MeTube Documentation
Database Management Systems (CpSc 4620)

Clemson University
Dr. James Z. Wang

Chris Levesque
Nathan Jones
Z. Schafer LeMieux

Project Due: April 19th, 2016

Contents:

System Design

MeTube is an implementation of the software architecture pattern, “Model View Controller” or MVC. MVC is a design technique that organizes data into a “model,” provides a means of accessing this data via “controller,” and provides a user interface to interact with the system, or “view.”

The model represents the real data and components of the system. This is represented through a MySQL database. MySQL lets us create databases and tables with attributes to represent the major elements of the structure. In our MeTube system the “model” or data was created by MySQL tables of the following:

account
channelmedia
channels
comments
favorites
media
messages
playlistmedia
playlists
subs

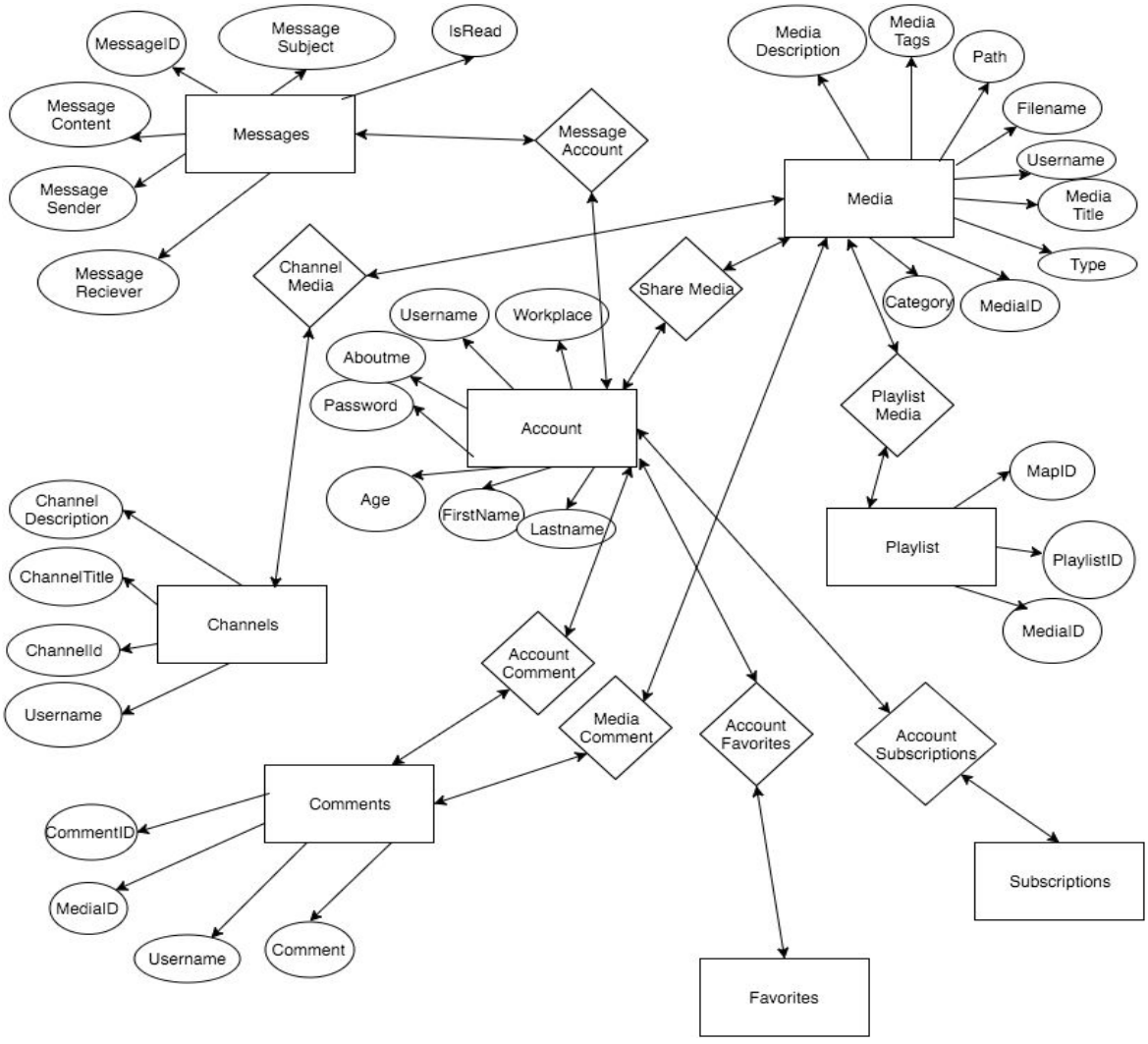
Each of these tables has its own fields or attributes that describe the name of the table. For example, the table account has a username, password, age, workplace, aboutme, firstname, and lastname. Likewise, media has attributes such as filename, username, type, etc.

