



RESEARCH ARTICLE

DESIGNING MANAGEMENT CLIENT FOR THE VMS APPLICATION

Parth Patel.

M.Tech Student, Embedded and VLSI Department, U. V. Patel College Of Engineering, Ganpat University, Kherva, Gujarat.

Manuscript Info

Manuscript History

Received: 26 February 2017
Final Accepted: 18 March 2017
Published: April 2017

Key words:-

VMS, MVC, Qt framework, Management client, Desktop client, Desktop application.

Abstract

The VMS(Video Management System) Management Client is a front end side Desktop based workstation used for administration of VMS Management Services. The Management Client is used for the login Server and provide various configurations such as Alarm Management, event Management, Scheduling configuration, PTZ configuration, Device management. Management Client communicates with the Management Server and Store all the data in the Management Server Database. VMS Management Client is also communicated the Desktop Client and Core services. Client, Management client and Core Services are one chain to the Data communication. They all are depending on each other. Management Client is a center of the Core Services and Desktop client. Management Client is useful in the Data communication between GUI to Core Services and Core to GUI (Graphics User Interface).

Copy Right, IJAR, 2017,. All rights reserved.

Introduction:-

The VMS is generally video Management Software or Video Management Server to perform an important role in VMS. VMS collects video data from the camera and other sources. They provide Live Video Stream from Camera. VMS recorded and store the video into the storage device RAID (redundant array of independent disks). It is also provided Live view and Recoding of the Video.

The Video Management System divided into the Desktop Client, Management Client (Server) and Core Services.

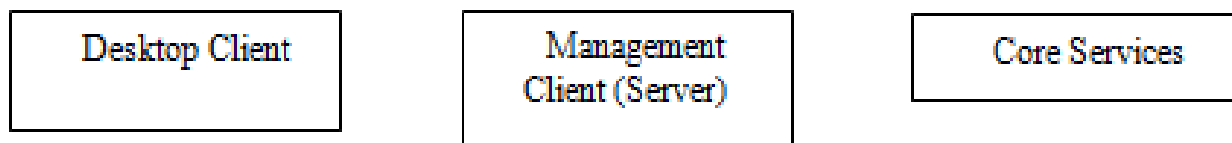


Fig. 1:- Basic Diagram.

The Block Diagram of the project Desktop client, Management client (Server) and Core Services communicate each other. The Desktop client (Graphics User Interface) is Communicate System User. The Management client and Core Services Backhand side of the Desktop client.

Corresponding Author:- Parth Patel.

Address:- M.Tech Student, Embedded and VLSI Department, U. V. Patel College Of Engineering, Ganpat University, Kherva, Gujarat.

The Desktop client used to live monitoring, alarm Handling, online event viewing, playback and map operation performed. Desktop login into the VMS system through the Management Server, Receive Live Video and Playback Video from Streaming Server and recording Server. Management client is a front-end of the desktop client. It is performed as a work station. Management client provides the Services to the User requirements. The User or client login into the Application then Management server provides various configuration setting services as an event management, scheduling configuration, PTZ configuration and device Management. The Management Client will communicate with Management server and store all the data in Management server Database PostgreSQL and other database also available. The Core Services provides services to the Management client requirement. The management client communicates the Core Services. Recorder records the video and uses this video playback side. It can be used to build a system that reads files in one format, process them and export them into another.

Literature Survey:-

Video Management System:-

The Management server is required for the multimedia streaming service can be limited by the various factors such as Internet bandwidth and so on. The Management server front end side of the Desktop Client (GUI).

