

《移动应用软件开发》期末软件设计报告

年级、专业、班级	2021级计算机科学与技术3班			姓名	刘成	学号	20215005
提交时间	2023. 12. 10	学年学期	2023-2024（1）		指导教师	杨瑞龙	
软件名称	“音之谷”音乐电台 APP						
<p>说明：</p> <p>1. 每人独立完成移动应用软件设计开发，并完成设计报告。</p> <p>2. 自拟软件题目。不准抄袭，应当是自己独立编写的软件。抄袭计 0 分。</p> <p>3. 软件开发平台选择 HarmonyOS，优先使用 ArkUI 开发。</p> <p>4. 从界面、技术、功能等方面对软件进行评价。程序有必要的注释。</p> <p>5. 从文档条理性、文档规范性、内容完整性等方面对设计报告进行评价。</p> <p>6. 所开发的程序源代码压缩成 ZIP 格式，命名为：学号姓名-期末软件设计.zip。报告命名为：学号姓名-期末软件设计报告.docx。演示视频清晰，MP4 格式，大小不超过 20M，命名为：学号姓名-期末软件设计演示.mp4。请在谷歌或者搜狗浏览器预览视频是否能够正常播放。三个文件分别提交到云班课。</p>							
<p>报告正文</p> <p>注：格式要规范，正文汉字一律使用宋体小四，文字行间距使用 1.5 倍行距。段首缩进 2 个汉字，图片居中，每个图片在底部有标题和编号。表格上面有表格标题和编号。版面漂亮、整洁、统一。不超过 20 页。</p>							
<p>一、软件设计意图、功能介绍和特色</p> <p>1. 软件设计意图：设计目的是创建一款用户友好、功能全面的音乐播放应用，旨在提供用户一个全面的音乐体验，通过直观的用户界面，使用户能够轻松地浏览、收听和收藏多种音乐。</p> <p>2. 功能介绍：（1）音乐按照不同的风格类型提供选择（2）不同歌单的简介信息（3）音乐播放器的播放功能【播放、暂停、上一首、下一首、单曲循环、随机播放】（4）查看歌词（5）查看音乐评论（6）收藏歌曲</p>							

二、开发和运行环境（可能包括开发平台、运行平台、开发工具、第三方组件使用情况、其他支撑软件运行的条件等）

开发平台：DevEco Studio 3.1 Release

运行平台：华为设备模拟器

其他运行条件：Stage 模型，支持 API version 9

三、设计说明（写明设计思想、程序的结构、功能设计、界面设计、模型设计、程序主要执行流程图，最后的核心源代码，截图等）

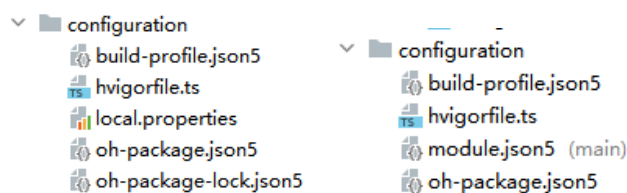
1. 设计思想

（1）用户导向设计：应用的设计以简洁直观的用户界面为基础，确保用户能够轻松地浏览、搜索和播放音乐。每个设计元素都着眼于提升用户体验，使用户可以在不费力的情况下享受到多样化的音乐。

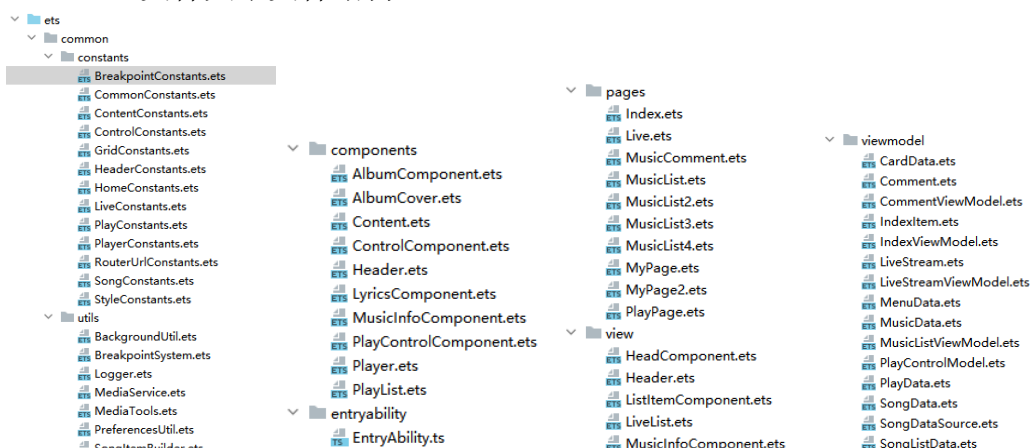
（2）多元化音乐内容：设计应用以容纳各种音乐类型、歌手和专辑，满足不同用户的音乐口味。通过分类让用户能够方便地找到并享受到他们感兴趣的音乐内容。

