1. Class diagram

2. Class description

2.1 – CameraDetail

|  |  |  |
| --- | --- | --- |
| CameraDetail | | Model – contains camera information (video url, camera id, group id ...) |
| Data type | Method | description |
|  | CameraDetail() | construction |
| string | **getUrlSD() *const*** | return ws stream url SD  ex:  CameraDetail cam;  ...  string url = cam.getUrlSD();  \*sd-resolution: 720x480 |
| void | **setUrlSD(*const* string &value)** | set ws-stream url SD  ex:  CameraDetail cam;  cam.setUrlSD("ws://url"); |
| string | **getUrlHD() *const*** | return ws stream url HD  \*hd-resolution: 1280x720 |
| void | **setUrlHD(*const* string &value);** | set ws-stream url HD |
| string | **getCamID() *const*** | return camera unique ID    ex: camera\_id: "6581" |
| void | **setCamID(*const* string &value)** | set camera ID  ex:  cam.setCamID("6851") |
| string | **getGroupAddress() *const*** | return group camera address in nature text  ex:  cout<<cam.getGroupAddress()<<endl;  result: "52 Nguyen Chi Thanh" |
| void | **setGroupAddress(*const* string &value)** | set group camera address  ex:  cam.setGroupAddress("52 Nguyen Chi Thanh") |
| string | **toJsonString()** | convert infomation (camID, url, groupID, group adress..) to json type then return as string |
| Json::Value | **toJson()** | convert infomation (camID, url, groupID, group adress..) to json type then return as Json::Value |
| int | **getGroupID() *const*** | return unique groupID |
| void | **setGroupID(int value)** | set groupId value |

2.2 - LoginCmsVp9

|  |  |  |
| --- | --- | --- |
| LoginCmsVp9 | | Controller - control login to VP9-CCTV and request video url |
| Data type | Method | description |
| *explicit* | LoginCmsVp9(QObject \*parent = Q\_NULLPTR) | constructor |
|  | *LoginCmsVp9(string pathFile)* | constructor with path to file config in local disk  \* config file include information need to login and declare to front-end web site. Detail info looking in SR document |
| void | **ReadFromConfigFile**(string pathFile) | read info in config file |
| void | ***ProcessLine(QString strInput)*** | process specific info within config file to get value in config file  \* process current using Qt framework. |
| void | **Login2Cms()** | login to VP9-CCTV to get token and camID |
| void | ***RequestVideoUrl(int siteID, string token)*** | request new ws-stream url using token get from login  \*receive json data return |
| string | **getPath2File() *const*** | get path to config file |
| void | ***setPath2File(const string &value)*** | set path to config file |
| void | **setAccount(*const* string &value)** | set user-name use to login |
| void | **setPassword(*const* string &value)** | set password for user-name to login |
| string | **getToken() *const*** | return token  \*default: get from login |
| void | ***setToken(const string &value)*** | set token use to request video url  \*using for test without login |
| int | **getSiteId() *const*** | return site ID get from login |
| void | **setSiteId(*const* int &value)** | set site ID  \*for test without login |
| QList<CameraDetail> | ***getListCamera() const*** | get list CameraDetail model |
| void | **setListCamera(*const* QList<CameraDetail> &value)** | reutrn list CameraDetail model  \*default: from request video |

2.3 - VPConnect2Server

|  |  |  |
| --- | --- | --- |
| VPConnect2Server | | Controller - control calling engine, get local machine info, send processed data to server |
| Data type | Method | description |
|  | VPConnect2Server() | constructor |
| std::function<void(std::string mFaceID, std::string data)> | *SendDataDelegate* | delegate argument use in face recognize engine. Send processed data and local machine info to server |
| void | **SetUrl**(std::string address) | set websocket connection which use to send data and local machine info. |
| void | **SendData**(std::string mFaceID, std::string data) | send processed data and local info to server throught websocket.  \*modify this method mean modify delegate |
| std::string | **WrapData**() | wrap processed data and local info to json |
| void | **Connect2Server**() | initial connection from client to server. |
| bool | **CvtMat2Base64**(cv::Mat image) | convert opencv::Mat to base64 data type. |
| std::string | **GetCpuUsage**() | return % cpu usage |
| std::string | **GetRamUsage**() | return % ram usage |
| std::string | **GetDiskUsage**() | return % disk usage |
|  |  |  |