Controller is implemented using PHP and MySQL queries. PHP is a server side scripting language that is commonly used to perform MySQL queries. PHP is used as a means of manipulating the data or “controlling” it. Use of this technology is demonstrated in files that end in “_process.php, or, for example media_upload_process.php. These “_process.php” files use an HTML feature called HTML Forms. Using HTML forms, these process files, or “controller” files perform the real work of the system. This leads us to the question of how these processes get initiated. This is where “view.” comes in.

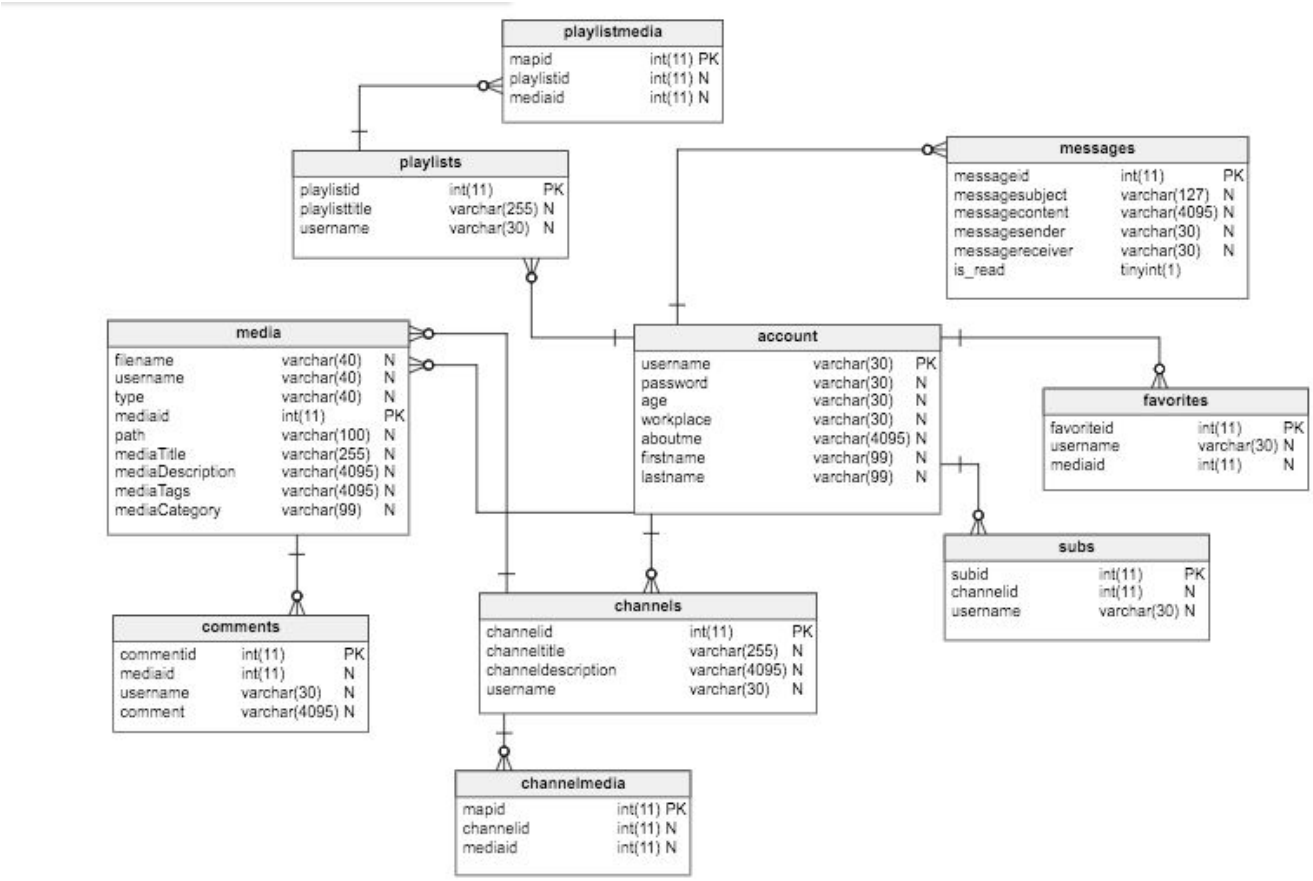
View supplies the user with an interface to interact with the controller previously described. In other words, this is where the user is able to click around the website and manage their media. To create this we used HTML, CSS, and PHP. HTML is a mark up language which allows us to display data. CSS is used to style this data. Users navigate to the URL of our website and the system lands them on “index.php.” This file has HTML embedded into it. All other actions carried out by the user either call a controller php process or land the user on a new PHP file.

