

# 重庆邮电大学计算机学院

## Show Time KTV （STV）

### 软件设计说明书

文件状态： [ ] 草稿 [✓] 正式发布 [ ] 正在修改	文件标识：	
	当前版本：	v0.91
	作 者：	封 莎 2014211574、符星宇 2014211471 黄浩翔 2014211420、王康宇 2014211408 张宝宇 2014211483、周冰玉 2014211553
	审 核：	周冰玉
	完成日期：	2016.12.5

版 本 历 史

版本/ 状态	作者	审核	起止日期	备注
草稿	封莎、符星宇、 张宝宇、王康宇	周冰玉	2016.11.10-11.20	起草版本
正在 修改	周冰玉、黄浩翔	周冰玉	2016.11.20-11.30	扩充内容
正式 发布	封莎、符星宇、 黄浩翔、王康宇、 张宝宇、周冰玉	封莎、符星宇、黄浩翔、 王康宇、张宝宇、周冰玉	2016.12.01-12.05	最终复查定稿

目 录

0. 文档介绍.....5

    0.1 文档目的与范围.....5

    0.2 读者对象.....5

    0.3 术语与缩写解释.....5

1. 系统概述.....6

    1.1 总体概述.....6

    1.2 软件功能.....6

2. 设计约束.....7

    2.1 运行环境的约束.....7

    2.2 性能需求.....7

        2.2.1 静态的量化需求.....7

        2.2.2 动态的量化需求.....7

    2.3 外部接口需求.....7

        2.3.1 用户接口.....7

        2.3.2 软件接口.....8

    2.4 总体设计约束.....10

        2.4.1 标准约束.....10

        2.4.2 技术约束.....10

        2.4.3 硬件约束.....10

    2.5 软件质量属性.....11

        2.5.1 可维护性.....11

        2.5.2 可靠性.....11

        2.5.3 安全性.....11

        2.5.4 易用性.....11

    2.6 其他需求.....11

3.开发、测试与运行环境.....11

4. 软件系统结构图.....12

    4.1 高层功能结构图.....12

    4.2 模块数据流程图.....13

    4.3 模块流图.....15

5. 功能模块设计概述.....16

    5.1 模块汇总.....16

    5.2 模块之间的关系.....17

        5.2.1 桌面客户端模块类图.....17

5.2.2 服务器端模块类图.....	17
5.2.3 移动客户端模块类图.....	18
5.3 命名规则.....	18
6. 数据库设计概述.....	19
6.1 数据库环境说明.....	19
6.2 数据库命名规则.....	19
6.3 安全性设计说明.....	19
6.4 表汇总和表设计.....	20
7. 用户界面设计概述.....	22
7.1 设计要素.....	22
7.2 注意事项.....	23
7.3 桌面客户端界面.....	23
7.3 移动客户端界面.....	25
8. 综合考虑.....	26
8.1 稳定性和可扩展性.....	26
8.1.1 不变需求.....	26
8.1.2 变化的需求.....	26
8.2 性能分析.....	26
8.2.1 用户需求的性能指标.....	26
8.2.2 实现性能指标.....	27
8.3 复用和移植.....	27
8.3.1 复用.....	27
8.3.2 移植.....	27
8.4 防错与出错处理.....	27
8.5 其它.....	27

## 0. 文档介绍

### 0.1 文档目的与范围

Show Time KTV 点歌系统是结合以往娱乐软件的开发经验,以娱乐业科学化规范化的管理为出发点而开发的视频歌曲点唱软件,该软件要实现的就是更加方便用户的使用和系统需求者的管理,增强娱乐性,力求为用户达到其预期的效果。

本文档为该系统的软件设计说明书,详细阐述了对用户所提出需求的设计方案,对系统中的各项功能需求、技术需求、实现环境及所使用的实现技术进行了明确定义。同时,对软件应具有的功能和性能及其他有效性需求也进行了定义。

### 0.2 读者对象

本文档预期读者是与 Show Time KTV 点歌系统软件开发有联系的决策人,开发组成人员,辅助开发者,支持本项目的老师和助教人员,软件验证者。

### 0.3 术语与缩写解释

缩写、术语	解 释
STV	Show Time KTV, 项目简称。
UI	User Interface, 用户直接可见的交互界面。
Json	JavaScript 文档对象格式。
GTK	GTK+为 Unix-like 系统下开发图形界面的应用程序的主流开发工具。

# 1. 系统概述

## 1.1 总体概述

本项目的名称是 Show Time KTV (STV) 点歌系统,要实现的就是更加方便用户的使用和系统需求者的管理,增强娱乐性,力求为用户达到其预期的效果。

桌面软件主要应用于 KTV 音乐会所的终端,也可以应用在 PC 端。

手机 APP 应用在使用桌面软件的用户手机上。

本系统包含三部分:桌面客户端,手机客户端,服务器端。

(1) 手机客户端可以从服务器获取当前桌面客户端的情况,以及上传歌曲元数据, MV 到服务器端。

(2) 桌面客户端可以从服务器端请求获取歌曲元数据,歌曲排行榜, MV 等。

(3) 服务器端负责桌面客户端与手机客户端之间的通信,获取到的歌曲元数据存储到数据库中, MV 文件存储到硬盘中。

## 1.2 软件功能

桌面客户端具备多种途径的歌曲查询点播功能(近期热榜、中文热榜、英文热榜、猜你喜欢、歌星点歌、音乐名点歌、心情点歌、随便来几首)并可以播放 MV。

桌面客户端对演唱列表可以进行顶歌(选择顶到第几)、删除、清空列表等操作。

支持手机客户端通过服务器传输歌曲(MV)到桌面客户端。

支持桌面客户端和手机客户端对正在播放的歌曲进行点赞评论。

## 2. 设计约束

### 2.1 运行环境的约束

本项目的桌面客户端依赖 Python 运行环境。

本项目的手机客户端依赖 Android 系统。

### 2.2 性能需求

#### 2.2.1 静态的量化需求

支持的终端数为  $n$  台 ( $n \geq 3$ )

处理  $n$  个文件及  $n$  条记录( $n \geq 100$ )

