Video Tag Portal

System Design Document | Current Version [0.3.0]

Prepared By:

Anudeep Potlapally

Travis Rous

Revision History

Date	*Authors*	*Version*	*Comments*
9/20/12	*Travis Rous & Anudeep Potlapally*	*1.0.0*	*Initial version*
10/4/12	*Travis Rous & Anudeep Potlapally*	*1.1.0*	*Added prototype 2*
11/2/12	*Travis Rous & Anudeep Potlapally*	*1.2.0*	*Added prototype 2*
12/6/12	*Travis Rous & Anudeep Potlapally*	*1.3.0*	*Added prototype 3*

Table of Contents

1.0 Overview

The goal of the document is to show the way that we have implemented the video tag portal. There will be details on the purpose, goals, as well as requirements and implementation.

1.1 Scope

There will be a detailed discussion on how the project is implemented and what tools are necessary to create the project.

1.2 Purpose

We are to make a video tagging system that allows users to tag points in videos with a various number of predefined tags. After tagging a video, the user then has the ability to share the video with his or her friends. The friends then will have the ability to also tag the shared video. The goal of the system is to make videos more socially connectable between friends.

1.2.1 Major System Component #1: Database

The Database will hold all of the information that the system will need in order to function. The database will contain all of the information to control the users, friends and the tags used by the player.

1.2.2 Major System Component #2: APIs

We will need many APIs that will give use access to the different videos and displaying the videos.

1.2.3 Major System Component #3: YouTube and other video Hubs

We will need to have support for many different videos from major video sites on the internet as well as being able to upload the videos that users want to add.

1.2.4 Major System Component #4: Popcorn.js

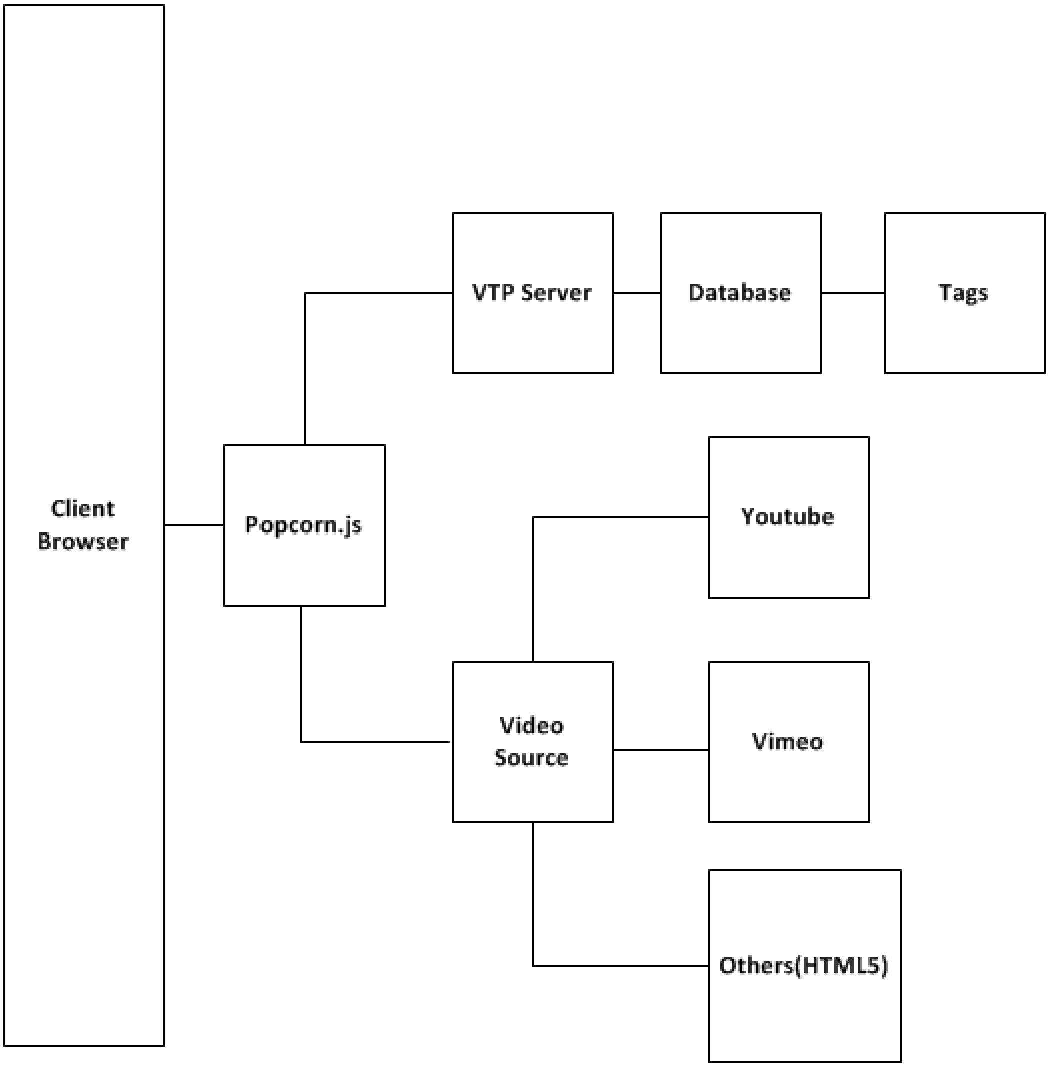
The popcorn framework will be what is used to make the tags appear and disappear dynamically.