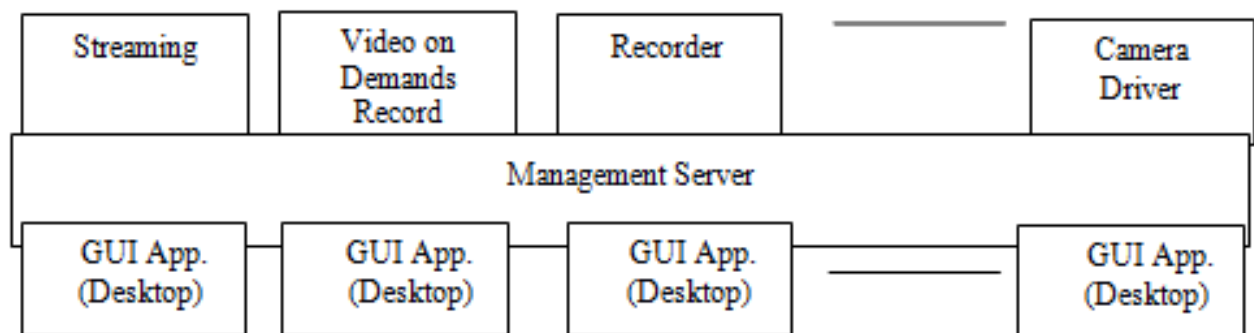


Fig. 2:- Video Management System

The Fig. 2 VMS System multiple User required different Services By the Server. The Multiple Services are available on Core Side. The Management Server Communicate Core Services and give Proper output to the user request. Hear Multiple user Communicate GUI Side. For the user requirement live video Streaming then Management Server Send The Request Live Streaming Driver. The Driver Start The Camera live Video Streaming. Other imported Services like as a User required Video Recording then user Send Request Management Server. Management Server Call the Core Services Recorder driver. The Recorder Start And the live video recording will be Starts. The Other Core Services Video On demands (VOD) to The User Requirement recorded Video Will play. Other Camera Driver and Multiple Services are Available into the Video Management System. The Multiple User easily connected to The Management Server. The Management Server Manage the user Request and fulfillment it.

MVC Architecture use in Management Server:-

The MVC Architecture is a Model-View-Controller Architectural. The [1] Model-View controller (MVC) is a software architecture concept considered as an architectural pattern in software engineering. The Data Store in the database and provide original data into the system.

The Model and View program Write in Qt open source Software. [2][3] The Qt is Open Source C++ programming Software. The Qt build All modules have a common scheme and build from the same API Design idea. Cross platform application builds from one source. QOBJECT class is a base class from the Qt. QOBJECT is a base class of the all all widget and Qt Classes. It contains many of the application mechanism that makes up Qt. Event, Signal and Slots, properties. Qt is meta data generated at compile time by the meta object compiler. Qt is a Meta object Compiler, it's generated MOC file.

The Video Management System Application all View .cpp programs and all Model View class programs written in Qt. The Controller.Zip files write in pure C++ programming. The controller does not use any Qt class. They use Standard C++ Libraries. The Controller is independent. They have not dependent Qt inbuilt class.

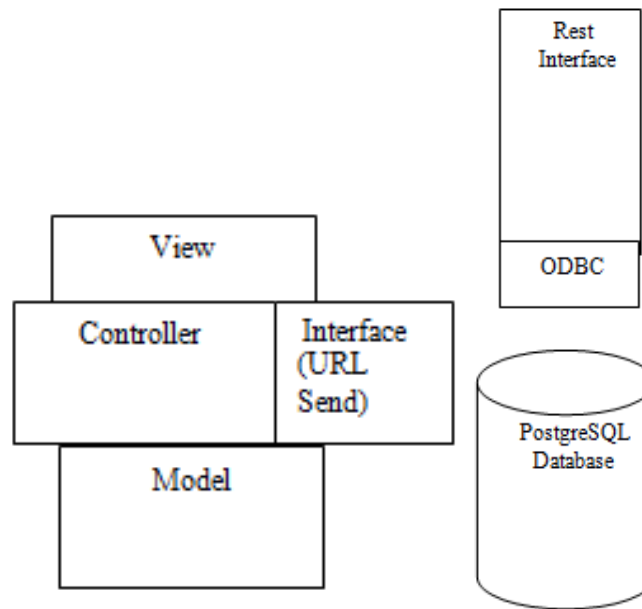


Fig. 3:- MVC Architecture use in Management Server.

The MVC Architectural in Fig. 3 The Model set in the View. The View Communicate Controller. Controller accepted View Requirement and making some changes in the model. The updated Model set in the View. The User Requirements all Changes doing into the Controller and Management Server. The User generally login into the system they enter the user name and password that time View send the data into the controllers. The Controller converts user and password data into the array Structure and send the Structure into the Back hand controller Interface. The Interface Convert the Structure into the JSON format. JSON is a Java Script Object Notation. It is a lightweight data-interchange format. It is easy to read write. It is based on Java script programming language. The Backhand controller interface sends JSON URL to the

Rest Interface. Rest Interface is connected to the ODBC connector. ODBC is an Open Database Connectivity. The ODBC is open Standard application programming interface (API) to access database. The ODBC is connected to the differences database. The PostgreSQL use currently. In future cases, we will change mySQL that time only, nor any change for view side. We use different database in VMS system using ODBC. The JSON String passes into PostgreSQL Database. The user name and Password check into the PostgreSQL database login table and send the rest Interface. The Rest interface checks the input user name and password. Rest interface checks the user password and user name are correct or not. The send True, false to the backhand controller Interface to The JSON format. The Interface Convert the JSON string into the Structure and send the Controller. The Controller sets the output in the model. Model updated to the controller data and Model Set into the View. If User Password true, then view give permission login else show pop-up messages user password incorrect.