2. 程序的结构

（1）配置文件



（2）ets 文件夹中文件结构



3. 功能设计

(1) 音乐播放盒设计：根据华为开发者音视频开发指导，音乐类应用通过调用 JS 接口层提供的 AVPlayer 接口实现相应功能时，框架层会通过播放服务（Player Framework）将资源解析成音频数据流（PCM），音频数据流经过软件解码后输出至音频服务（Audio Framework），由音频服务输出至音频驱动渲染，实现音频播放功能。完整的音频播放需要应用、Player Framework、Audio Framework、音频 HDI 共同实现。AVPlayer 主要工作是将 Audio/Video 媒体资源转码为可供渲染的图像和可听见的音频模拟信号，并通过输出设备进行播放，同时对播放任务进行管理，包括开始播放、暂停播放、停止播放、释放资源、设置音量、跳转播放位置、获取轨道信息等功能控制。

(2) 后台播放设计：需要申请 `ohos.permission.KEEP_BACKGROUND_RUNNING` 权限，声明后台模式类型。在 `module.json5` 配置文件中为需要使用长时任务的 `UIAbility` 声明相应的长时任务类型。

4. 界面设计

使用线性布局、层叠布局、弹性布局等自适应布局，当外部容器大小发生变化时，元素可以根据相对关系自动变化以适应外部容器变化的布局能力。使用栅格布局的响应式布局当外部容器大小发生变化时，元素可以根据断点、栅格或特定的特征（如屏幕方向、窗口宽高等）自动变化以适应外部容器变化的布局能力。