手机，桌面客户端最大并发量 10。

#### 2.2.2 动态的量化需求

网络环境较好时，并发量达到 8-9。

网络环境较差时，并发量至少达到 3。

### 2.3 外部接口需求

#### 2.3.1 用户接口

本产品的用户通过桌面终端或者手机终端进行操作,进入客户端主界面后点击相应的标签进入相应的界面。

## 2.3.2 软件接口

```
##BEGIN_SRC C
/* 请求URL: http://127.0.0.1/top/{type} */
/* 功能描述: 获取type类型的歌曲排行榜排行榜,例如获取中文榜单:http://127.0.0.1/top/zh */

/* 输入: 无 */
/* 输出: 无 */
/* 返回值: 返回相应榜单歌曲信息的JSON文件. */
/* 调用关系: 略 */
/* 其他: 无 */
##END_SRC

##BEGIN_SRC C
/* 请求URL: http://127.0.0.1/recommend/{type} */
/* 功能描述: 获取各类个性推荐榜单,例如获取猜你喜欢:http://127.0.0.1/recommend/guess */

/* 输入: 无 */
/* 输出: 无 */
/* 返回值: 返回个性化推荐榜单的JSON文件. */
/* 调用关系: 略 */
/* 其他: 无 */
##END_SRC

##BEGIN_SRC C
/* 请求URL: http://127.0.0.1/search/{way}/{type}/{content} */
/* 功能描述: 通过搜索方法way按搜索类型type搜索内容content.例如按拼音缩写搜索歌星林俊杰:http://127.0.0.1/search/abbreviation/singer/ljj */

/* 输入: 无 */
/* 输出: 无 */
/* 返回值: 返回搜索结果JSON文件. */
/* 调用关系: 略 */
/* 其他: 无 */
##END_SRC

##BEGIN_SRC C
/* 请求URL: http://127.0.0.1/download/{id} */
/* 功能描述: 获取指定ID的歌曲MV媒体文件 */

/* 输入: 无 */
/* 输出: 无 */
/* 返回值: 返回指定ID的歌曲MV文件. */
/* 调用关系: 略 */
/* 其他: 无 */
##END_SRC

##BEGIN_SRC C
/* 请求URL: http://127.0.0.1/upload */
/* 功能描述: 移动客户端上传歌曲MV文件,以及相应歌曲描述信息 */

/* 输入: MV描述信息表单,MV媒体文件. */
/* 输出: 无 */
/* 返回值: 上传结果. */
/* 调用关系: 略 */
/* 其他: 无 */
##END_SRC

##BEGIN_SRC C
/* 请求URL: http://127.0.0.1/control/link/{id} */
/* 功能描述: 移动客户端请求控制指定ID的桌面客户端 */

/* 输入: 无 */
/* 输出: 无 */
/* 返回值: 返回是否能进行控制 */
/* 调用关系: 略 */
/* 其他: 无 */
##END_SRC

##BEGIN_SRC C
/* 请求URL: http://127.0.0.1/playing/fetch/{cid} */
/* 功能描述: 获取指定ID的客户端的正在播放歌单 */
```



```
##+BEGIN_SRC C
/* 请求URL: http://127.0.0.1/playing/fetch/{cid} */
/* 功能描述: 获取指定ID的客户端的正在播放歌单 */

/* 输入: 无 */
/* 输出: 无 */
/* 返回值: 返回正在播放的歌单列表的JSON文件。 */
/* 调用关系: 略 */
/* 其他: 无 */
##+END_SRC

##+BEGIN_SRC C
/* 请求URL: http://127.0.0.1/playing/{operation}/{cid}/{sid} */
/* 功能描述: 客户端新增,删除一首歌.例如新增一首歌曲:http://127.0.0.1/playing/add/00001/00003 */

/* 输入: 无 */
/* 输出: 无 */
/* 返回值: 返回是否操作成功 */
/* 调用关系: 略 */
/* 其他: 无 */
##+END_SRC

##+BEGIN_SRC C
/* 请求URL: http://127.0.0.1/resort/{cid}/{sid}/{rank} */
/* 功能描述: 调整一首歌的播放顺序.例如将00003歌曲顶到第一位:http://127.0.0.1/resort/00001/00003/1 */

/* 输入: 无 */
/* 输出: 无 */
/* 返回值: 返回是否调整成功 */
/* 调用关系: 略 */
/* 其他: 无 */
##+END_SRC
```

```
##+BEGIN_SRC C
/* 请求URL: http://127.0.0.1/history/fetch/{cid} */
/* 功能描述: 请求获取制定ID的客户端的历史播放歌单 */

/* 输入: 无 */
/* 输出: 无 */
/* 返回值: 返回历史播放歌单列表的JSON文件。 */
/* 调用关系: 略 */
/* 其他: 无 */
##+END_SRC

##+BEGIN_SRC C
/* 请求URL: http://127.0.0.1/history/add/{cid}/{sid} */
/* 功能描述: 客户端演唱完一首歌 */

/* 输入: 无 */
/* 输出: 无 */
/* 返回值: 返回是否添加成功 */
/* 调用关系: 略 */
/* 其他: 无 */
##+END_SRC

##+BEGIN_SRC C
/* 请求URL: http://127.0.0.1/comment/add/{sid} */
/* 功能描述: 向指定歌曲添加一条评论 */

/* 输入: 评论表单 */
/* 输出: 无 */
/* 返回值: 返回是否添加成功 */
/* 调用关系: 略 */
/* 其他: 无 */
##+END_SRC
```

```
/*+BEGIN_SRC C
/* 请求URL: http://127.0.0.1/comment/fetch/{sid} */
/* 功能描述: 获取制定歌曲的评论信息 */

/* 输入: 无 */
/* 输出: 无 */
/* 返回值: 返回评论JSON文件 */
/* 调用关系: 略 */
/* 其他: 无 */
#+END_SRC

/*+BEGIN_SRC C
/* 请求URL: http://127.0.0.1/phrase/add/{sid}/{cid}/{count} */
/* 功能描述: 向指定客户端的当前演唱歌曲点赞数量加count */

/* 输入: 无 */
/* 输出: 无 */
/* 返回值: 返回是否点赞成功 */
/* 调用关系: 略 */
/* 其他: 无 */
#+END_SRC

/*+BEGIN_SRC C
/* 请求URL: http://127.0.0.1/phrase/fetch/{sid}/{cid} */
/* 功能描述: 获取客户端当前演唱歌曲的点赞数目 */

/* 输入: 无 */
/* 输出: 无 */
/* 返回值: 返回点赞数目JSON文件 */
/* 调用关系: 略 */
/* 其他: 无 */
#+END_SRC

/*+BEGIN_SRC C
/* 请求URL: http://127.0.0.1/phrase/{sid}/{cid}/{count} */
/* 功能描述: 向指定客户端的当前演唱歌曲点赞数量加count */

/* 输入: 无 */
/* 输出: 无 */
/* 返回值: 返回是否点赞成功 */
/* 调用关系: 略 */
/* 其他: 无 */
#+END_SRC
```

