

Multimedia Services : Architecture Design

Group BeNine
TI2316 Context Project

Bryan van Wijk (bryanvanwijk, 4363329)
Dorian de Koning (tcmdekoning, 4348737)
Ege de Bruin (kedebruin, 4400240)
Jochem Lugtenburg (jlugtenburg, 4370805)
Naomi de Ridder (nderidder, 4383109)

April 28, 2016

Supervisor: Dr. Cynthia Liem
Software Aspect TA: Valentine Mairet
Context Aspect TA: Alessio Bazzica

Delft University of Technology
Faculty of EEMCS

Contents

1	Introduction	3
1.1	Design goals	3
2	Software architecture views	4
2.1	Subsystem decomposition	4
2.1.1	Java back-end	5
2.1.2	NodeJS server	5
2.1.3	Clients view	5
2.2	Hardware/software mapping	5
2.3	Persistent data management	6

1 Introduction

In this document the architectural design of the system which will be developed during the context project Multimedia Services is described. The initial version of this document is created in the first weeks, but it will be extended where necessary during the project.

1.1 Design goals

In the product vision goals are described about the features in the system from the users point of view. Here the goals for the system design which are not direct visible for the users are described. These are as necessary for the succes of the project as the goals in the product vision.

Performance

Performance is very important in the system, camera operators must be able to load fast a preset when the director asks for it. Otherwise the system will be useless because they could do it faster when they do it manually.

Reliability

The presets stored are expected to be the right presets as they are loaded again and they are not changed because of an error. The system must also be reliable as in it will be very unlikely that it will crash during a recording.

Portability

The system must be possible to use on different hardware devices. The change of camera type must for example be possible without throwing away the whole system, but with simply change a part of the system.

Maintainability

It must be easy to change parts of the system, for example when the users want an additional feature. The possibility of defect must be as small as possible but when they are detected they must be easy to find and fix.

Usability

Working with the system take place under time pressure during the recordings, so everything has to be work smoothly to make this process less stressful.

Robustness

The system must be capable of handling invalid inputs.

2 Software architecture views

2.1 Subsystem decomposition

The system will be divided into several subsystems to make the system easier to maintain. One of the parts of the system is the client side part, this is basically a HTML web interface. This web interface will be created by the Node JS server and send on an HTTP request to the client. The Node JS server gets the information about the camera's from the java back end. Commands received in the node JS server from clients will be send to the java back end. The Java back-end is the central system and will do all the computations and use the camera module to send IP commands to the camera's. Presets and other information that needs to be stored will be stored in a database. Information from this database can be retrieved using SQL queries which will be send to the database by the java back-end. In the next subsections the subsystems will be described further.