5. 模型设计

表 1： 模型设计结构表

主体	功能
主菜单	选择歌曲风格、音乐视频和我的收藏 界面
歌单首页	展示歌单图片和文字介绍,展示个单 中歌曲曲目

歌曲	播放，歌词，评论，收藏
音乐播放盒	播放，暂停，上一首，下一首，随机播放，单曲循环，查看歌词

6. 程序主要执行流程图

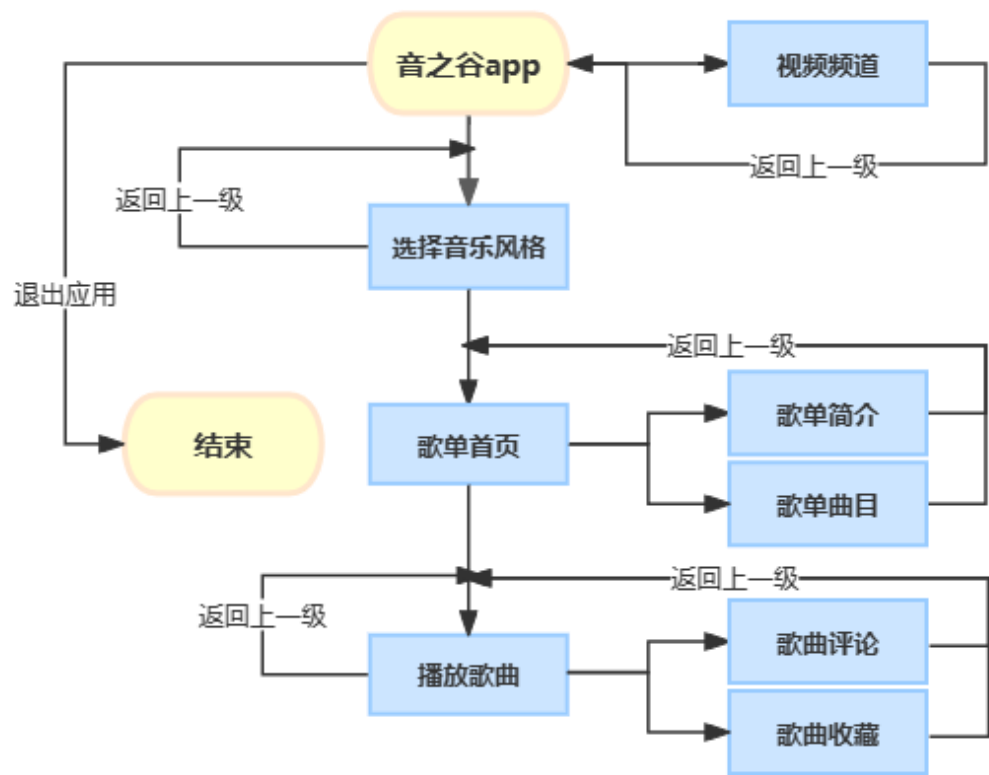


图 1：音之谷 app 程序主要流程图

7. 核心源代码

1. Index.ets

```
import router from '@ohos.router';
import IndexItem from '../viewmodel/IndexItem';
import IndexViewModel from '../viewmodel/IndexViewModel';
import { StyleConstants } from '../common/constants/StyleConstants';
import { BreakpointConstants } from '../common/constants/BreakpointConstants';
import { HomeConstants } from '../common/constants/HomeConstants';

@Entry
@Component
```

```
struct Index {
    @State indexItemList: IndexItem[] = IndexViewModel.getIndexItemList();

    build() {
        Scroll() {
            Column() {
                Column() {
                    Image($r('app.media.yinzhigu'))
                        .width("70%")
                    Text("最了解你的音乐电台")
                        .fontColor(Color.Gray)
                        .fontSize(13)
                }
                .width("100%")
                .height(120)
                .padding({
                    top: 20,
                    left: $r('app.float.column_padding_left'),
                    right: $r('app.float.column_padding_right'),
                })

                GridRow({
                    breakpoints: {
                        value: BreakpointConstants.BREAKPOINT_VALUE,
                        reference: BreakpointsReference.WindowSize
                    },
                    columns: {
                        sm: BreakpointConstants.COLUMN_SM,
                        md: BreakpointConstants.COLUMN_MD,
                        lg: BreakpointConstants.COLUMN_LG
                    },
                    gutter: { x: BreakpointConstants.GUTTER_X },
                    direction: GridRowDirection.Row
                }) {
                    GridCol({
                        span: {
                            sm: BreakpointConstants.SPAN_SM,
                            md: BreakpointConstants.SPAN_MD,
                            lg: BreakpointConstants.SPAN_LG
                        },
                        offset: {
                            md: BreakpointConstants.OFFSET_MD,
                            lg: BreakpointConstants.OFFSET_LG
                        }
                    }
                }
            }
        }
    }
}
```

```

    )) {
      Column({ space: HomeConstants.COLUMN_SPACE }) {
        ForEach(this.indexItemList, (item: IndexItem) => {
          Stack() {
            Image(item.icon)
              .width(StyleConstants.FULL_WIDTH)
              .height($r('app.float.item_height'))
              .borderRadius($r('app.float.item_border_radius'))
            Column() {
              Text(item.title)
                .fontSize($r('app.float.title_font_size'))
                .fontColor(Color.Black)
              Text(item.description)
                .fontSize(16)
                .opacity(HomeConstants.TEXT_OPACITY)
                .fontColor(Color.Black)
                .margin({
                  top: $r('app.float.description_margin_top')
                })
              Blank()
              Column() {
                Button() {
                  Text(item.button)
                    .fontSize($r('app.float.button_font_size'))
                    .fontColor(Color.White)
                }
                .backgroundColor("#708090")
                .borderRadius($r('app.float.button_border_radius'))
                .width($r('app.float.button_width'))
                .height($r('app.float.button_height'))
                .onClick(() => {
                  router.pushUrl({
                    url: item.url
                  }, router.RouterMode.Single);
                })
              }
              .alignItems(HorizontalAlign.End)
              .width(StyleConstants.FULL_WIDTH)
            }
            .width(StyleConstants.FULL_WIDTH)
            .height($r('app.float.item_height'))
            .borderRadius($r('app.float.item_border_radius'))
            .padding($r('app.float.item_padding'))
            .alignItems(HorizontalAlign.Start)
          }
        })
      }
    }
  }
}