## 2.4 总体设计约束

### 2.4.1 标准约束

该软件的开发完全按照华为标准开发，包括硬件、软件和文档规格。

### 2.4.2 技术约束

本项目的设计是在汉语程序化语言的条件下进行的技术采用软硬件一体化的设计方法。无特别技术限制。

### 2.4.3 硬件约束

桌面客户端：CPU 主频 2.5GHz， 内存 4GB。标准输入键盘，鼠标。

手机客户端：普通智能安卓手机。

服务器端：CPU 主频 3.0GHz，内存 8GB。千兆以太网接口。

## 2.5 软件质量属性

### 2.5.1 可维护性

该系统开放源代码，采用 GPLv2 许可证发布源码，可维护性健壮。

### 2.5.2 可靠性

该系统为学习样例，低并发量下可稳定运行。无法可靠对外运行。

### 2.5.3 安全性

该系统获取权限低，不涉及客户隐私，安全性极高。

### 2.5.4 易用性

易懂性：界面区域名称易懂并且易于区分，用词准确，能达到望文知意。

易学性：会汉语、懂计算机的基本操作便可快速学会使用该系统。

易操作性：以用户为中心的设计理念，友好的用户交互界面，易于操作。

## 2.6 其他需求

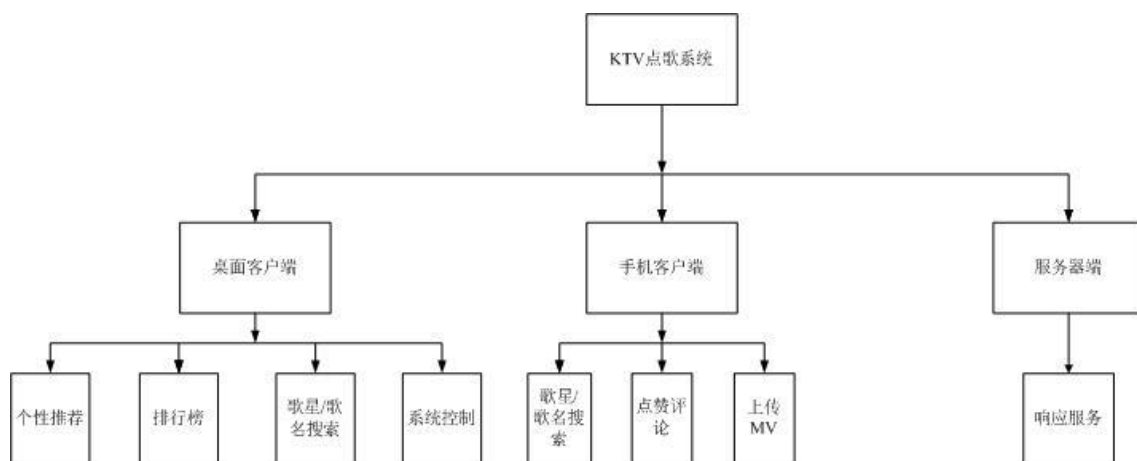
没有其他需求。

## 3.开发、测试与运行环境

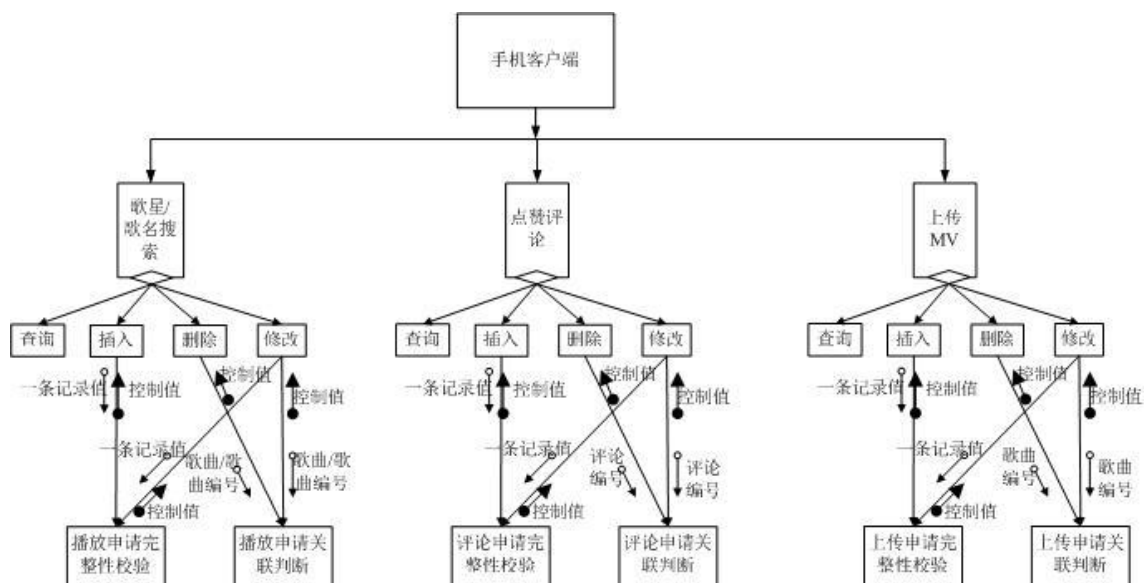
类别	标准配置	最低配置
开发环境	Intel® Core™ i5-4210U CPU @ 1.70GHz × 4	Intel® Core™ i3-4210U CPU
	Antergos Linux 64-bit	Ubuntu 16.04 64-bit
	Memory 3.8 GiB	Memory 1.0 GiB
测试环境	Intel® Core™ i5-4210U CPU @ 1.70GHz × 4	Intel® Core™ i3-4210U CPU
	OS Antergos Linux 64-bit	Ubuntu 16.04 64-bit
	Memory 3.8 GiB	Memory 1.0 GiB
运行环境	Intel® Core™ i5-4210U CPU @ 1.70GHz × 4	Intel® Core™ i3-4210U CPU
	OS Antergos Linux 64-bit	Ubuntu 16.04 64-bit
	Memory 3.8 GiB	Memory 1.0 GiB