1.3 Systems Goals

System goals include giving the user the ability to use our application to make video part of the online social experience. We want the user to have a "Natural" user interface to make the tagging process simple for the user. Also one major goal is to make the product as much as possible a social networking site.

1.4 System Overview and Diagram

Much of the project will depend on the various APIs that are needed to make the videos display and show up well on the web page. We will also need to use the APIs to manage the users that access the site using Facebook and Google. In addition to the APIs needed in the project there will be a database that is used to manage the different users and the tasks that they will be able to do. There are many different things that will be going on per user such as knowing there favorites to logging in to the site. The most significant system component is the popcorn.js framework that will actually allow us to make time based tags on the fly which is the main part of the project.



1.5 Technologies Overview

PHP: A scripting language used in web development that can be embedded into HTML.

HTML5: A markup language that is a web standard for creating web sites, HTML5 adds many other features that the previous HTML versions.

JavaScript: A scripting language that make is used in making more advanced user interfaces and add scripts to HTML.

Popcorn.js: A JavaScript frame work for editing media in real time, based on timings of a piece of media.

Google APIs: There a number of various API that we are planning to use in order to make the use of YouTube videos simpler and also make the overall project simpler.

Facebook API: We needed to use the Facebook API in order to use the login from Facebook and to see the friend list for a given user.

jQuery: A JavaScript framework that makes scripting with JavaScript simpler and allows for many dynamic features in a webpage.

Viemo: A website for hosting videos.

YouTube: A website for hosting videos.

MySQL: A database that is widely known and supported by many applications.

2.0 Project Overview

This section will provide some of the team roles and the way the project is managed and kept on track.

2.1 Team Members and Roles

The two members of the project are Anudeep Potlapally and Travis Rous. Each member is responsible for attending the scrum meetings and doing what is assigned to them after each meeting.

2.2 Project Management Approach

We are using the Scrum development cycle to manage the project. This consists of almost daily scrum meetings that are less than 5 minutes each. There is a scrum master that is the lead on the project who is really there to keep the team going and on track. There are sprints that range from 2 to 3 weeks. We are using GitHub and Trello to manage our code and progress over time. Trello is like a digital board with task card that are able to be moved from to show the progress we have done in the current sprint and assign some cards to have a higher priority than others to show the significance of each task or what is left for each task at hand.

