1. **System overview.**

## System Context

Based on Team assignment document - provided information about POS-System, the implemented Smart Mart POS system will provide a numbers of function that help head manager control, monitor and report status of store system efficiently.

The following table will describes roles and responsibility of system.

|  |  |
| --- | --- |
| Role | Responsibility |
| Head administrator (manager) | 1. Manage head office, build policies about system items and prices. 2. Synthesis and report business information. 3. Manage store administrators and branches. |
| Store administrator | 1. Manage store office, build policies about local items and prices 2. Collect sales data to analyze and report to head office. 3. Manage cashier accounts. 4. Manage customer information. |
| Cashier | 1. Sales. 2. Manage customer information. |

List of quality attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Attribute** |
| Performance | QA.01 | Update customer’s score immediately | When the score are used, the number of score used is immediately subtracted from the number of score accrued by the member. |
| The synchronization between the branches and center. | Moreover, in addition to the sales operation, the system is also capable of performing the statistical analysis on the sales records of all stores in near real-time manner. |
| Availability | QA.02 | No down time | No down time when connection with head server have been cut. |
| No down time | No down time when connection of POS terminal with store server have been cut or store server has been broken. Switch to use POS local database to ensure transactions will be continued. |

## Purpose

This document is intended for:

1. Project Manager and Architect to manage and evaluate if Smart Mart POS system meets function requirements and quality requirements or not.

## Overview

This document will provide a big picture of Smart Mart POS system (SMPs) structures in architect. It will be present by using three different views.

1. Allocation view, describe a big picture of SMPs architect and physical device location.
2. C&C view, describe how to SMPs operate and data flow through each function.
3. Module view, describe structure of system.

Glossary and References will be listed in the end of document

# Allocation view

## Allocation View (Deployment views)

### Primary Presentation



### Element catalogue

#### Elements and their properties

|  |  |
| --- | --- |
| **Element** | **Responsibility** |
| Application server (head office server) | Provide functions help manager collect analyzed information from the stores to make business report. Help manage store administrator accounts. |
| Application server (store server) | Provide functions help administrators collect and analyze data from POS terminal, synthetic and report information sent to head office. |
| POS terminal | Provide functions help cashiers make transaction. |
| Database server SQL server 2003 | Database management. Store primary data in main activities. |
| Database SQLite | Store data when system was broken. |

#### Relationships and their properties

|  |  |
| --- | --- |
| **Element** | **Responsibility** |
| internet (between head and store server) | Internet connection between head and store server. Protocol HTTPs |
| Local network | Connection between POS terminals with their store server. Connect between database server with head application server. Protocol TCP/IT |



### Context diagram





# C&C view.

1. **Module view.**

# 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 | | | |