#### 4. 软件系统结构图

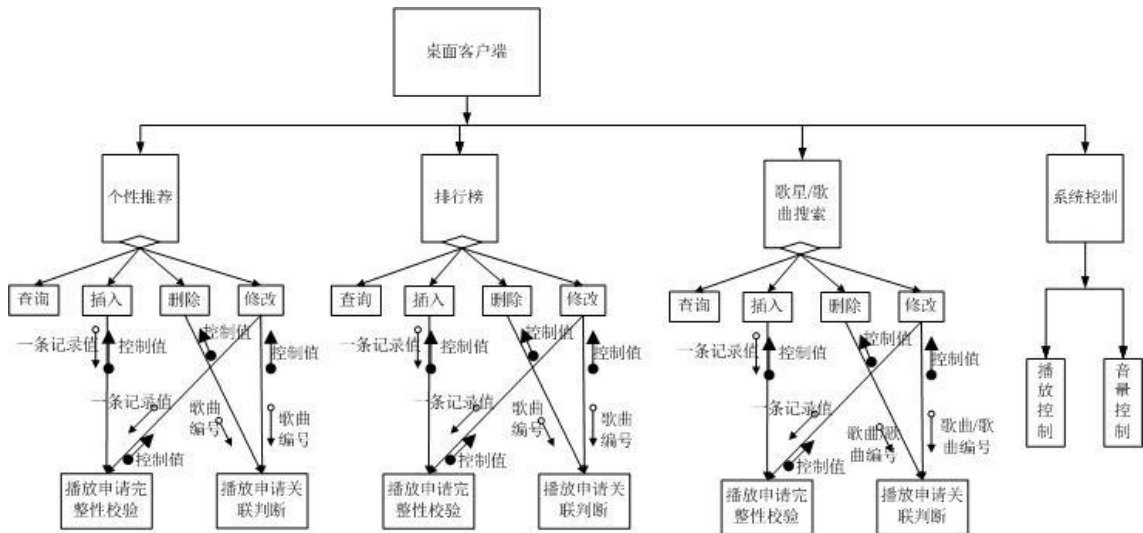
## 4.1 高层功能结构图



### 高层功能模块图

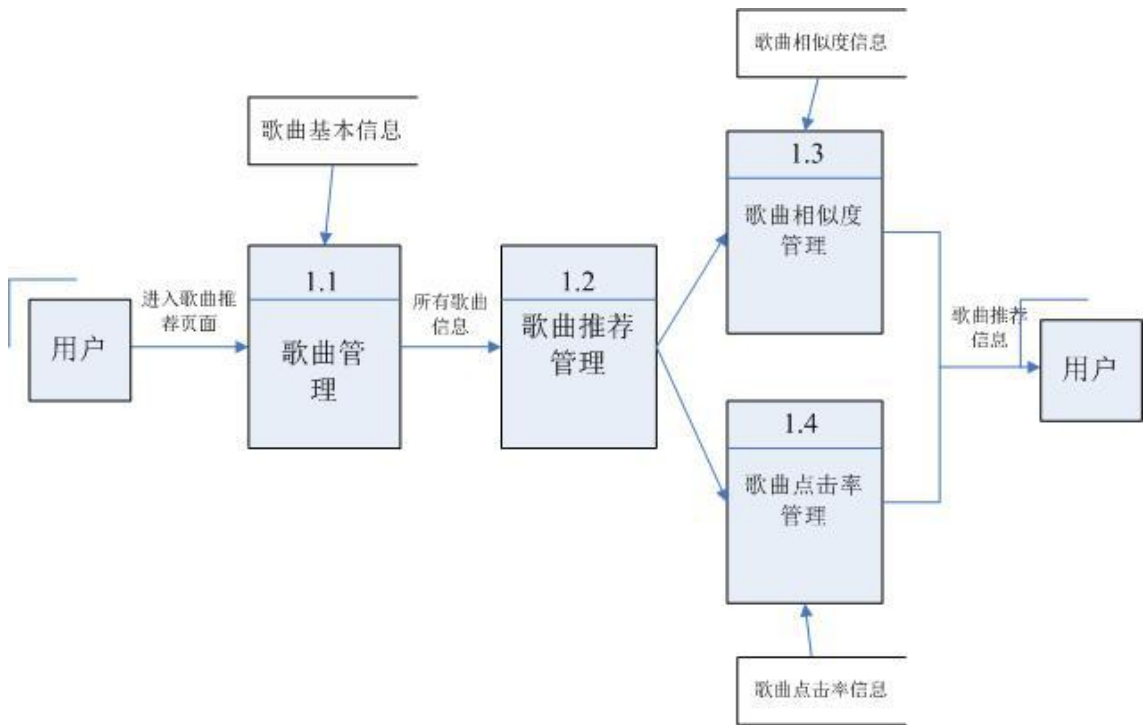


### “手机客户端”详细设计结构图

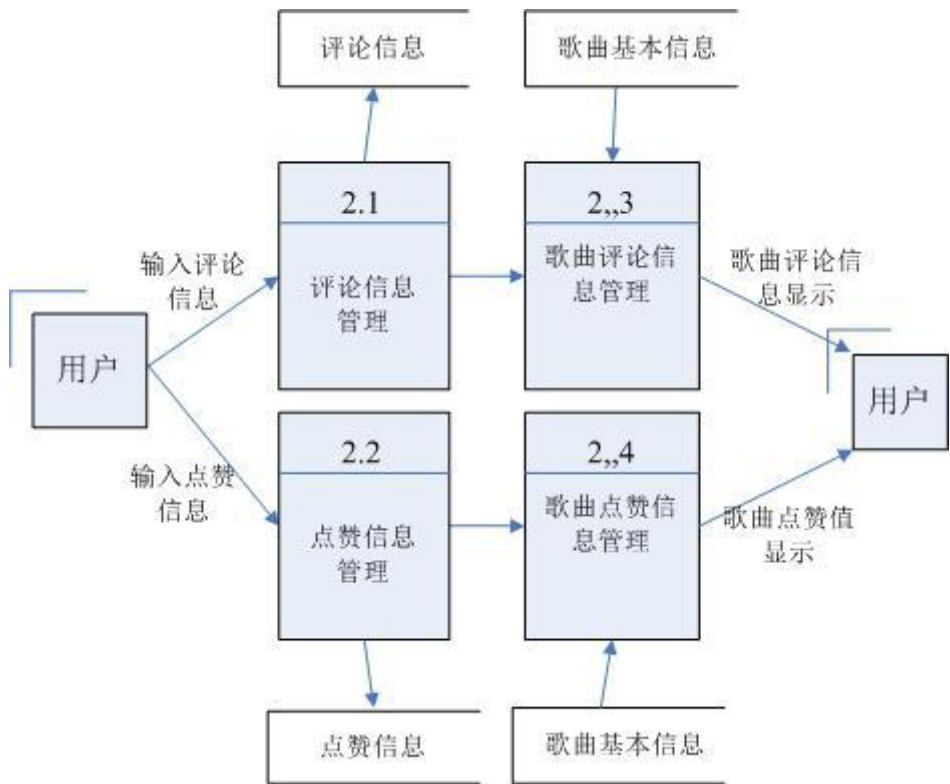


“桌面客户端”详细设计结构图

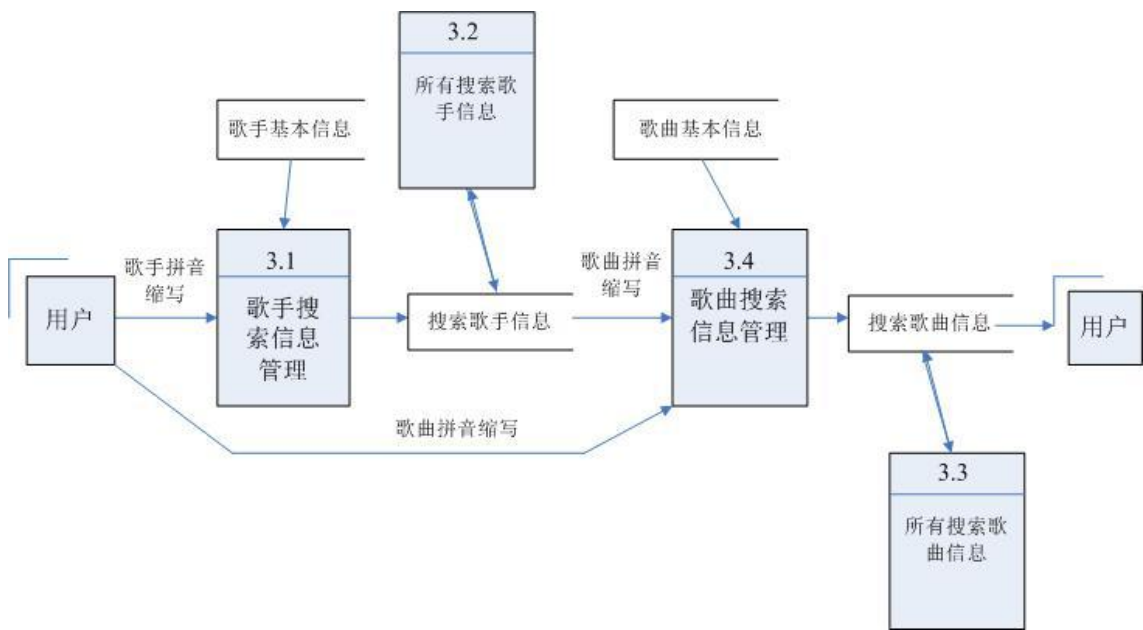
4.2 模块数据流程图



歌曲推荐模块数据流程图

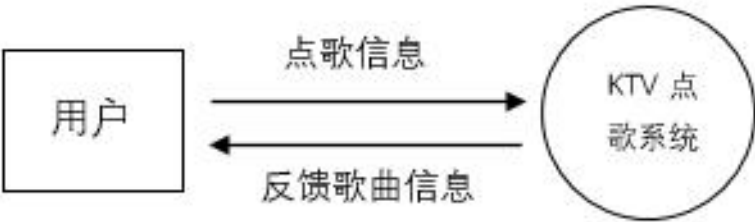


评论点赞模块数据流程图

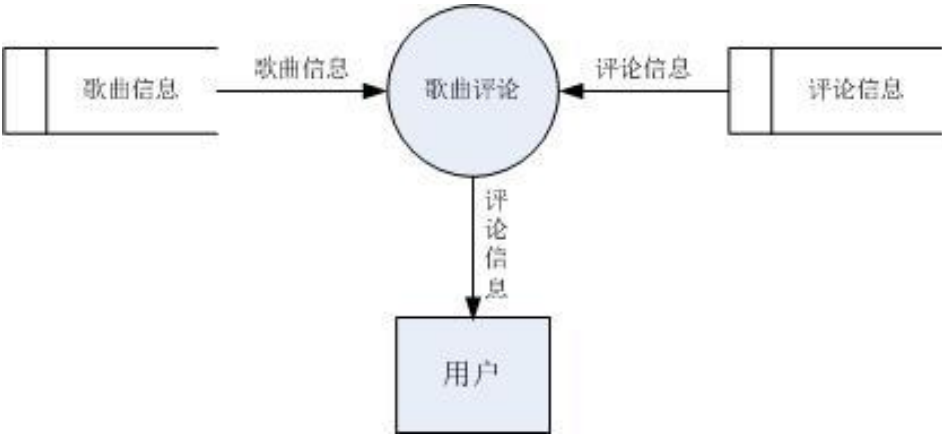


点歌模块数据流程图

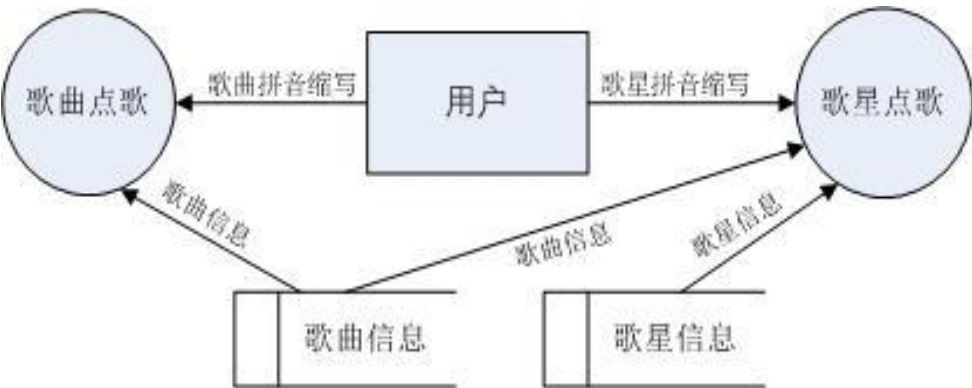
4.3 模块流图



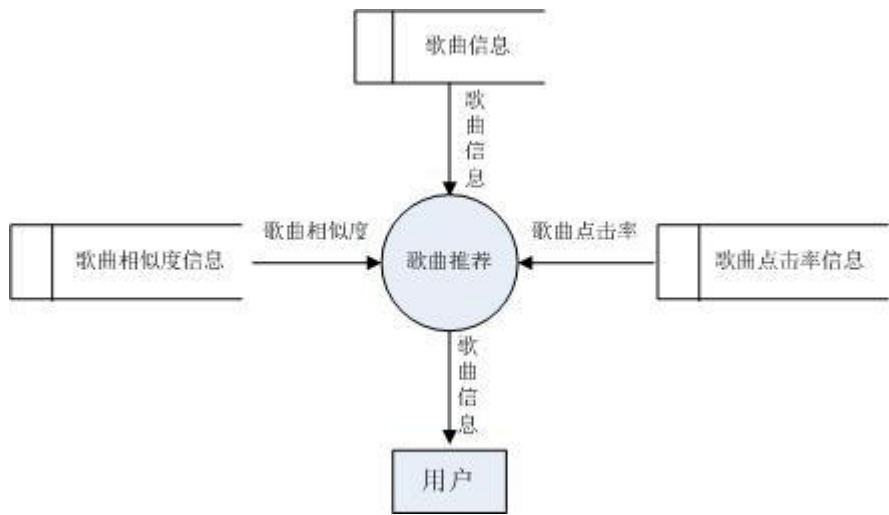
顶层数据模块流图



歌曲评论模块数据流图