```

```

        .justifyContent(FlexAlign.SpaceBetween)
    }
    }, (item: IndexItem, index?: number) => index +
JSON.stringify(item))
    }
    }
}

.padding({
    left: $r('app.float.column_padding_left'),
    right: $r('app.float.column_padding_right'),
    bottom: 12
})

Column()
    .width("100%")
    .height(20)
    .onClick(() => {
        router.pushUrl({
            url: "pages/MyPage2"
        })
    })
})

}

}

.width('100%')
.height(1000)
.scrollBar(BarState.Off)
}
}

```

2. PlayPage.ets

```

import router from '@ohos.router';
import { StyleConstants } from '../common/constants/StyleConstants';
import { BreakpointType } from '../common/utils/BreakpointSystem';
import { SongItem } from '../viewmodel/SongData';
import { MusicList } from '../viewmodel/MusicListViewModel';

import { ControlComponent } from '../components/ControlComponent';
import { LyricsComponent } from '../components/LyricsComponent';
import { MusicInfoComponent } from '../components/MusicInfoComponent';
import { PlayConstants } from '../common/constants/PlayConstants';

@Entry
@Component

```

```
struct PlayPage {
  @State currentTabIndex: number = 0;
  @StorageProp('currentBreakpoint') currentBreakpoint: string = 'sm';
  songList: SongItem[] = MusicList;
  @StorageProp('selectIndex') selectIndex: number = 0;

  build() {
    Stack({ alignContent: Alignment.TopStart }) {
      if (this.currentBreakpoint === PlayConstants.LG) {
        Stack({ alignContent: Alignment.Top }) {
          LyricsComponent()

          Flex({ direction: FlexDirection.Column }) {
            MusicInfoComponent()
            ControlComponent()
          }
        }
      } else {
        Flex({ direction: FlexDirection.Column }) {
          Tabs({ barPosition: BarPosition.Start, index: this.currentTabIndex })
        {
          TabContent() {
            MusicInfoComponent()
          }
          .tabBar(this.TabTitle(PlayConstants.TAB_SONG, 0))

          TabContent() {
            LyricsComponent()
          }
          .tabBar(this.TabTitle(PlayConstants.TAB_LYRICS, 1))
        }
        .onChange(index => this.currentTabIndex = index)
        .vertical(false)
        .barHeight($r('app.float.fifty_six'))
        .barWidth(PlayConstants.TAB_WIDTH)

        ControlComponent()
      }
    }

    Image($r('app.media.ic_back_down'))
      .width($r('app.float.image_back_size'))
      .height($r('app.float.image_back_size'))
      .margin({ left: $r('app.float.twenty_four'), top:
```



```

    $r('app.float.image_back_margin_top') ))
        .onClick(() => router.back())
    }
    .backgroundImage(this.songList[this.selectIndex].label)
    .backgroundImageSize(ImageSize.Cover)
    .backdropBlur(PlayConstants.BLUR)
    .linearGradient({
        direction: GradientDirection.Bottom,
        colors: [
            [PlayConstants.EIGHTY_WHITE_COLOR, PlayConstants.EIGHTY_WHITE],
            [PlayConstants.NINETY_WHITE_COLOR, PlayConstants.NINETY_WHITE]
        ]
    })
    .height(StyleConstants.FULL_HEIGHT)
    .width(StyleConstants.FULL_WIDTH)
}

@Builder
TabTitle(title: string | Resource, index: number) {
    Text(title)
        .fontColor(this.currentTabIndex === index ? $r('app.color.text_color') :
$r('app.color.text_forty_color'))
        .fontWeight(this.currentTabIndex === index ? PlayConstants.FIVE_HUNDRED :
PlayConstants.FOUR_HUNDRED)
        .fontSize(new BreakpointType({
            sm: $r('app.float.font_sixteen'),
            md: $r('app.float.font_twenty')
        })).getValue(this.currentBreakpoint))
        .border({
            width: { bottom: this.currentTabIndex === index ?
$r('app.float.tab_border_width') : 0 },
            color: $r('app.color.text_color')
        })
        .padding({ bottom: $r('app.float.tab_text_padding_bottom') })
    }

    pageTransition() {
        PageTransitionEnter({ duration: PlayConstants.FIVE_HUNDRED, curve:
Curve.Smooth }).slide(SlideEffect.Bottom);
        PageTransitionExit({ duration: PlayConstants.FIVE_HUNDRED, curve:
Curve.Smooth }).slide(SlideEffect.Bottom);
    }
}

