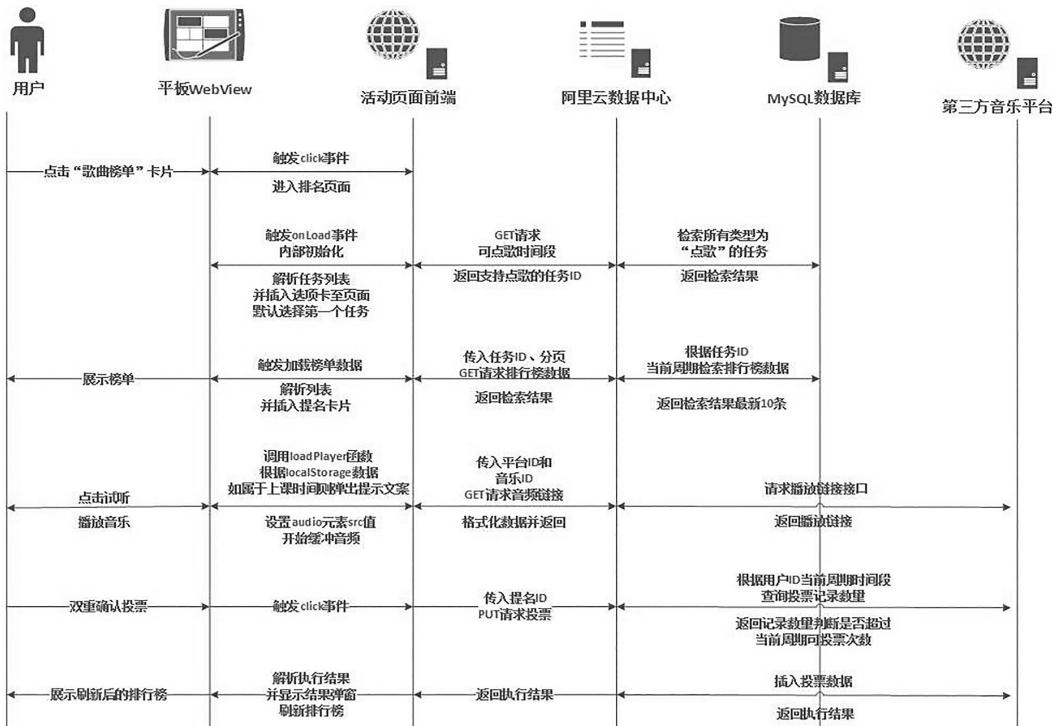


图 5 提名排行榜交互流程

4.4 获取点歌时间段

活动页面向阿里云后端发送 GET 请求，获取可点歌

时间段。阿里云后端通过 MySQL 检索所有类型为“点歌”的任务，并返回检索结果给阿里云后端。最后，阿里云后端将支持点歌时段的任务 ID 返回给活动页面。

4.5 加载提名榜单

活动页面向阿里云后端发送 GET 请求，传入任务 ID

和分页，获取排行榜数据。阿里云后端使用 MySQL 根据任务 ID 和当前周期的开始结束时间检索排行榜数据，并返回检索结果的最新十条数据。阿里云后端将检索结果返回给活动页面，活动页面使用 WebView 解析列表并插入提名卡片，如图 6 所示。

操作	状态	审核意见	详情	档期
通过 试听 拒绝	未审核		2023-3 郑真 Death Cab For Cutie - Summer Skin 原因: yeeeeee~	GWITC.2.午间.2.点歌
通过 试听 拒绝	通过		2025-融创1 王睿 贾年糕 - homeless (无家可归) 原因: 小贾: 支持	GWITC.2.午间.2.点歌
通过 试听 拒绝	通过		2025-融创1 庄子越 羽生Tristan - 【原神】日间登录曲 (竖琴+长笛简单翻奏片段) 原因: 求了求了优哉别走了	GWITC.1.早间.3.36分
通过 试听 拒绝	通过		2025-融创1 庄子越 武青圣 - 原神登录BGM (Genshin Impact Main Theme) 原因: 求了别走了	GWITC.1.早间.3.36分
通过 试听 拒绝	通过		2025-3 洪嘉颖 KeShi - more 原因: keshi ! !	GWITC.3.下午.1.点歌

图 6 提名歌曲审核页面

4.6 音乐播放

WebView 调用 loadPlayer 函数，根据 localStorage 数据判断是否属于上课时间，如果是，弹出提示文案，活动页面向阿里云后端传入平台 ID 和音乐 ID 并发起 GET 请求获取音频链接，阿里云后端向音乐平台接口请求播放链

接，音乐平台接口返回播放链接，阿里云后端格式化数据并返回给活动页面，活动页面将数据传给 WebView，WebView 设置 audio 元素的 src 值，开始缓冲音频，最后播放音乐，如图 7 所示。

图 7 点歌计划安排页面

5 小结与启示

通过在学校内教学系统与互联网系统的连通，改变传统广播歌曲播放模式，需要将教学系统、广播系统、第三方音乐平台集成搭建校园线上点歌台。在此基础上才能实

现用户在学校数字化教学系统终端上挑选歌曲、点歌提名、展现点歌台排行榜和校园广播定时播放的互动模式，增加了广播系统播放内容的新颖、热门内容，提高同学们互动参与度，也能反映同学们对校园生活的热爱。

(收稿日期：2023-02-01)