Software Design Description
StreamCam

Juan S. Carrillo Quinche

March 8, 2016

1. Scope

2. Referenced Documents

3. Requirements

4. Architectural Design

4.1 CSCI Architectural Design

4.2 CSCI Components

4.3 Concept of Execution

4.4 Interface Design

4.4.1 Interface Identification and Diagrams

4.4.2 Project-unique Identifier and Interface

4.4.3 Project Interactions

Directory Structure

Startup Control

5. CSCI Detailed Design

5.1 CSC 1 Detailed Design

5.2 CSC *n* Detailed Design

5.3 Hardware Interfacing

5.4 Intra-Level Processing and Data Transfer

5.5 Unit Test Scripts

6. Requirements Traceability

6.1 Introduction

Stream Cam is an Android application that will allow users to record and simultaneously stream videos to a remote database. Users can manage their videos on the database via a desktop web client.

6.1.1 System Objectives

The goal of StreamCam is to provide users a way to safely store videos that they record on their phones. By streaming their videos to a remote database, users can be sure that even if their phones are damaged or destroyed or confiscated their videos will remain saved.

Each video can have an associated location manifest which provides the user's location during the time the video was recorded. This is done in hopes that if user's need to provide legal evidence to support their account of a traffic accident, for example.

StreamCam should be able to handle losing connection to the database by streaming the stored video as soon as it gets internet connection.

Users will be able to manage their videos through a web client, which will provide a simple interface for them to manage and download their videos in the database.

6.1.2 Interfaces

The Android phone will communicate with the server in one of two ways. The client can provide server with HTTP requests for actions such as login, creating new users, or notifying the server that the phone is about to stream a video to the server. The Android phone will act as an RTP server to stream videos to the remote server, which will request the video stream from the client.

StreamCam API

The Android client will communicate with the StreamCam API and make general requests and authorized requests. General requests can be done by anyone, and they include login and creating new accounts. Authorized requests require that the user be signed in to StreamCam. Once they have been authorized through the use of web tokens, users can perform authorized requests.

1. The Android client can notify the server if the phone is about to start or end a stream.
2. The Android client can post the user's location when recording a video.
3. The web client can request to view the user's videos.
4. The web client can request to download a user's videos.
5. The web client can request to delete a user's videos.

Server-Database Interface

I make use the **pg** Node library, which allows me to write server-side CoffeeScript that interface with the database.

6.2 Architectural Design

My system will use a simple three-tier architecture. It will constist of clients, a server that will handle all the logic and requests from users, and a database that will store user information.

While this may change depending on what works best with the streaming library **libstreaming** that I am using, for now I consider a system where the Android phone serves as an RTP server that streams video to a VLC client on the side of the StreamCam server.

Below is a figure of the overall architecture.

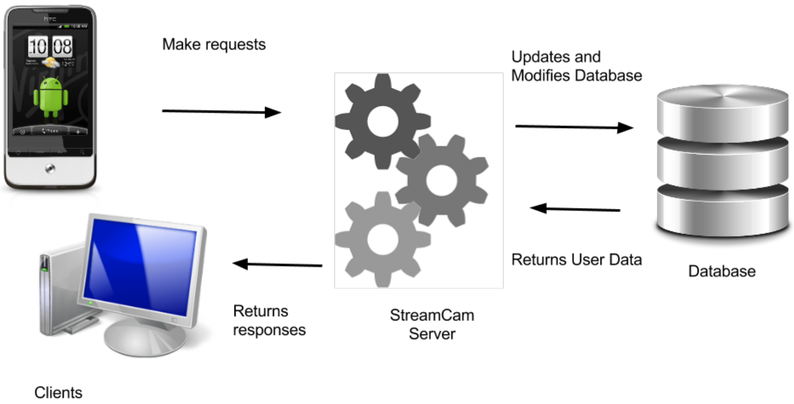

Figure 1: Figure of Architecture

Below is a figure that explains the streaming architecture.

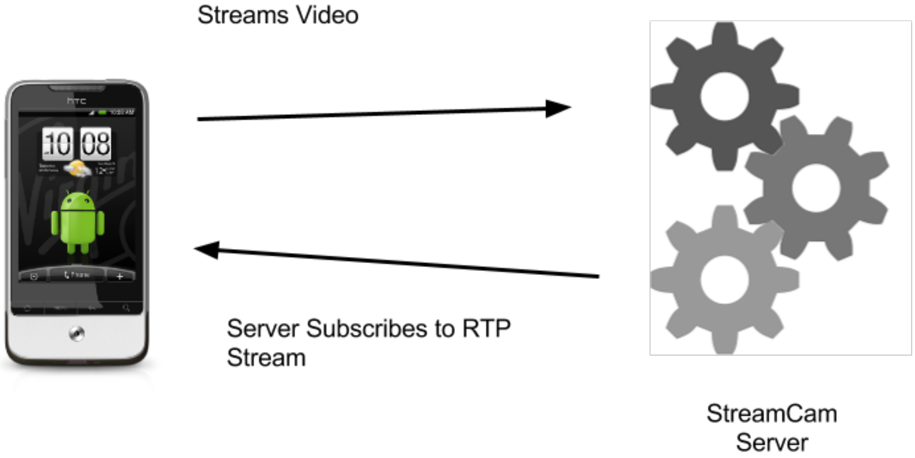

Figure 2: Figure of Architecture

6.2.1 Major Software Components

6.2.2 Major Software Interactions

6.2.3 Architectural Design Diagrams

6.3 CSC and CSU Descriptions

6.3.1 Class Descriptions

Class Description 1

Class Description 2

Class Description n

6.3.2 Detailed Interface Descriptions

6.3.3 Detailed Data Structure Descriptions

6.3.4 Detailed Design Diagrams

6.4 Database Design and Description

6.4.1 Database Design ER Diagram

6.4.2 Database Access

6.4.3 Database Security

7. Notes

8. Acronyms and Abbreviations