```

3. MusicList.ets

```
import { BreakpointSystem } from '../common/utils/BreakpointSystem';
import { MediaService } from '../common/utils/MediaService';
import { StyleConstants } from '../common/constants/StyleConstants';
import { BreakpointConstants } from '../common/constants/BreakpointConstants';
import { SongConstants } from '../common/constants/SongConstants';
import { Header } from '../components/Header';
import { Player } from '../components/Player';
import { Content } from '../components/Content';

@Entry
@Component
struct MusicList {
    private breakpointSystem: BreakpointSystem = new BreakpointSystem();
    @State currentBreakpoint: string = BreakpointConstants.BREAKPOINT_SM;

    aboutToAppear() {
        MediaService.getInstance();
        this.breakpointSystem.register();
    }

    aboutToDisappear() {
        this.breakpointSystem.unregister();
    }

    build() {
        Stack({ alignContent: Alignment.Top }) {
            Header({ currentBreakpoint: $currentBreakpoint })
            Content({ currentBreakpoint: $currentBreakpoint })
            Player({ currentBreakpoint: $currentBreakpoint })
        }
        .width(StyleConstants.FULL_WIDTH)
        .backgroundColor(this.currentBreakpoint ===
BreakpointConstants.BREAKPOINT_SM ?
    $r('app.color.page_background_sm') : $r('app.color.page_background_other'))
    }

    pageTransition() {
        PageTransitionEnter({ duration: SongConstants.TRANSITION_DURATION, curve:
Curve.Smooth, type: RouteType.Pop })
        PageTransitionExit({ duration: SongConstants.TRANSITION_DURATION, curve:
Curve.Smooth, type: RouteType.Push })
    }
}
```

```
}
```

4. MusicComment.ets

```
import { BreakpointConstants } from '../common/constants/BreakpointConstants'
import { StyleConstants } from '../common/constants/StyleConstants'
import { Comment } from '../viewmodel/Comment';
import CommentViewModel from '../viewmodel/CommentViewModel';
import { ListItemComponent } from '../view/ListItemComponent';
import { HeadComponent } from '../view/HeadComponent';
import { MusicInfoComponent } from '../view/MusicInfoComponent';
import { CommonConstants } from '../common/constants/CommonConstants';

@Entry
@Component
struct MusicComment {
    @State currentBp: string = BreakpointConstants.CURRENT_BREAKPOINT;
    @State wonderfulComment: Comment[] = CommentViewModel.getWonderfulReview();
    @State newComment: Comment[] = CommentViewModel.getNewComment();

    @Builder
    ShowTitle(title: ResourceStr) {
        Row() {
            Text(title)
                .fontSize($r('app.float.comment_title_size'))
                .fontColor($r('app.color.comment_title_color'))
                .lineHeight($r('app.float.title_line_height'))
                .fontWeight(FontWeight.Medium)
                .margin({
                    top: $r('app.float.title_margin_top'),
                    bottom: $r('app.float.title_margin_bottom'),
                    left: this.currentBp === BreakpointConstants.BREAKPOINT_SM ?
                        $r('app.float.margin_left_sm') : $r('app.float.margin_left'),
                    right: this.currentBp === BreakpointConstants.BREAKPOINT_SM ?
                        $r('app.float.margin_right_sm') : $r('app.float.margin_right')
                })
        }
        .justifyContent(FlexAlign.Start)
        .width(StyleConstants.FULL_WIDTH)
    }

    build() {
        GridRow({
            breakpoints: {
```



```

        .margin({
            left: this.currentBp ===
BreakpointConstants.BREAKPOINT_SM ?
                0 : $r('app.float.margin_left_list'),
            right: this.currentBp ===
BreakpointConstants.BREAKPOINT_SM ?
                0 : $r('app.float.margin_right_list')
        })
    }
    .width(StyleConstants.FULL_WIDTH)
    .padding({
        bottom: $r('app.float.padding_bottom')
    })
}
} else {
    ListItem() {
        ListItemComponent({ item: comment })
        .margin({
            left: this.currentBp ===
BreakpointConstants.BREAKPOINT_SM ?
                0 : $r('app.float.margin_left_list'),
            right: this.currentBp ===
BreakpointConstants.BREAKPOINT_SM ?
                0 : $r('app.float.margin_right_list')
        })
    }
    .width(StyleConstants.FULL_WIDTH)
    .padding({
        bottom: $r('app.float.padding_bottom')
    })
}
}, (item: Comment, index?: number) => index + JSON.stringify(item))
}

.lanes(this.currentBp === BreakpointConstants.BREAKPOINT_LG ? 2 : 1)
.scrollBar(BarState.Off)
.divider({
    color: $r('app.color.list_divider'),
    strokeWidth: $r('app.float.stroke_width'),
    startMargin: this.currentBp === BreakpointConstants.BREAKPOINT_SM ?
$r('app.float.start_margin') : $r('app.float.start_margin_lg'),
    endMargin: this.currentBp === BreakpointConstants.BREAKPOINT_SM ?
        0 : $r('app.float.divider_margin_left')
})
.margin({