In Summary, the Model View Controller architecture pattern is the main design of our system. Model being MySQL, controller being PHP and MySQL queries, and view being HTML, CSS, and PHP.

ER Diagram



Database Schema



Function Design

To come up with a function design we focused on the actions to be carried out by the user. To start the web page we knew there would have to be a registration page for new users and a login for each of the users. The registration page is basic form that new users must fill out. The field include a unique username, password, first name, last name, and age. The password must match the repeat password field as well. When the user submit the site goes to the view media page. Login has the users fill in their username and password and it checks with the database to see if there information is correct and then has them go to the media page. These profile details can be updated if the user goes to their profile and clicks “update profile”. Update profile is a form similar to registration page with the addition of an about me the user can fill out. If the user leaves a field blank it does not change their profile information. On that page they can change their password, name, age, and even add an about me. The username, first name, last name and about me displays on the user’s profile that other users logged in or not can view.

The next group of features that were key for this site to work is a feature to upload media, download the media, and view. A user must upload media by clicking on the upload media button that can be found on the browse media page or my media page. This opens a page that the user will put in metadata about the media they are going to upload. The metadata includes a title, description, and tags. The user can also select a category for the media to be grouped under and add it to a channel. The media can be viewed by clicking on it in any of the lists that the media can show up in or on the user that uploaded the media's profile. That will then link you to a page that presents the media and some of its meta information. The title of the media is displayed at the top, followed by the media itself. There are 4 options for the media display. First are images that just show right on the page, a video that comes up in a video player, an audio file that has an audio player that will play the media, and other media that does not show up but has a download link. The download link allows user to download the media onto their computer. The description then comes below the download. This page also includes a few other features that will be discussed later.

The organization of the media that has been uploaded was one of the largest sections of this project. There are several ways to browse media, the first being by category. The categories are predefined by the owner of the site and selected by the user when uploading media. ON the category page the user can select a category to view. This then populates a list with all of the media in that category for the user to view. The next two features are rather similar, playlist and channels are two different ways to have lists of media. A playlist is for yourself to list and collect media on and stays private so other users can not see the list. A channel is a lists of media a user can create and add media to. Making a channel can be done on the My Channels page at the bottom by clicking the Add Channel button. This takes them to the create channel page where the user can supply a title and description for the channel. These bits of information show up on the channel's page when a user is viewing it. The channel that was just created can be seen on their profile under Channels. Other users can subscribe to the channel one of two ways. The first being clicking the subscribe to channel button on the view media page or by finding the channel on a user's profile on and subscribing there. A user can then view any channels they have subscribed to on the My subscriptions. For playlists the user can go to My playlist and at the bottom type in a playlist title and click the create button to create one. For media to be added to the playlist the user can select it from a drop down menu on the view media page. A user can also create a new playlist with that drop down menu. This will create the playlist and add the media to the page. The final bit of browsing is a favorites list. For a user to favorite any media they can click the favorite button on the view media page. This then adds it to the favorites list when is viewed by clicking My favorites on the browse media page. If a user does not mean to favorite a certain media or they no longer favorite it the favorite button now becomes an unfavorite button and clicking it removes the favorite. This can also be done from the favorites list page.

To communicate with another user can be done through messaging. To begin messaging another user either go to their profile and click Send Message or go to the messages page which is linked in the top right corner. This begins a message thread with the user. A test field is there to input the title and a text area for the content of the message. Then click send message and it adds it to the bottom of the message thread. If a user has receive a message a

badge will appear on the Message link in the top right. That link will bring you into messages and show you the users that have there are message thread with. When a thread with a new message opens up the unread message turns blue for a few seconds to indicate it is the new message. To write a response fill out the message content on the bottom of the thread and hit send.