点歌模块数据流图



歌曲推荐模块数据流程图

## 5. 功能模块设计概述

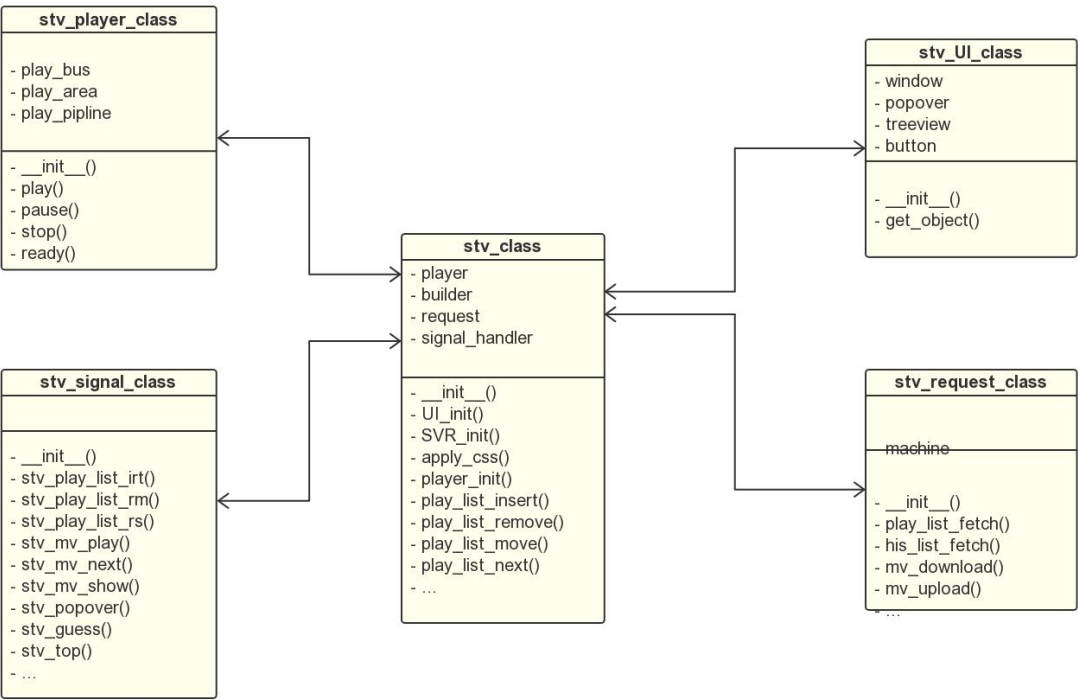
### 5.1 模块汇总

桌面客户端	
模块名称	功能概述
视频播放器类	实现媒体文件的播放、以及播放控制功能。
UI 控制类	实现 UI 的控制、页面的交互、逻辑数据与用户之间的沟通。
后台服务请求类	实现桌面客户端与后台服务器的通信功能。
后台服务端	
模块名称	功能概述
URL 类	后台服务器负责处理前台到来的 URL 请求。
Server 类	URL 类解析 URL 后分析出具体请求，调用 Server 类进行相应服务。
Database 类	后台与数据库交换数据的抽象。
Recommendation 类	推荐系统相关类。
移动客户端	
模块名称	功能概述
Fragment 类	定义主界面 UI，包含各子 UI 类。
Adapter 类	为某些特定 UI 设置适配器。
JSONParse 类	解析来自服务器的 JSON 数据包。
SVRConnection 类	与服务器建立连接，通过该连接接受服务器传来的数据。