2.3 Phase Overview

If the system will be implemented in phases, describe those phases/sub-phases (design, implementation, testing, delivery) and the various milestones in this section. This section should also contain a correlation between the phases of development and the associated versioning of the system, i.e. major version, minor version, revision.

The major phases in the project are as follows:

2.3.1 Phase 1: Initial prototype

The initial prototype was needed to get our idea out to the client and make sure that we were on the same page. This was completed in sprint 1.

2.3.2 Phase 2: Database Design

The Database is currently only for making the application useable and is not yet a final. The Database will not be final until we know about the outcome of some of the items that are on the remaining backlog.

2.3.3 Phase 3: Design and Connect pages

This is the current phase we are working on at the end of sprint 3. This phase is a large multi-sprint phase that is just really adding in all of the pages to the site and making them usable and communicating with each other.

2.3.4 Phase 4: Giving the users an attractive interface

This phase will be used to give the users the "Natural" interface for using the application. This phase will also be used to give the users an attractive interface, which means the user will not run from the application because it looks bad.

3.0 Requirements

For the detailed list of requirements check the Software Requirements Document. Here is the undetailed list of requirements (User stories that were initially provided).

- Play the videos within a web application
- Make the application have the ability to play local videos
- Possible integration with sling.com
- The first time viewing of the video should store all the information of the video that is being played
- Design some kind of login system that allows us to know who one's friends are
- We need to be able to pause a video at any one moment and place one of many kinds of tags
- We have a set of predefined Tags to choose from
- Need to have the ability to change the list of tags
- We need to set up a filter system to filter out tags placed on a video

4.0 Design and Implementation

This section is used to describe the design details for each of the major components in the system. This section is not brief and requires the necessary detail that can be used by the reader to truly understand the architecture and implementation details without having to dig into the code.

4.1 Popcorn.js

4.1.1 Technologies Used

We are using open source library “Popcorn.js” developed by Mozilla Foundation. Popcorn.js allows VTP to interact with the limited 3rd party video sources like YouTube, Vimeo and any HTML5 videos either saved locally or from other 3rd party sources.

4.1.2 Component Overview

Features:

Play local videos in HTML5 and interact with them

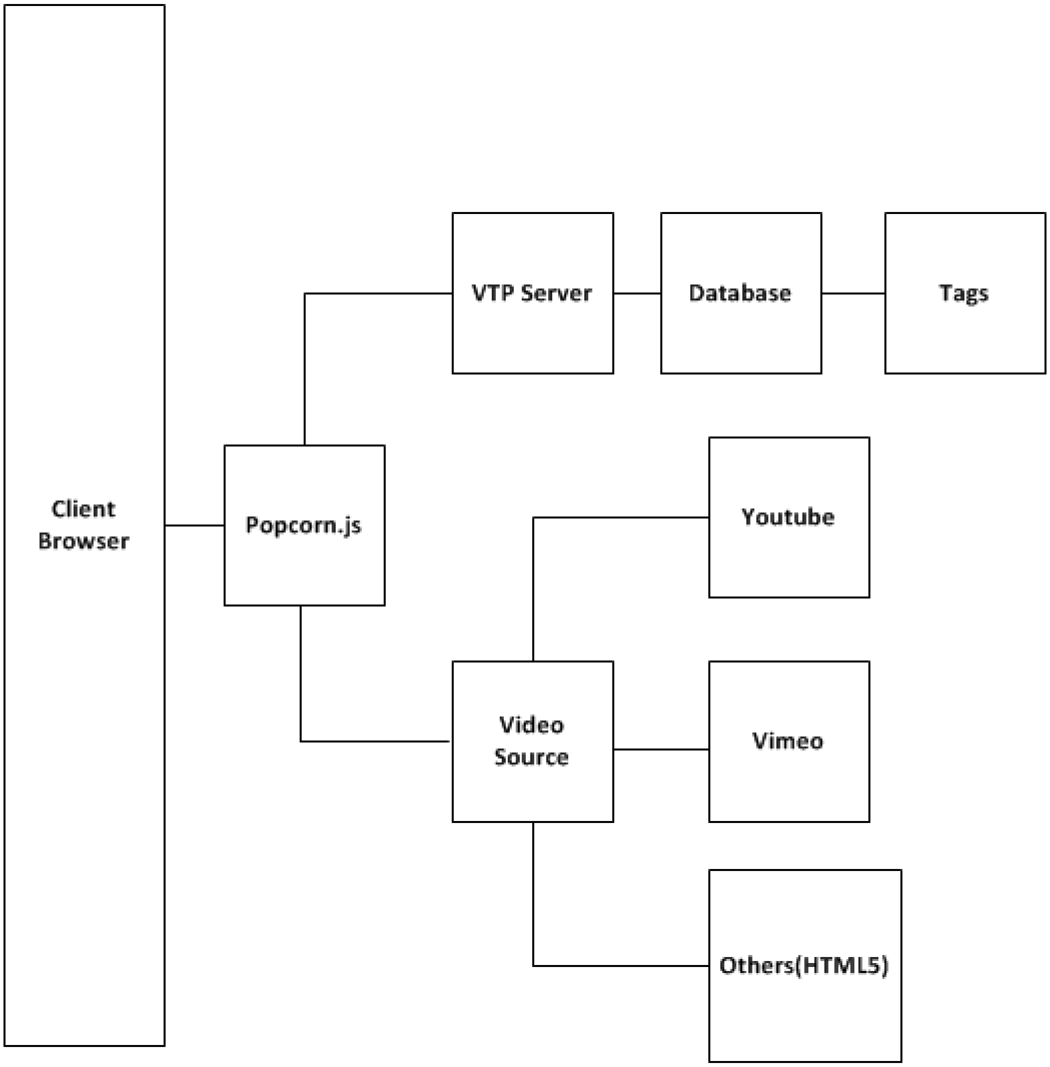
Play both HTML5 and flash videos from YouTube and Vimeo.

Triggers JavaScript functions when video is on a particular time.

4.1.3 Phase Overview

Phase I:
Embed videos from 3rd party sources.
Phase II:
Implement feature to play videos from YouTube using Popcorn.js.
Link video player to HTML elements to display hard-coded tag information.
Use Database to save and retrieve tags.
Phase III:
Implement feature to play videos from other possible sources and able to play display multiple tag types simultaneously..

4.1.4 Architecture Diagram



4.1.5 Design Details

This is where the details are presented and may contain subsections.

4.2 Database Management

4.2.1 Technologies Used

Created DbConnector class in PHP to interact with MySQL Database. All queries are written as individual functions in DbConnector class.

4.2.2 Component Overview

Features:

Connect to database and authenticate
Execute frequent queries from functions
Disconnect when PHP execution is completed

4.2.3 Phase Overview

Phase I:

Start database design

Phase II:

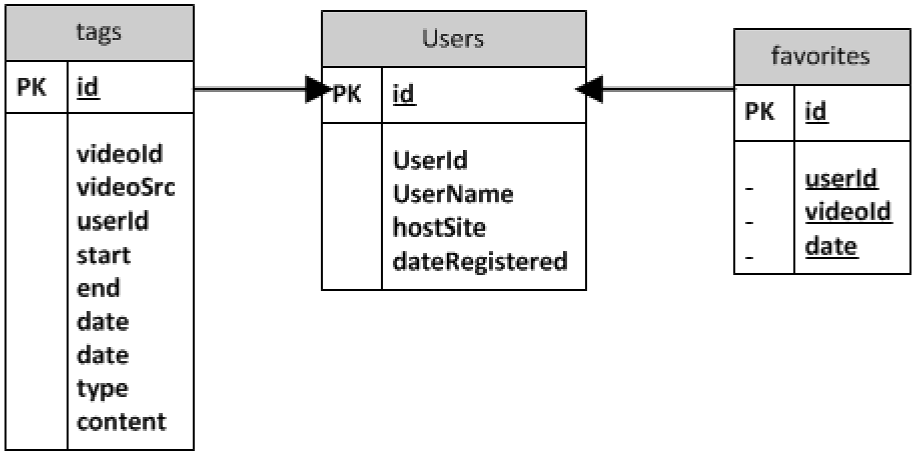
Implement Database design.