Users can also comment on individual media. At the bottom of each media page there is a comments section. To write a comment a user has to fill out the text area with whatever they would like to say and hit submit comment. The comment then shows up below the media. The older comments are at the top and newer ones that the bottom. If the user wishes they can delete the comment by clicking delete right next to their username on the comment.

The last feature to implement is the search feature. The search bar sits at the top of every page on the site. The search is run with the tags that were given to each media file when uploaded to the website. A user can type a single word or multiple word tag to search as well as partial words and 2 different tags. When the user clicks search it then brings up the list of all media that were found in the search. At the end of all of the results the page displays the message “No more search results to display.”

Implementation Details

The first decision we had to make with regards to implementation was exactly what information needed to be stored. We tried to keep the number of tables and data in them to a minimum, and used the relational model to extract the data that was required. Since we used a bootstrap almost all of the styling was done simply by including class names and therefore will not be mentioned much. Some of the trickier parts of the implementation were: a few mysql queries that required joins, submitting forms using buttons that were not contained in the form, using links to submit forms (to allow for extra values to be passed), and making pages work whether you were logged in or not and whether the page belonged to you or not.

The earliest and some of the most important parts of the project were the login and registration pages. The important aspects of these pages were provided with the template, but we added some additional fields. The information from these fields (only some are required) are inserted into the account table in the case of registration, verified against the account table in the case of login, and the user is then logged in. The registration page uses the function user_exist_check() to verify that the user exists and then to do the insertion into the table.

The profile update page was created with similar html and php to the registration page. The fields are pre-populated with the current values of the user’s account and all can be changed except for user. Since user is the primary key of the table it is immutable. Any fields that are left blank will not be updated in the table, any other edited fields are updated in the table. The alter table is done by calling the function update_profile_info() which takes in the information and prepares the mysql query.

The basic functionality of upload was provided in the template, we added fields to include more information about the file. Title, tags, and category are particularly important here

as they are widely used throughout our code. Title is displayed almost everywhere you see a list of media, tags are used as a basis for searching the media on, and category is used to sort the media in the category section of the site.

The ability to download and view media is provided in media.php. The download button is a simple html link with an onclick attribute set to run a simple javascript function that downloads the media. View gets the type of media and selects the appropriate way to display it depending on the type.

At the top of most pages that contain some sort of list of media there is a bar that allows for quick navigation between these different lists. The first in the list is view media by category. We decided to hardcode a small number of categories so the users could see media broadly as there are many other ways for them to specify information about their own media. Since category is a column in our media table we can select all media from a category simply with "SELECT * from media WHERE mediaCategory='$category';" Within an html table, we then loop through the results one row at a time, printing the media title, which links to the media page, and a download button. This table will be copied many times, simply using different queries to change the content that is shown.

Channels and subscriptions were one of the larger tasks for this project. There are multiple conditions here that alter the content of the page. Much of this alteration is simply done with simple php if statements, although many of our links are actually small javascript functions to allow us to submit a form instead of going to a link with <a href="">. No matter what you get a table showing a list of the channels from the user that you selected. My channels links to channels for the logged in user, which includes buttons to delete the channel and to add a new channel. If you go to a different user's page you will not see the delete option for that channel. Clicking on a channel brings up the channel title, description, and media. If it is your channel, you will see an option to remove or add media for the playlist. It is worth noting that we decided media could only be in 1 channel at any given time. If you are logged in and visiting another user's channel, you will see an option to subscribe or unsubscribe depending if you are already subscribed or not. If you are not logged in you will only see the list of media in the channel with none of these options. There is also a subscription page for logged in users that allow them to access any channel they are currently subscribed to.

Playlists, while similar in structure to channels, have similar code to channels as well, albeit much simpler. Since playlists are private to the user, there are no complex if statements required to check who is the owner of the page and whether or not you are logged in. From the playlists page you can see a table of all the playlists you currently have as well as the option to delete or add playlists, again using simple if statements. Selecting a playlist displays an html table of all media in the playlist with the option to remove media from the selected playlist. You may also add media to a playlist from the media's page or to add the media to a new playlist. This used javascript, that if the "Add new playlist" option was selected would hide the add to playlist button and pop up with a text field with its own add button. The user can then enter the new playlist name and use the new button to submit. Favorites are built similar to playlists, and use the same table to display media. The main difference is that favorites will query the favorites table, while playlists will query the playlists and playlistmedia tables.

