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)

May 6, 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

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 directly visible to the users are described. These are as necessary for the success 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 if 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 should be usable with different operating systems. The web interface should be runnable on different tablet brands. Also the system should be extend-able, so a change of camera type must be possible without throwing away the whole system, but with a simple change in 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 takes place under time pressure during the recordings, so everything has to be work smoothly to make this process less stressful. Also, the system should require a low amount of training before use.

Robustness

The system must be capable of handling invalid inputs. This refers to the reliability, because crashes cannot be tolerated during a recording.

2 Software architecture views

2.1 Subsystem decomposition

The system is divided in two main parts. The front-end server, providing a tablet web interface, and the back-end server, proving the model. This decomposition makes it possible to change the front-end view of the system by for instance replacing it with a native app for other devices instead of a web interface. In Figure 1, a global overview of the system is given.

The front-end interface consists of an HTML, CSS and jQuery web interface and a NodeJS server, serving the web interface. The web interface communicates with the back-end and NodeJS server over the HTTP protocol, using HTTP requests. The NodeJS server provides all files necessary for the web interface to run, and configuration options. The web interface itself is responsible for the communication with the back-end server.

The back-end interface consists of a model written in Java. The back-end provides the central system and will perform all computations, control the camera's and handle communication with the preset database. These presets will be stored in a database. Information from this database can be retrieved using SQL queries which will be sent to the database. Next, we will further describe the various subsections.

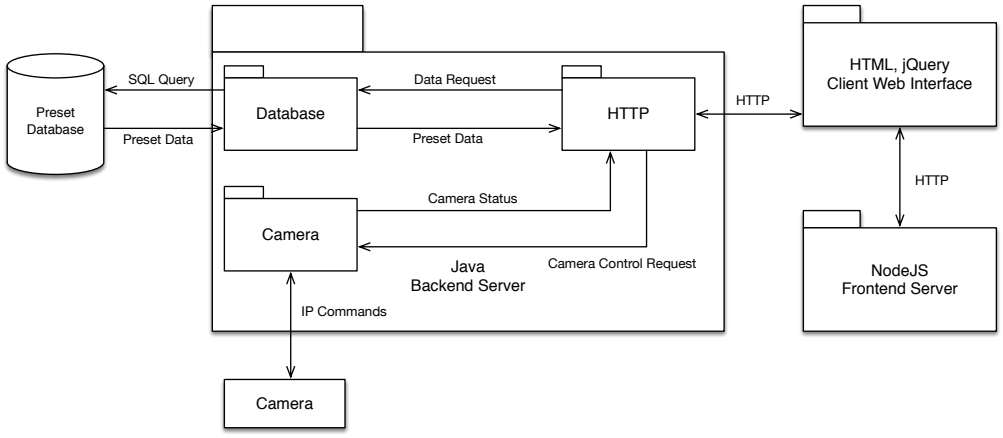

Figure 1: Global overview of the system

2.1.1 Java back-end

The Java back-end server is where all the main computations are done. The intelligent presets are calculated and corresponding commands are sent to the camera's. These commands are sent using the java camera module, implementing the camera communication for various camera types. If the client requests information about the camera's using HTTP requests, this will be processed by the HTTP module of the server. This module provides endpoints for controlling camera's, fetching preset information and general server status information.

2.1.2 NodeJS server

The NodeJS subsystem provides the resources needed for the clients view. It provides routes serving the HTML, JavaScript, jQuery and CSS files required to run the client. It can also provide the user with configuration for the web-interface such as the location of the back-end server to send requests to.

2.1.3 Clients view

The only thing the user will see of the program, is the client view. The client view is a tablet (touch) controlled web interface primarily designed for Apple iPad, however it should be working on multiple tablet brands. The clients view allows selection of camera's, selection of corresponding presets and control of a camera. It also provides the user with a live feed of all available cameras. This part is one of the most essential subsystems. If it fails, the user may not be able to use the system at all. It should be easy to use, and robust to failure. A global mockup of the user interface is given in Figure 2

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.