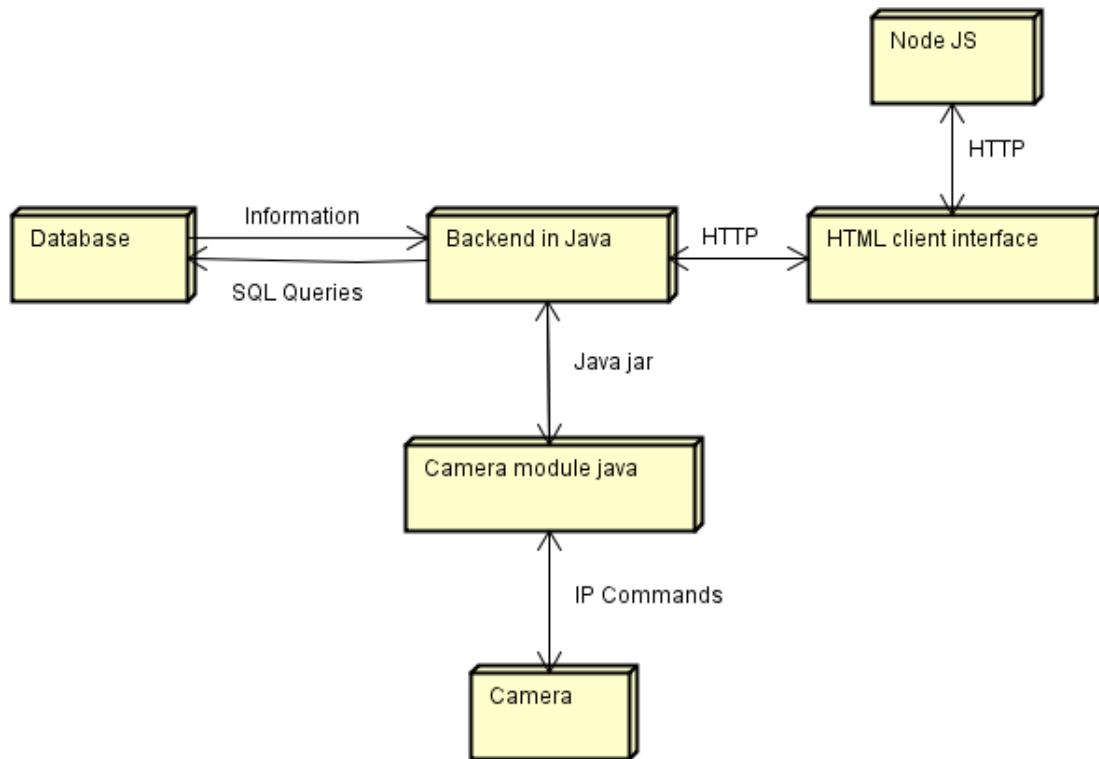


Figure 1: Overview of the system

2.1.1 Java back-end

The Java back-end is where all the main computations are done. The intelligent presets are calculated and corresponding commands are send to the camera's. These commands are send using the java camera module which implements the communication with the camera's. The client gets information about the camera's from the backend using HTTP requests. HTTP endpoints for controlling the camera's are also provided in this subsystem.

2.1.2 NodeJS server

This subsystem provides the resources needed for the clients view. It provides the files for the web page and also provides scripts and information needed for the client to interact with the backend.

2.1.3 Clients view

The users of the system will only see this view which they can retrieve from the NodeJs server. This is for example where the camera operators can select the presets. It is necessary for the success of the product that this part of the system is really easy to use.

2.2 Hardware/software mapping

The software can be used on every device that has a web browser. This makes it possible for the users to use any device they like for operating with the system. The camera's will be able to receive IP commands from the java back-end. If in the future the camera's will be replaced the only part of the system that also needs to be replaced is the camera module. This module takes care of the communication between the java back-end and the camera's.

2.3 Persistent data management

The data will be stored in a database from which the back-end can retrieve the data using SQL queries.

Database Design:

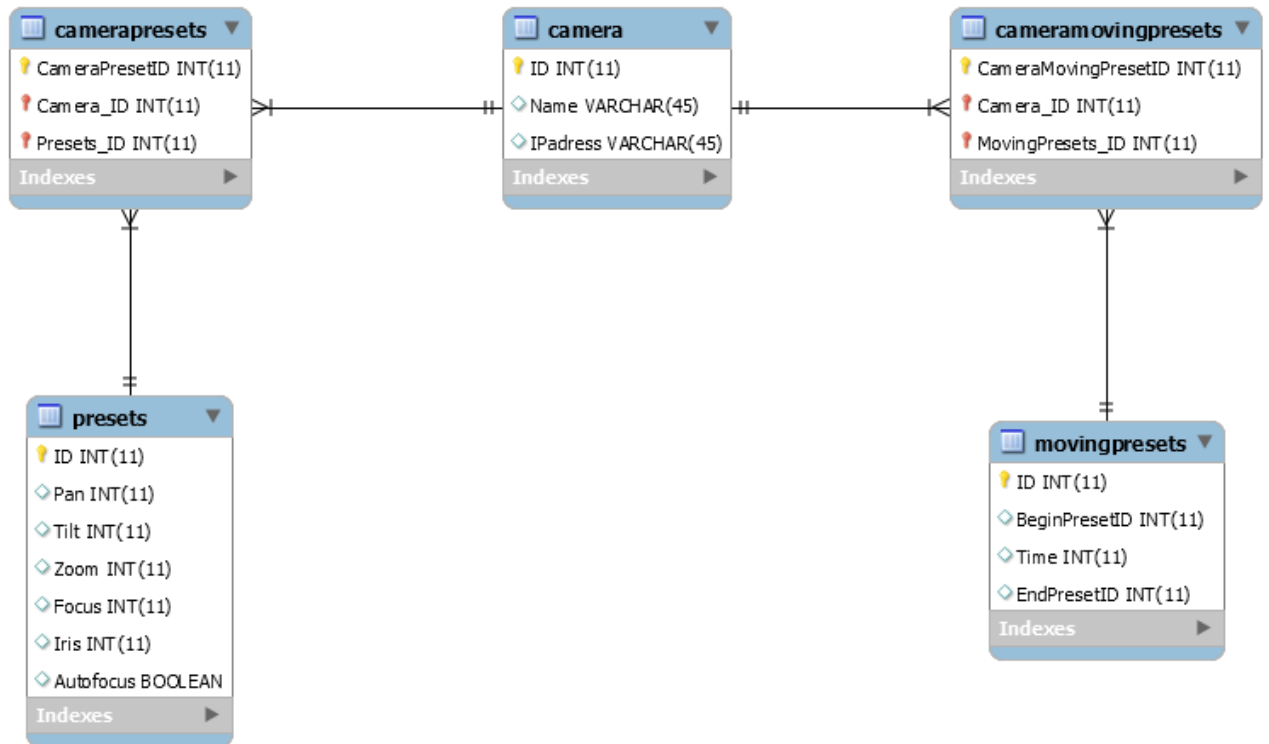


Figure 2: Database design

Glossary

HTTP is a protocol for exchanging information over a network.. 5

HTTP requests are messages send to a server to request some data.. 5

IP commands commands send using the Internet protocol which is widely used to communicate between all kinds of systems. 5

Java a widely used object oriented programming language. 4

Node JS an open source run-time environment for developing server-side web applications.
4