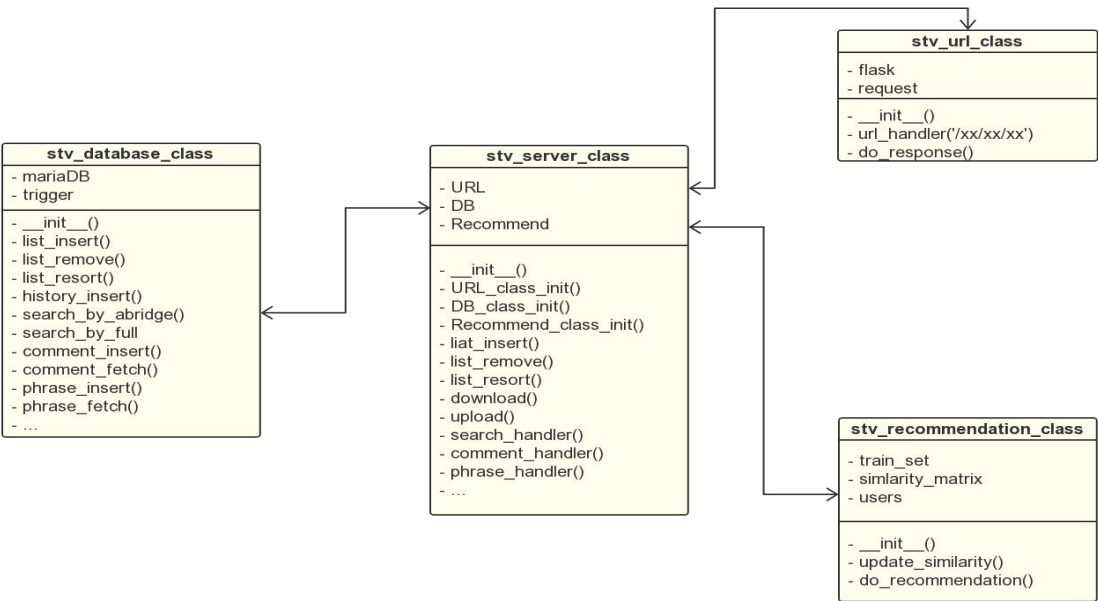


## 5.2 模块之间的关系

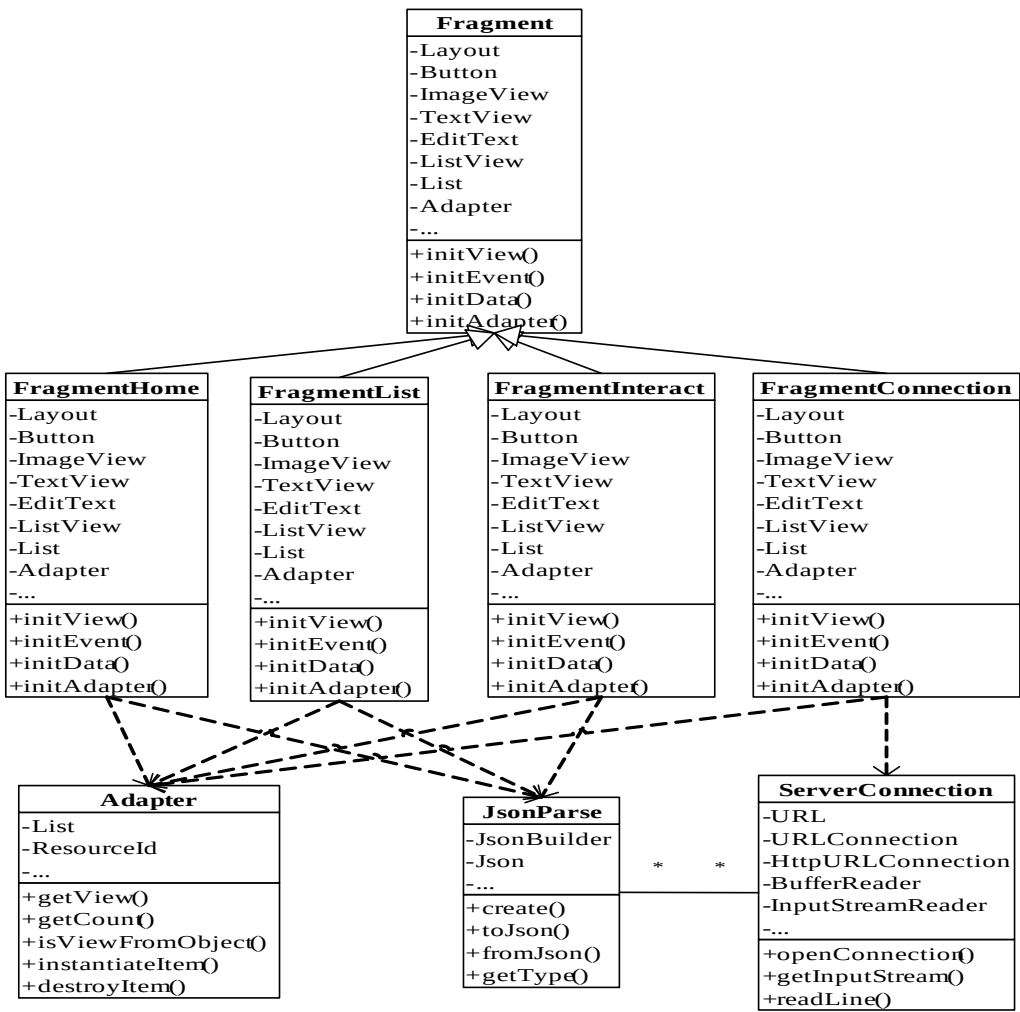
### 5.2.1 桌面客户端模块类图



### 5.2.2 服务器端模块类图



5.2.3 移动客户端模块类图



5.3 命名规则

类型	命名规则 （Linux 内核命名法）
函数名	类_实体_方法（）：player_play();player_pause()
变量	小写英文字母，数字，下划线：window;popover;
源文件名	小写英文字母，数字，下划线：stv.py
宏名	大写英文字母，数字，下划线：LIBS; PATH

## 6. 数据库设计概述

### 6.1 数据库环境说明

本软件采用 MYSQL 数据库系统，因为 MYSQL 可以处理拥有上千万条记录的大型数据，拥有广泛的应用程序支持，并且支持常见的 SQL 语句规范，可移植行高，安装简单小巧，良好的运行效率，有丰富信息的网络支持，调试、管理，优化简单，因此我们选用它为数据库系统。

采用 MySQL Workbench 为数据库建模工具，并且可以用 MySQL Workbench 设计和创建新的数据库图示，建立数据库文档，以及进行复杂的 MySQL 迁移。MySQL 是一种开放源代码的关系型数据库管理系统（RDBMS），MySQL 数据库系统使用最常用的数据库管理语言--结构化查询语言（SQL）进行数据库管理。

数据库环境说明

数据库系统	数据库部署环境	数据库设计工具	数据库存放位置	说明
MYSQL	WIN8	MySQL Workbench	/bin	存放、记录各表信息

### 6.2 数据库命名规则

有固定英文意义的字段按照英文名命名。

没有固定英文意义的字段名按照 A\_B 两段式为规则进行命名。

中间无空格。

数据库表的字段长度不超过 254 。

### 6.3 安全性设计说明

应用级的用户帐号密码不能与数据库相同，防止用户直接操作数据库。用户只能用帐号登陆到应用软件，通过应用软件访问数据库，而没有其它途径操作数据库。

角色与权限

角色	可以访问的表与列	操作权限
上级用户	可以访问该用户所有数据	完全控制权限
	可以访问该用户下级所有数据	只读
下级用户	可以访问该用户所有数据	完全控制权限
管理员	可访问所有数据	完全控制权限

## 6.4 表汇总和表设计

表汇总	用途
Star	歌手信息存储表
Song	歌曲信息存储表
Comment	评论信息表
Client	客户端信息表
C_Song	客户端歌曲播放信息表
Song_List	客户端歌曲信息存储表
Similarity_Form	歌曲相似度信息

### 详细表结构设计

歌星信息表(Star)

