## System Context

Based on the Requirement specification [SEGVN-Team5-RS 10], the implemented Jukebox system will provide a number of functions that help manage and play music efficiently

As Administrator role, He/she can do the following functions:

* Configure Jukebox system in Stand-alone or Distributed system
* Assign Volume control for Users
* Adjust Volume for all Users
* Manage music store (Add/Delete music song & Add genre)
* List statistic (the list of music song with the number of playing times)

As User role, He/she can do the following functions:

* Deposit money from credit card for playing music (check it by available banking system)
* Select/Play music
* Adjust Volume

Here is the list of quality attributes (2.1 of [SEGVN-Team5-RS 10])

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Use case** | **Attribute** |
| Performance | QP01 | Response time | Manage music store UCA03 | Table-side Jukeboxes automatically update changing information within 15s |
| QP02 | Delay time | UCU03 - Play music | Delay time between pressing “Play” button of user and the running of the first song is less than 15s |
| Modifiable | QM01 | Easy to configure | UCA03 - Manage music store, UCA06 - Configure clients volume control | Manage music store and configure clients volume control without restarting the system |
| QM02 | Support many music formats | UCU03 - Play music | Support three music’s formats (mp3, wma, wav) |
| Usability | QU01 | Easy to select music | UCU03 - Play music | Easy to search and select music songs filtered by song author, title, or genre and can play the excerpt of a music song |
| QU02 | Easy to setup system configuration | Deployment phase | Easy to setup system as standalone or distributed version |
| QU03 | A pleasing esthetic appearance | UCU03 - Play music | Show pictures |
| Availability | QA01 | Availability of Table side Jukeboxes | During runtime execution | The Jukebox system is always available for Table side Jukeboxes.  When detecting errors, system should send email to administrator |
| Security | QS01 | User’s credit card information | UCU01 - Deposit | Not compromise user information (credit card info) |

## Purpose

This document is intended for:

1. Project Manager and Architect to manage and evaluate if Jukebox system meets function requirements and quality requirements or not,
2. Developers to make coding
3. QA members and Testers to make Testing plan and test cases
4. One that have an interest in documentation of the architectures and design of software systems.

## Overview

Jukebox system will be described as a number of different views:

* (2.) Combined View (Deployment View & Client-Server View): Provide the whole picture of Jukebox system’s Architecture
* (3.) Dynamic view (Component and Connector View) : provide the picture of runtime entities and potential interactions for each function (implement some use cases described in Requirement specification)
  + (3.1) Shared – data view Packet 1: Manage music store
  + (3.2) Shared – data view Packet 2: Select/Play/Adjust volume
  + (3.3) Shared – data view Packet 3: Configure/Adjust clients volume
  + (3.4) Filter and Pipe view Packet 4: List statistic
  + (3.5) Share – data view Packet 5: Availability of Table side Jukeboxes
* (4.) Statistic View (Module View):
  + (4.1) Decomposition View Packet 1: Jukebox system (4.1) to describe the modular structures of a Jukebox system’s software
* (5.) Physic View (Allocation View):
  + (5.1) Deployment view Packet 1: Jukebox system (5.1) to describe mapping of the software architecture onto its environments
  + (5.2) Implementation view packet 2: jukebox system (5.2) to describe where modules are allocated
  + (5.3) Work assignment view packet 3: Jukebox system (5.3) to describe which module will be implemented and tested by whom

Each view is presented as a number of related view packets. View packet is the small cohesive bundle of information about system. View template will be described in Appendix A

* (6.) Detail Design, including:
  + (6.1) Central Jukebox Database
  + (6.2) Graphic User Interface
  + (6.3) Class diagram
  + (6.4) Sequence diagrams

Glossary and References will be listed in the end of document

# Combined View

## Combined View (Deployment and Client- Server) Packet 1: Jukebox System

### Primary Presentation

Table-side Jukeboxes (Clients: UI and Control)



Central Jukebox (Database Server)

KEY

Network connection

Tier

A B: A Request B

A B: B Response A





### Element catalogue

#### Elements and their properties

|  |  |
| --- | --- |
| **Element** | **Responsibility** |
| Central Jukebox | Database Server MySQL and File Server (music files) |
| Table side Jukeboxes | Provide UI to provide the functionality of the system:  For Administrator: Login, Logout, Manage music store, Adjust clients’ volume, Configure clients volume control, List statistic  For User: Play music, Select music, Deposit money, and Adjust volume |

#### Relationships and their properties

|  |  |
| --- | --- |
| **Element** | **Responsibility** |
| Request/response (client/server) | Via Network connection (wire or wireless) , Table side clients connect to Central Jukebox to access Music Database |



Select music

Adjust Volume

Play music

Deposit

Configure Clients Volume Control

Adjust Clients Volume

Login

Logout

Mange music store

**Jukebox system**

**Users**

**(1..n)**

**Administrator**

List statistics

**Banking**

**System**

Credit Card



### Context diagram



### Architecture Background

#### Design rationale

Jukebox System uses the Laptops and requires the distributed configuration, so Client – Server is rational. Client – Server model has some following advantages compared with Peer To Peer:

* + Modifiability:
    - It is possible to replace, repair, upgrade, or even relocate a server while its clients remain both unaware and unaffected by that change.
    - Since data storage is centralized, updates to that data are far easier to administer in comparison to a P2P paradigm. In the latter, data updates may need to be distributed and applied to each peer in the network, which is both time-consuming and error-prone, as there can be thousands or even millions of peers.
    - It functions with multiple different clients of different capabilities.
  + Security: All data is stored on the servers, which generally have far greater security controls than most clients. Server can better control access and resources, to guarantee that only those clients with the appropriate permissions may access and change data.