```

```

        left: this.currentBp === BreakpointConstants.BREAKPOINT_SM ?
        $r('app.float.margin_left_sm') : $r('app.float.margin_left_list'),
        right: this.currentBp === BreakpointConstants.BREAKPOINT_SM ?
        $r('app.float.margin_right_sm') : $r('app.float.margin_right_list')
    ))

    this.ShowTitle($r('app.string.new_comment'))

    List() {
        ForEach(this.newComment, (comment: Comment) => {
            ListItem() {
                ListItemComponent({ item: comment })
                    .margin({
                        left: this.currentBp === BreakpointConstants.BREAKPOINT_SM ?
                        0 : $r('app.float.margin_left_list'),
                        right: this.currentBp ===
BreakpointConstants.BREAKPOINT_SM ?
                        0 : $r('app.float.margin_right_list')
                    })
            }

            .width(StyleConstants.FULL_WIDTH)
            .padding({
                bottom: $r('app.float.padding_bottom')
            })
        }, (item: Comment, index?: number) => index + JSON.stringify(item))
    }

    .layoutWeight(1)
    .lanes(this.currentBp === BreakpointConstants.BREAKPOINT_LG ? 2 : 1)
    .scrollBar(BarState.Off)
    .margin({
        left: this.currentBp === BreakpointConstants.BREAKPOINT_SM ?
        $r('app.float.margin_left_sm') : $r('app.float.margin_left_list'),
        right: this.currentBp === BreakpointConstants.BREAKPOINT_SM ?
        $r('app.float.margin_right_sm') : $r('app.float.margin_right_list')
    })
    .divider({
        color: $r('app.color.list_divider'),
        strokeWidth: $r('app.float.stroke_width'),
        startMargin: this.currentBp === BreakpointConstants.BREAKPOINT_SM ?
        $r('app.float.start_margin') : $r('app.float.start_margin_lg'),
        endMargin: this.currentBp === BreakpointConstants.BREAKPOINT_SM ?
        0 : $r('app.float.divider_margin_left')
    })
}

```

```

        .height(StyleConstants.FULL_HEIGHT)
    }
}

.backgroundColor(Color.White)
.onBreakpointChange((breakpoint) => {
    this.currentBp = breakpoint;
}))
}
}

```

5. Live.ets

```

import { Header } from '../view/Header';
import { LiveList } from '../view/LiveList';
import { LiveConstants } from '../common/constants/LiveConstants';

@Entry
@Component
struct Live {
    build() {
        Column() {
            Header()
            LiveList()
        }

        .width(LiveConstants.FULL_WIDTH_PERCENT)
        .height(LiveConstants.FULL_HEIGHT_PERCENT)
    }
}

```

6. IndexViewModel.ets

```

import { SongItem } from './SongData';

const MusicList: SongItem[] = [
    { id: 1, title: '安河桥', singer: '宋东野', mark: $r('app.media.ic_vip'),
      label: $r('app.media.anheqiao'), src: 'anheqiaowave.mp3', index:0},
    { id: 2, title: '理想三旬', singer: '陈鸿宇', mark: $r('app.media.ic_sq'),
      label: $r('app.media.lixiang'), src: 'lixiangwav.mp3', index:1 },
    { id: 3, title: '杀死那个石家庄人', singer: '万能青年旅店', mark:
    $r('app.media.ic_vip'),
      label: $r('app.media.shasi'), src: 'shasiwave.mp3', index:2 },
    { id: 4, title: '南山南', singer: '马頔', mark: $r('app.media.ic_sq'),
      label: $r('app.media.nanshan'), src: 'nanshanwave.mp3', index:3},
    { id: 5, title: '写给黄淮', singer: '邵帅', mark: $r('app.media.ic_vip'),