字段名	数据类型	约束条件	是否为空	描述
StarID	Varchar(20)	Primary key	Not null	歌星编号
StarName	Varchar(50)		Not null	歌星姓名
StarStyle	Varchar(10)	男/女/组合	Not null	歌星类型
StarRegion	Varchar(50)		Not null	歌星地区
StarNameAbridge	Varchar(10)	字母	Not null	歌星拼音缩写
Route	Varchar(50)		Not null	文件路径

歌曲信息 ( Song )

字段名	数据类型	约束条件	是否为空	描述
SongID	Varchar(20)	Primary key	Not null	歌曲编号
SongName	Varchar(50)		Not null	歌曲名称
SongType	Varchar(20)		Not null	歌曲曲风
SongLanguage	Varchar(10)		Not null	歌曲语种
SongRoute	Varchar(100)		Not null	歌曲文件路径
SongNameAbridge	Varchar(10)	字母	Not null	歌曲拼音缩写
StarID	Varchar(20)	Foreign key	Not null	歌星编号
SongWeek	Long		Not null	周点击率
SongMonth	Long		Not null	月点击率
SongYear	Long		Not null	年点击率

评论信息表 (Comment)

字段名	数据类型	约束条件	是否为空	描述
C_ID	Varchar(20)	Primary key	Not null	评论编号
C_content	Varchar(80)		Not null	评论内容
SongID	Varchar(20)	Foreign key	Not null	歌曲编号

客户端信息表 (Client)

字段名	数据类型	约束条件	是否为空	描述
ClientID	Varchar(20)	Primary key	Not null	客户端编号
ClientArea	Varchar(20)		Not null	客户端产地
ClientType	Varchar(20)		Not null	客户端型号
ClientBrand	Varchar(20)		Not null	客户端品牌

客户端\_歌曲表(C\_Song)

字段名	数据类型	约束条件	是否为空	描述
SongID	Varchar(20)	Primary key/ Foreign key	Not null	歌曲编号
ClientID	Varchar(20)	Primary key/ Foreign key	Not null	客户端编号
C_Order	Int		Not null	歌曲顺序

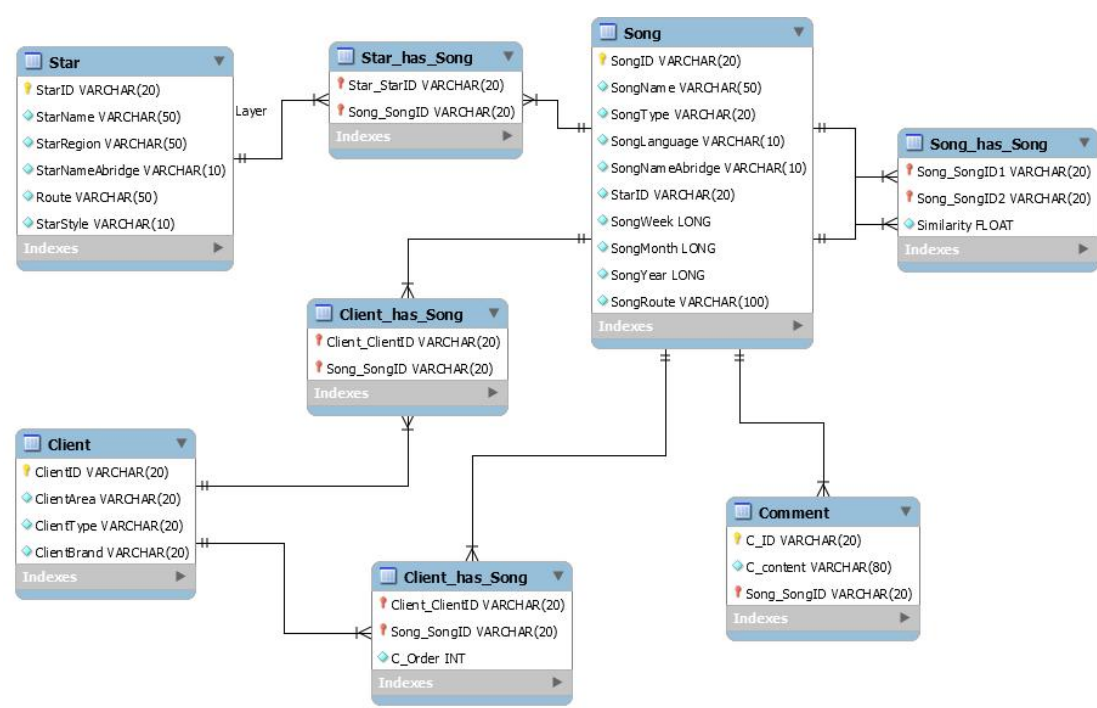
歌曲列表 (Song\_List)

字段名	数据类型	约束条件	是否为空	描述
ClientID	Varchar(20)	Primary key/ Foreign key	Not null	客户端编号
SongID	Varchar(20)	Primary key/ Foreign key	Not null	歌曲编号

相似度表 (Similarity\_Form)

字段名	数据类型	约束条件	是否为空	描述
SongID1	Varchar(20)	Primary key/ Foreign key	Not null	歌曲 1 编号
SongID2	Varchar(20)	Primary key/ Foreign key	Not null	歌曲 2 编号
Similarity	float		Not null	歌曲相似度

数据库 ER 关系图



## 7. 用户界面设计概述

### 7.1 设计要素

#### (1) 文字

字的内容应力求简单扼要，突出重点。过大的字体浪费空间资源。屏幕上的说明文字一般控制在 10~14 号，而标题文字要控制在 18 号以上。另外文字与背景之间要有一定的对比度，对比不明显，会造成文字不易识别。

#### (2) 图片

图形图像应尽量放于屏幕的中心位置。如果图片较大，不要再设置外框或花边。如果一个页面同时展示几张图片，要区别主次、顺序、大小等。作为背景的图像要简洁明了、颜色淡雅、和谐统一，最重要的是能使文字和图片具有较好的易读性。

#### (3) 图标

设计按钮和图标时，可以通过色彩、造型的形式来使按钮具有自己的特色，易发现，好掌握。按钮上的文字和图片要清晰，大小适中。

## 7.2 注意事项

### （1）界面清晰

界面是用户对于这款软件的第一印象，建立清晰界面的思维，从而将这种思维应用到实际开发之中去，这将会在很大程度上提高用户体验。模糊不清，主题不明确的界面是不会引起用户喜欢的。模糊的网站界面，会给用户以混乱的错觉，这样会很容易引起疲劳。因此我们首先以清晰为主大体上确定了界面框架。

### （2）简洁但不简单

但是光靠清晰的界面还远远不能称得上是一件好的产品。一个好的 UI 设计，不仅具备清晰的界面，还要给用户以简洁的感觉。用户的第一印象是非常重要的。而简洁的 UI 设计原则会让用户在最短的时间内找到自己想要的内容，避免用户在繁杂的内容中不断探索，因此扁平化是 UI 设计应该考虑的一种方式。因此我们针对软件功能进行了详细的分析，并且将各种功能在界面上的位置进行了设计。

### （3）给用户友好交互

整体要有一个同意的风格，不仅包含了颜色的搭配要统一，而且字体、布局等也要统一，这完全是站在用户的角度来思考问题的：只有保持统一的风格，才不会让用户在访问页面时产生错愕的感觉。在 UI 设计中加入“质感”，这会站点更加不同，更加完美。美好的事物总是会让人心神向往。而增加美观度，就是让产品成为美好事物的必备条件。