Write DbConnector class

Phase III:

Add queries as functions in DbConnector class

4.2.4 Database Schema



tags	
PK	id
	videoId videoSrc userId start end date date type content

Users	
PK	id
	UserId UserName hostSite dateRegistered

favorites	
PK	id
-	userId
-	videoId
-	date

4.2.5 Design Details

This is where the details are presented and may contain subsections.

4.3 User Authentication

4.3.1 Technologies Used

Implement user authentications using Google API PHP Client and Facebook SDK.

4.3.2 Component Overview

Features:

Facebook SDK:

Authenticate users; access basic information

Get friends list to compare with the VTP database for filtering video tags

Google API PHP Client:

Authenticate user accounts, access basic information

Upload videos into their YouTube account. (Requires YouTube Data API)

4.3.3 Phase Overview

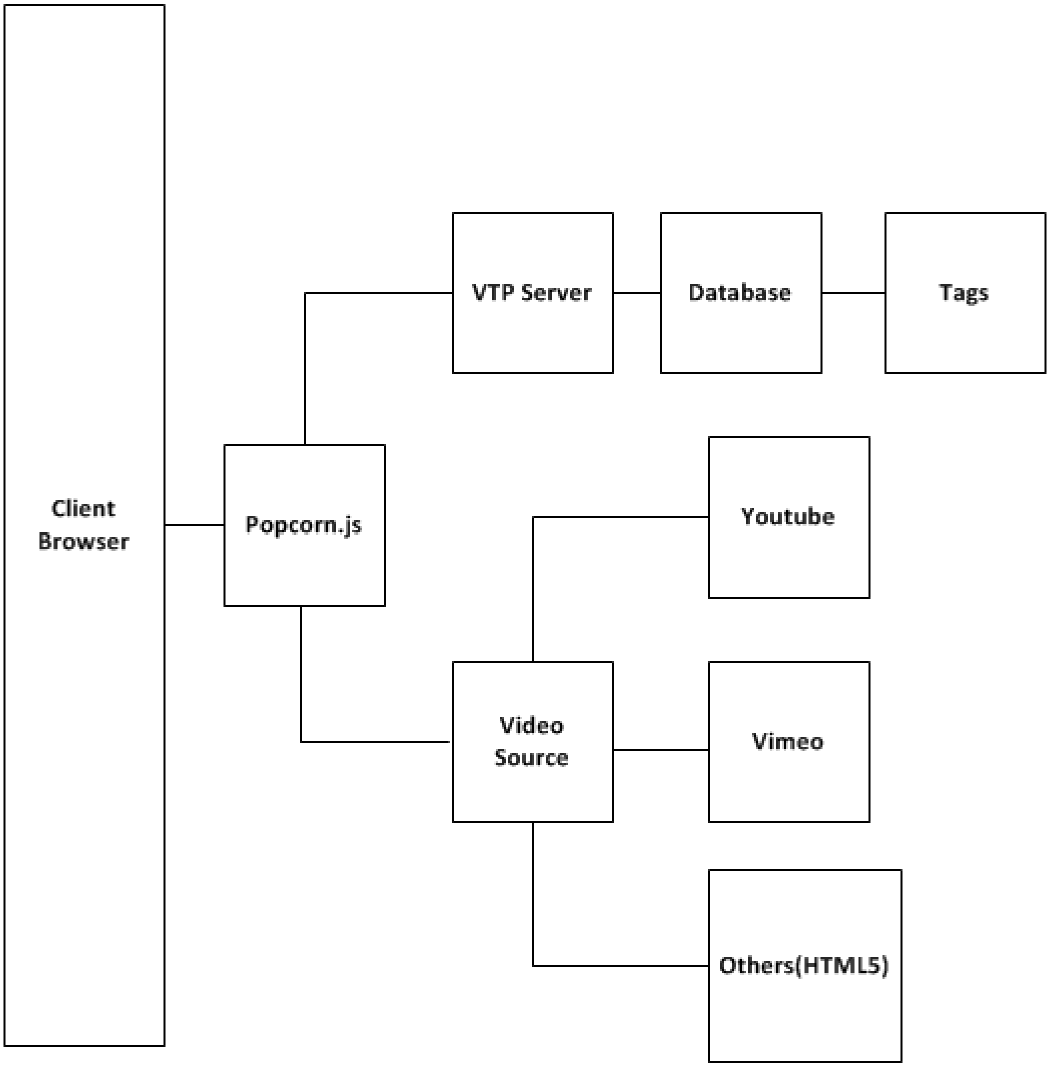
Phase I:

Research on both the API's

Phase II:

Implement both API and save basic information into Database.

4.3.4 Architecture Diagram



4.3.5 Data Flow Diagram

It is important to build and maintain a data flow diagram. However, it may be that a component is best described visually with an architecture diagram.

4.3.6 Design Details

This is where the details are presented and may contain subsections.

5.0 System and Unit Testing

5.1.1 Popcorn

We will need to make sure the popcorn is behaving as we expect, by checking the timings and making sure that they are consistent with what is happening on-screen.

5.1.2 Videos

We need to make sure that the videos are being displayed correctly and are sized properly. The videos need to be able to be played while to tags are being updated.

5.1.3 Database

We need to make sure the database is secure as well as able to be responsive for the users accessing it.

5.1.4 Host Server

Security is the major concern that is associated with the Host Server.

5.1.5 JavaScript

We will be using the JSLInt to test the quality of the JavaScript functions.

5.1.6 Range of Browsers

We will need to test the functionality of the different browsers on our site.

5.2 Overview

At this time we only have one framework for testing and that is the JSLint framework described below.

5.3 Dependencies

JSLint for testing the individual JavaScript functions to make sure that they are living up to some standards.

http://www.jslint.com/ --- the webpage for the tester.

5.4 Test Setup and Execution

The setup of the JSLint is go to the web sited and paste in the code that you want to test. (There are also command line versions for download)

6.0 Development Environment

The basic purpose for this section is to give a developer all of the necessary information to setup their development environment to run, test, and/or develop.

6.1 Development Tools

Development tools used in this project are Notepad++, PHPMyAdmin, and XAMPP.

Notepad++ is used just to do the markups on the code to see where the keyword are used and manage brackets.

PHPMyAdmin was used in the database management.

XAMPP is used to host the web server and MySQL server in one simple setup.

6.2 Source Control

For the source control we are using the GitHub repository that uses a windows interface to access the repository. GitHub is completely open to the public to see unless you pay for a private service.

6.3 Dependencies

Describe all dependencies associated with developing the system.

6.4 Build Environment

The project will need to be deployed on a PHP server and a MySQL server will need to have the tables setup using a .sql script that will be sent with the project. There will need to be a connection between the server and the PHP using a database connection in PHP.

6.5 Development Machine Setup

In order to make a machine able to develop you will need to download Notepad++ and have access to a PHP server and MySQL server.

7.0 Release | Setup | Deployment

Upon release the product will be hosted on a server that will have the bandwidth high enough to support thousands of users at a time. The project will need to be deployed on a PHP server and a MySQL server will need to have the tables setup using a .sql script that will be sent with the project. There will need to be a connection between the server and the PHP using a database connection in PHP.

7.1 User Properties

There will be one final website that will be used for the hosting that the users will connect to.

Appendix I: List of Figures

Appendix II: Supporting Information and Details

This document will contain several appendices used as a way to separate out major component details, logic details, or tables of information. Use of this structure will help keep the document clean, readable, and organized.