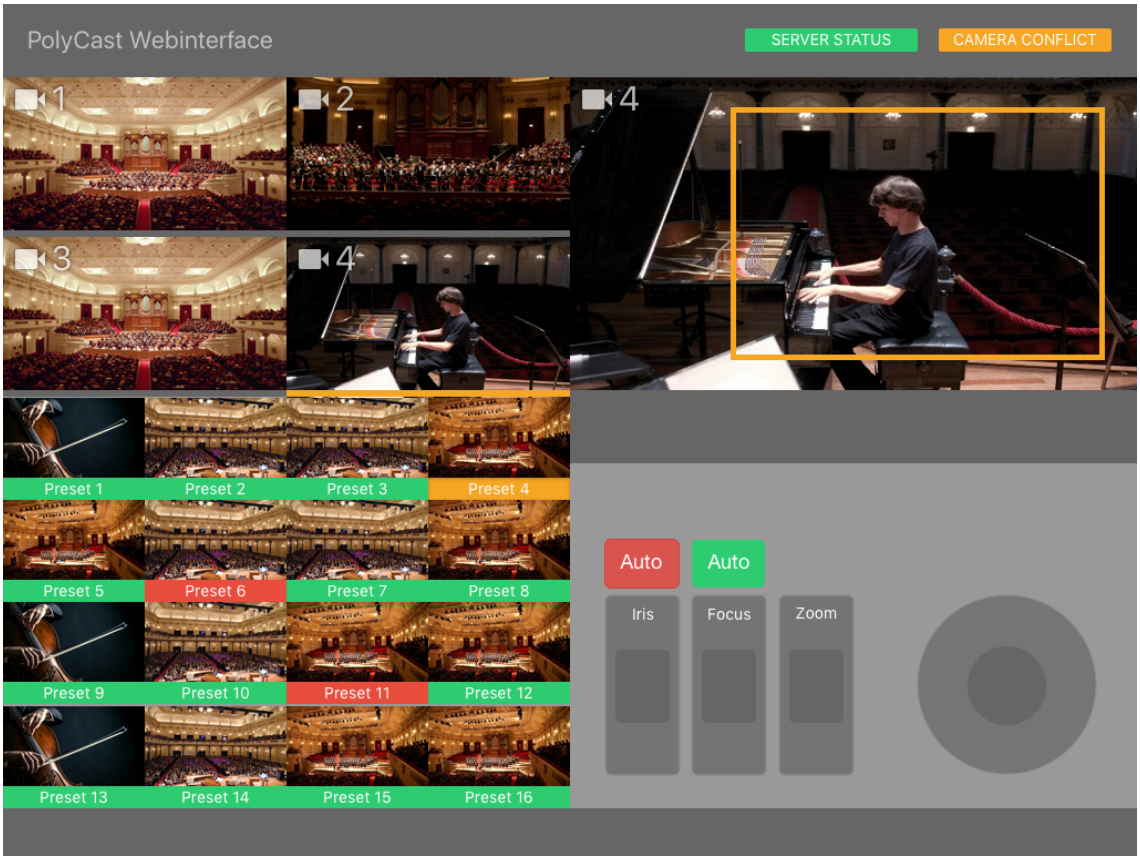

Figure 2: Mock-up of the user-interface

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. The database schema, is shown in Figure 3.

Database Design:

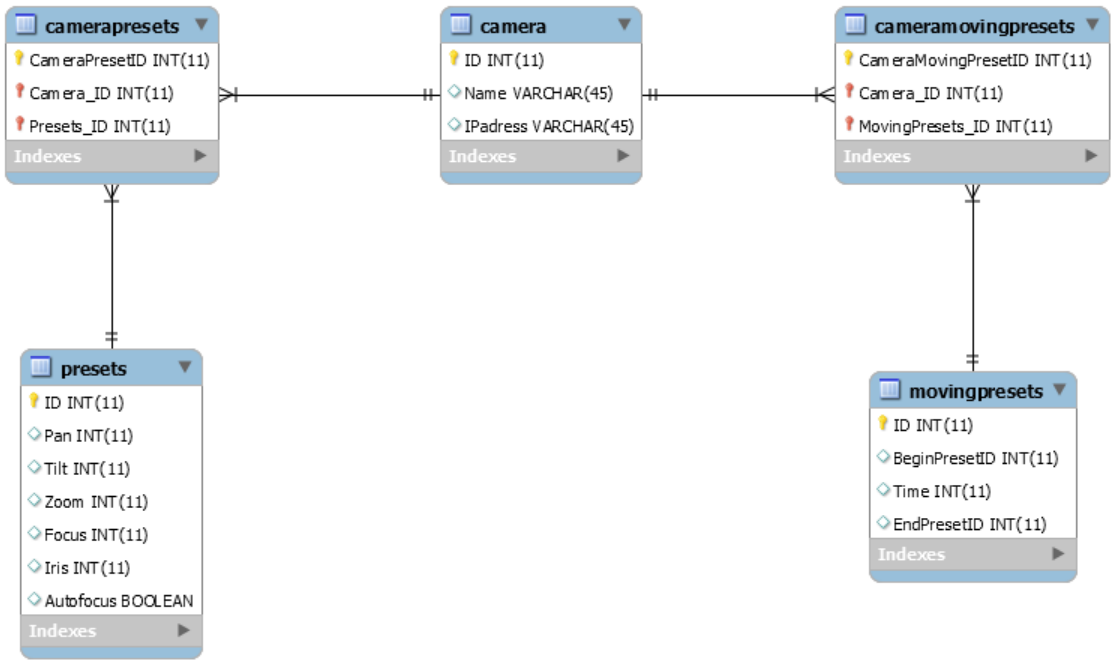

Figure 3: Database design

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

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