## 7.3 桌面客户端界面

个性推荐界面





歌手界面

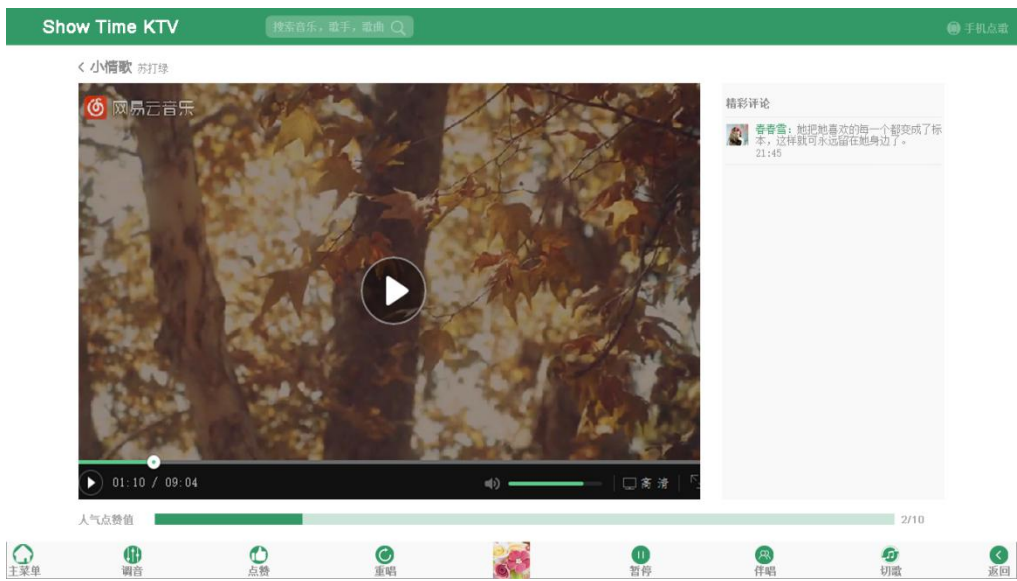


排行榜界面

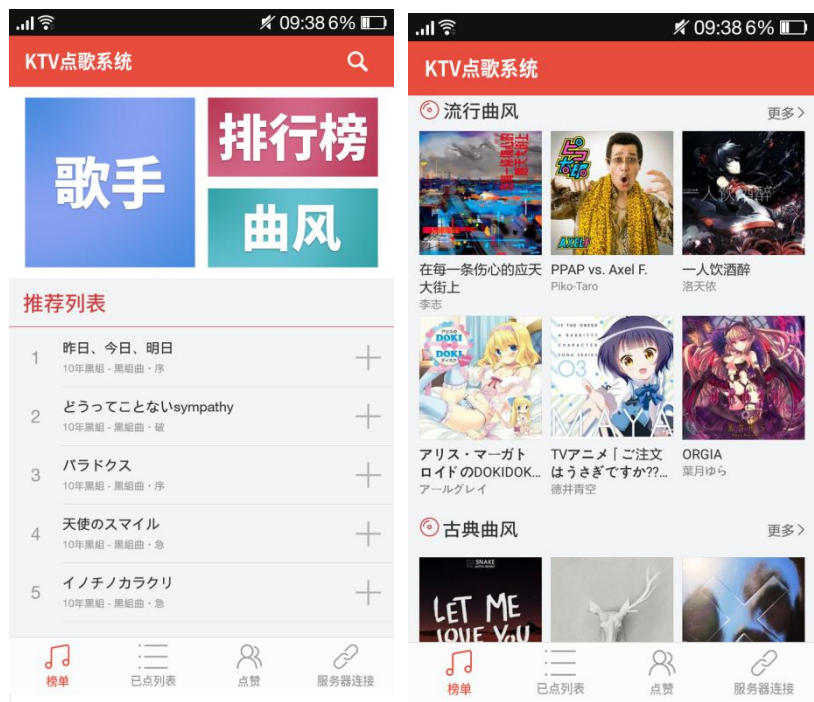


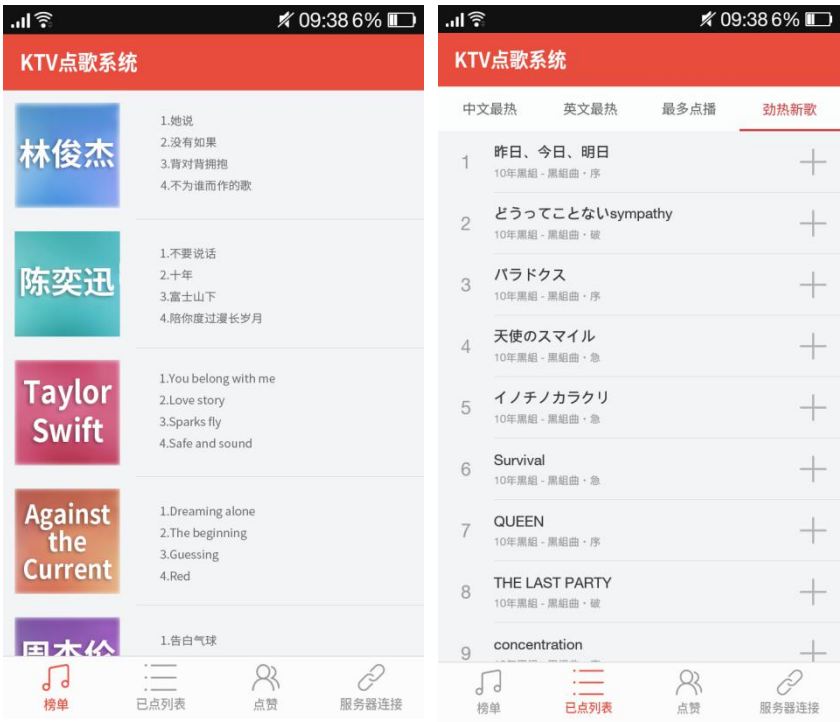


播放 MV 界面



7.3 移动客户端界面





## 8. 综合考虑

### 8.1 稳定性和可扩展性

#### 8.1.1 不变需求

STV 的基本点歌需求长期不会发生变化。

#### 8.1.2 变化的需求

一般而言，为了迎合用户的喜好，以及当下流行趋势，STV 的特色功能可能会不断发生变化。系统具备插件功能便于项目的二次自定义开发。

### 8.2 性能分析

#### 8.2.1 用户需求的性能指标

- 点歌操作响应不得多于 1 秒。
- MV 文件下载过程不能出现明显的延迟。
- 演唱过程中不得出现卡顿现象。

## 8.2.2 实现性能指标

通过缓存机制后台离线缓存媒体文件。  
本地数据库缓存近期后台服务器数据。

## 8.3 复用和移植

### 8.3.1 复用

系统媒体解码采用开源 **Gstreamer 1.0** 媒体框架。图形显示采用 **Gtk+-3.0** 开源图形库。  
系统将媒体播放进一步封装成播放机。作为一个独立子类。可供其他开发者使用。  
系统后台服务可供其他开发者开发不同的客户端。  
用户界面代码采用 **XML** 标记语言编写，而功能代码由 **Python** 实现。

### 8.3.2 移植

系统客户端以及后台服务端均采用 **Python3** 开发。  
系统使用的三方框架均为跨平台主流框架。  
系统代码内部包含平台判断代码，开发过程中已经考虑跨平台性。  
原生 **GUI** 框架，利用移植到嵌入式设备。

## 8.4 防错与出错处理

严格按照用户使用手册使用可防止出错。  
如果出现错误，重启应用程序即可。

## 8.5 其它

无。