```

```

    label: $r('app.media.ic_avatar5'), src: 'dynamic.wav', index:4 },
    { id: 6, title: '斑马斑马', singer: '宋东野', mark: $r('app.media.ic_sq'),
      label: $r('app.media.ic_avatar6'), src: 'boisterous.wav', index:5},
    { id: 7, title: '多想在平庸的生活拥抱', singer: '隔壁老樊', mark:
$r('app.media.ic_vip'),
      label: $r('app.media.ic_avatar17'), src: 'dynamic.wav', index:6 },
    { id: 8, title: '董小姐', singer: '宋东野', mark: $r('app.media.ic_sq'),
      label: $r('app.media.ic_avatar8'), src: 'boisterous.wav', index:7 },
    { id: 9, title: '乌拉巴托的夜', singer: '安来宁', mark: $r('app.media.ic_vip'),
      label: $r('app.media.ic_avatar9'), src: 'dynamic.wav', index:8 },
    { id: 10, title: '我记得', singer: '赵雷', mark: $r('app.media.ic_sq'),
      label: $r('app.media.ic_avatar10'), src: 'boisterous.wav', index:9 },
    { id: 11, title: '莉莉安', singer: '宋东野', mark: $r('app.media.ic_vip'),
      label: $r('app.media.ic_avatar11'), src: 'dynamic.wav', index:10},
    { id: 12, title: '胡广生', singer: '任素汐', mark: $r('app.media.ic_sq'),
      label: $r('app.media.ic_avatar12'), src: 'boisterous.wav', index:11 },
    { id: 13, title: '鼓楼', singer: '赵雷', mark: $r('app.media.ic_vip'),
      label: $r('app.media.ic_avatar13'), src: 'dynamic.wav', index:12 },
    { id: 14, title: '玫瑰', singer: '贰佰', mark: $r('app.media.ic_sq'),
      label: $r('app.media.ic_avatar14'), src: 'boisterous.wav', index:13 },
    { id: 15, title: '浪子回头', singer: '茄子蛋', mark: $r('app.media.ic_vip'),
      label: $r('app.media.ic_avatar15'), src: 'dynamic.wav', index:14 },
    { id: 16, title: '向云端', singer: '小霞、海洋 Bo', mark: $r('app.media.ic_sq'),
      label: $r('app.media.ic_avatar16'), src: 'boisterous.wav', index:15 },
    { id: 17, title: '春风十里', singer: '鹿先森乐队', mark: $r('app.media.ic_vip'),
      label: $r('app.media.ic_avatar17'), src: 'dynamic.wav', index:16 },
    { id: 18, title: '走马', singer: '陈粒', mark: $r('app.media.ic_sq'),
      label: $r('app.media.ic_avatar1'), src: 'boisterous.wav', index:17 },
    { id: 19, title: '去大理', singer: '郝云', mark: $r('app.media.ic_vip'),
      label: $r('app.media.ic_avatar2'), src: 'dynamic.wav', index:18 },
    { id: 20, title: '其实也一样', singer: '我是小树', mark: $r('app.media.ic_sq'),
      label: $r('app.media.ic_avatar13'), src: 'boisterous.wav', index:19 }
  ]

export { MusicList }