#### Assumptions

* The function related to Credit card is available by Banking System

### Related view packet

* Parent: None
* Children: None
* Siblings: None of this view. Other view that express the same scope are:

Decomposition View Packet 1: Jukebox system (4.1)

Uses View Packet 2: Jukebox system (4.2)

Deployment View Packet 1: Jukebox system (5.1)

Implementation View Packet 2: Jukebox system (5.2)

Work Assignment View Packet 3: Jukebox system (5.3)



# Dynamic View (C&C View)

## Shared – Data Packet 1: Manage music store (use cases UCA03 and UCA04)

### Primary Presentation

Music Dict.

Music Dict.

**KEY** Call – Return

Request – Reply (SQL-JDBC)

Request – Reply (Network socket)

Software Component

Database File Screen UI

Add new genre

Admin music store control (DB)

Central Database

(Music Dictionary + Configuration)

Auto update

(each 10 s)

Add genre/Add or Delete song

Music Dictionary

Configuration

Add song UI

Add genre UI

Delete song UI

Brow Music files

Write Music files stination

Select/Play Music UI

Add new song

Delete song

Music source

Music destination

Write music Dict.

Music Store Control (DB)

Read music Dict.

Write Configuration

Configure VolumeControl (DB)

Read Config.



### Element catalog

#### Elements and their properties

|  |  |  |
| --- | --- | --- |
| **Element** | **Type** | **Description** |
| Add song UI | Screen UI | Allow the Administrator specify where the music source is and select a song to add and specify its attributes (author, title, genre) |
| Delete song UI | Screen UI | Allow the administrator select a song to delete |
| Add genre UI | Screen UI | Allow the Administrator to specify a new genre to add |
| Admin music store control (DB) | Software Component | This component can:  Brow music files in Central Jukebox  Write selected music files to the given destination in Central Jukebox  Read music dict. from Database  Write music dict. to Database |
| Central Database | Database | Where available music dictionary and Configuration are stored |
| Music source | Driver/Directory where music source files are stored | Where “Music store control” will read music files (mp3, wav, wma ) in Central Jukebox |
| Music Destination | Driver/Directory where available music files are stored | Where “Music store control” will store available music files for the user to play |
| Auto update | Software Component | This component each 10s:   * + - 1. Forces “Configure Volume Control”, and “Music Store Control” to load Music Dictionary and Configuration from Database (in Central Jukebox) into Music Dictionary and Configuration (in Table side Jukebox)       2. Force “select/play music songs UI” to update data in “Music Dictionary” and “Configuration” |
| Music Dictionary | Object | Where Music dictionary is stored in Table side Jukebox |
| Configuration | Object | Where Configuration is stored and in Table side Jukebox |
| Select/Play music UI  (List 10 the newest music songs) | Screen UI | Display the list of 10 newest music songs |
| Music Store Control(DB) | Software Component | This component to read music Dict. from Database (central Jukebox) into music Dictionary (Table side Jukebox) |
| Configure Volume Control (DB) | Software Component | This component to read Configuration from Database (central Jukebox) into Configuration (Table side Jukebox) |
| Server Listener | Network socket connector (Brow Music files and Write Music Files) | To listen the requests from Table side Jukebox |
| File Brower | Network socket connector (Brow Music File and Write Music Files) | It can:   * To brow drivers/directories/music files in Central Jukebox * To copy music file chosen by the user to the location specified in Configuration table in Central Jukebox |

#### Relationships and their properties

The relation of this view is attachment, dictating how components and connectors are attached to each other. The relations are shown in the primary presentation; there are no additional ones

#### Element behavior:

1. Add genre

Admin music store control

Add genre

Read genres

Music Dictionary

Add new genre

Read genres

1. Add song

Add song

Admin music store control

Music Dictionary

Music source

Music Destination

Read music files

Write music files

Write song Attributes (file name, tile,

Author, genre, singer)

Read genres

2.1) Select Music File

List Music File From folder

Add Song

Music store control

Request MusicList (path)

Server Listener

Open Connection

File Browser

create

2.2)Copy Music File

Write music file

Add Song

Music store control

Request copy file

Server Listener

Close Connection

File Browser

1. Delete song

Delete

song

Admin music store control

Music Dictionary

Read Music songs

Delete Music songs

### Context diagram

View Packet described in this section

Select music

Adjust Volume

Play music

Deposit

Configure Clients Volume Control

Adjust Clients Volume

Login

Logout

Mange music store

**Jukebox system**

**Users**

**(1..n)**

**Administrator**

List statistics

**Banking**

**System**

Credit Card

### Architecture Background

#### Design rationale

Design decisions:

* All music files are stored in Central Jukebox’s disks
* Central Music dictionary is stored in Central Jukebox’s Database (MySQL)
* Each 10s central music dictionary and Volume Configuration will be updated in Table side Jukebox from Central Jukebox

The rationale for this decomposition is to further divide into 3 software components: GUI, Control (to connect with Server), and Auto update:

* The Producer (Create/update music store in Server) is separated from the Consumer (read music store from Server into client) that will promote concurrent communication.
* GUI (for presenting data) is separated from Control (for connecting Database and music files in Server) that will promote modifiability

Why our design meets quality attributes will be explained in the following table

|  |  |  |  |
| --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Response measure** |
| Performance | QP01 | Response time | table-side Jukeboxes automatically update within 15s |
| **Rationale**: Auto Update is the timer that each 10s will read music store in Server and write to Clients | | | |