Results:-

This paper, We will discuss the results of the framework. They develop a simple application name “Video Management System” test Qt C++ framework. We displayed the Some snapshots of over application. The different type operation such as a data insert in database, data editing and updating and Searching a data.

Auto Add Device

Use this Option if you know the IP address of the device you want to add. you can either type the IP address manually or import them from Hardware Configuration file

Manual Add Device

VMS will scan your network for available devices you can extend scan to include other network. when scan include other network. when scan is complete. select which hardware to add.

☐ Use This Option to set device naming conversion

Previous **Next** **Cancel**

Fig. 4:- Select Camera Add Mode.

Type IP addresses

Type Ip Address of the hardware you want to add your System or import the information from configuration file. You can Speed up the scanning by Selecting the manufactures of the manufacture(s) of the device you want to add.

Ip Address	UserName	Password	Port	Manufacture	Model
------------	----------	----------	------	-------------	-------

+

port CSV File

Previous **Next** **Cancel**

Fig. 5:- Manual Camera Add set.

The fig 4 and fig 5 show the how to select camera add Selection Mode. Two ways to add a camera in over system. Auto Detect and Manually. Fig 4 gives the permission which way or add a camera. We select the Manually add

Camera in over system. Fig 5 manually add camera page. The some information like as a Camera IP Address, Model name, Port, User Name and Password fill by User.

Type IP addresses
Type Ip Address of the hardware you want to add your System or import the information from configuration file.
You can Speed up the scanning by Selecting the manufactures of the manufacture(s) of the device you want to add.

Ip Address	UserName	Password	Port	Manufacture	Model
10.103.3.6	root	pass	554	Axis	AXIS_P5534
10.103.3.7	root	paa	553	Sony	SONY_001
10.103.3.6	root	pass	554	Axis	AXIS_P5534

Fig. 6:- Add Camera Manually.

You have Successfully added 2 Camera(s) to your System Out of 3
Your Camera(s) are ready to use

IP	Camera Name	Status	Reason
10.103.3.6	Cam_10.103.3.6_Axis	Added	Success..
10.103.3.7	Cam_10.103.3.7_Sony	Not Added	Fail..
10.103.3.6	Cam_10.103.3.6_Axis	Added	Success..

Fig. 7:- Successfully added Camera.

The VMS System Fig 6 and Fig 7 show the manual camera added and how many cameras successfully added in Server. The Fig 6 User add Three cameras Successfully and go the next button. The GUI sends the request Management server then the Management server, check camera IP, user, model and Password true or false using core Services. If all information is true for adding camera request on the server, Fig 7 shows that it will be added to the server.

Conclusions:-

In this paper, We have presented a Qt Framework to develop the desktop application software using MVC Architecture. In this Software development MVC Model is separate using model, view and controller. The Management Server front-end side of the GUI. The Management Server Manage the database and Core Services. The Send Requirement information to the GUI Side. This Application MVC model is a separation of business logic in the controller. This Application Testing then MVC Architecture Perform effectively and Management Server Send and receive URL Request properly. The Management Server Communicate the Core Services and give proper output to the Desktop Client. The actual operation has proved this framework is stable, efficient and able to develop high quality applications.

References:-

1. Iqbal H. Sarker and K. Apu, "MVC Architecture Driven Design and Implementation of Java Framework for Developing Desktop Application" International Journal of Hybrid Information Technology Vol.7, No.5 (2014).
2. C++ GUI Programming with Qt 4, Second Edition., Prentice Hall: Jasmin Blanchette ; Mark Summerfield, 2008, pp. Preface.
3. The Book of Qt 4 :The Art of Building Qt Applications, by Daniel Molkentin, July 2007, 440 pp.ISBN-13: 978-1-59327-147-3.