```

四、源程序调试过程（运行、调试截图和文字）

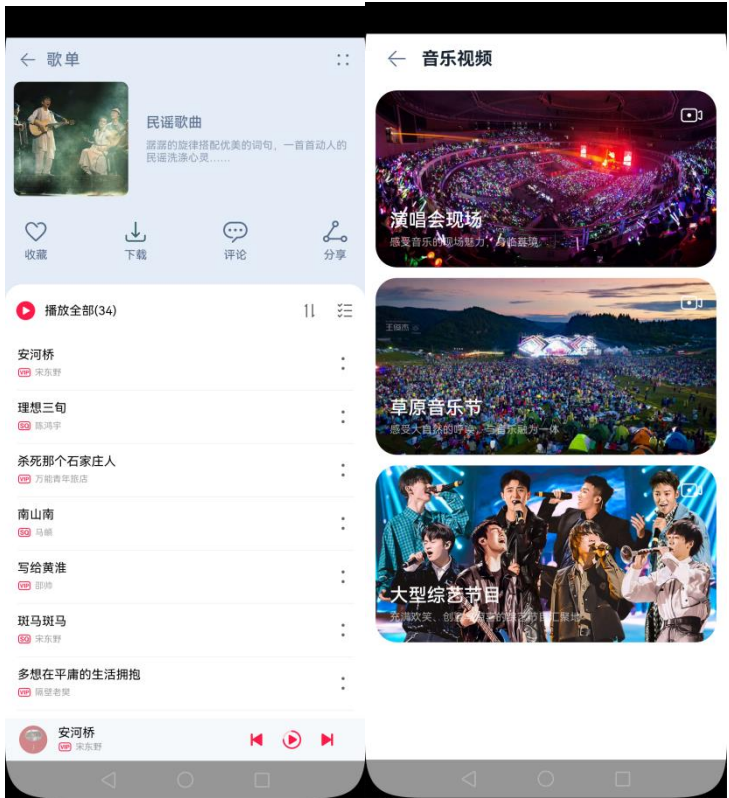
1. 应用首页展示

图 2 应用首页



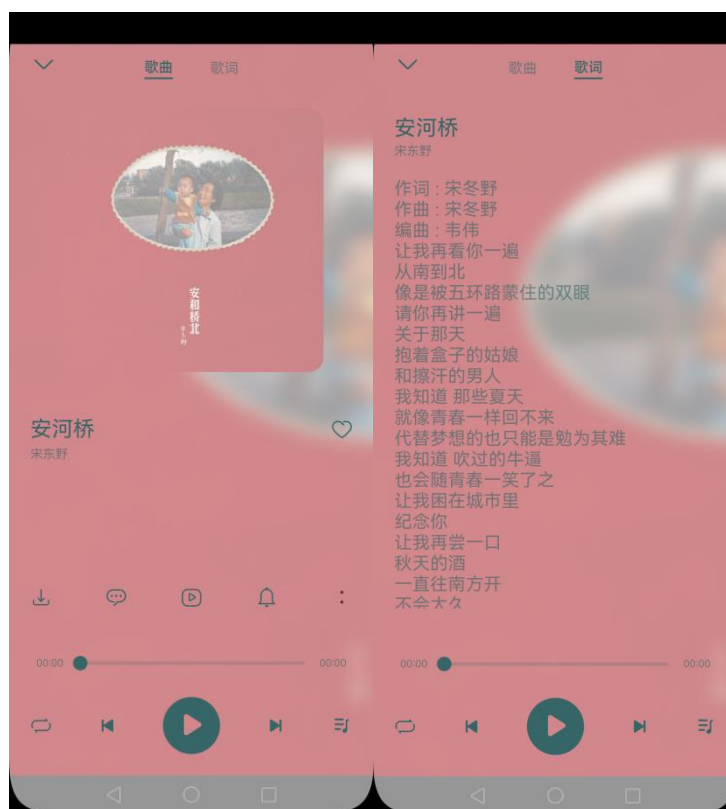
2. 歌单页面展示

图 3 歌单页



3. 播放页面展示

图 4 播放页



五、总结及分析（完成过程中的心得体会，成功与失败之处，经验，收获，建议、将来计划和展望）

在本次软件设计与开发过程中，我完成了“音之谷”音乐电台的移动应用软件的设计与开发，并撰写了相应的设计报告。这个 app 旨在为用户提供一款功能全面、用户友好的音乐播放应用，通过直观的用户界面，使用户能够轻松地浏览、收听和收藏多种音乐。

在这次软件设计过程中，我明确了用户导向的设计理念，以简洁直观的用户界面为基础，确保用户能够轻松地浏览、搜索和播放音乐。同时，通过多元化的音乐内容设计，满足不同用户的音乐口味，使用户能够方便地找到并享受到他们感兴趣的音乐内容。软件的功能介绍包括音乐按照不同的风格类型提供选择、不同歌单的简介信息、音乐播放器的基本功能（播放、暂停、上一首、下一首、单曲循环、随机播放）、查看歌词、查看音乐评论以及收藏歌曲等功能。

这次软件设计与开发过程是一次全面而深入的实践，不仅考验了我的编码和设计能力，还促使我更深入地了解 HarmonyOS 平台的开发特性。通

过对界面、技术、功能等方面的评价，我认为我在设计中注重了用户体验，同时保持了程序的结构清晰和模块化。源代码的注释也使得程序更易读懂。在文档方面，我遵循了规范，使得设计报告具备良好的条理性、规范性和完整性。整个设计过程为我提供了丰富的实践经验，让我更加熟悉移动应用的设计与开发流程，为今后的学习和工作打下了坚实的基础。