The messaging feature contained one of our more complex mysql queries. The messages table stored the sender and receiver, but a message thread contained both messages where the other person was the sender and I was the receiver or visa-versa. The first pass of the query was incorrect, but after thinking about it in normal English terms we were able to correctly query the messages table. Another feature we decided to include in messaging was a javascript function that would set a timeout and after a few seconds all new messages would turn from a blue background to a normal color. This as well as a badge on the messages button showing the number of unread messages (defined by is_read in the messages table) allowed users to easily see when the received a new message.

Comments, like almost every other html table, was created inside a php while loop that looped over a returned mysql_query. We simply selected all comments where the mediaid was the same as the current page's mediaid. We also added functionality depending on if the comment was yours or not. If it was yours you can delete the comment, if it is someone else's, clicking their name will take you to their profile.

Searching uses a php function that is new to our project . The words that a user types in the search bar are passed into a ta function called explode. This returns an array containing every word the user typed in. It then searches then a foreach loop is made to loop through each element for the array that explode returned. This loop creates a sql query to find that string in the tags if any media. The query is constructed with Like "%searchword% so that if the word is included in a sentence or in part of a word it will be found. This loop keeps an array to store all media ids found and the in_array() function is used to check if that media has been found and displayed to the list already. This makes it so there are not multiple instances of any media that could contain multiple tags.

Test Cases

User Account

- User can register
  - What if field is left blank and submit is clicked
- Password is confirmed
- Email is validated
- User can update profile
- User can log out

Data Sharing

- User can upload
  - Upload for files over 5 mb, 10 mb
- User can attach metadata to the file

Media Organization

- User can organize media in favorites and playlists and add and delete into both
- User can organize media into a broadcasting channel

User Interaction

User can send and receive messages to another user
User can add or delete a comment on media
User can make media “commentable” or not

Search

User can search based on keyword

Testing Results

User Account

User can successfully register. The user account feature does not validate for an email address. Our system only validates a username found in the database. Another interesting thing to note is when fields are left blank. The system lets you proceed but doesn’t inform the user about what happened or if the register login or profile update was successful. There are some cases however such as login and register where error messages are displayed. No requirements for a password strength but, password confirmation works correctly.

Data Sharing

User can successfully share media. User cannot upload large files. User can upload and download all file types successfully.
User can also attach metadata.

Media Organization

User can successfully add and delete from channel, playlist, or list of favorites and organize their media in such a manner.

User Interaction

User can successfully send and receive messages to and from other users. A user cannot send a message to himself and the error case is handled gracefully.

Search

User can successfully search based on keyword. One interesting thing to note however is that it doesn’t try to match filenames or titles. It works strictly for keywords provided at upload time.

User Manual or Instruction

Nav bar - the bar at the top of the page, that is always on every page.
Browse bar - the bar that appears on some pages when logged in, gray and below the "Welcome to MeTube" greeting

Registration: select Register from the right side of the nav bar or the main page, fill in the fields with appropriate information (username, create password, and repeat password are required) then press Submit
Login: select Login from the right side of nav bar, enter your username and password, then press Login
Logout: select Logout from the right side of the nav bar
Profile Update: after logging in, select Profile from the right side of the nav bar, select Update Profile at the top of the page, update any information you want to update, then press Update
Upload & Meta Information: after logging in, select Media from the left side of the nav bar, click Upload Media, enter the meta information, click browse, select a file, click upload
Download/View: select Media from the left side of the nav bar (or find any page that contains links to media), click a media to view, download will appear beneath the media. This page also allows many features including subscribe to channel, favorite, add to playlist, and comments.
Browse by Category: select Media from the left side of the nav bar if there is no browse bar, select Categories from the left side of the browse bar, select a category
Channels: select Media from the left side of the nav bar if there is no browse bar, select My Channels from the center of the browse bar, select a channel to view
Subscriptions: select Media from the left side of the nav bar if there is no browse bar, select My Subscriptions from the right side of the browse bar, select a channel to view
Playlists: select Media from the left side of the nav bar if there is no browse bar, select My Channels from the right side of the browse bar, select a playlist to view
Favorites: select Media from the left side of the nav bar if there is no browse bar, select My Favorites from the left side of the browse bar
Messaging: select Messages from the right side of the nav bar, select a thread to view, or start a new one by entering a username
Comments: view any media (see Download/View), scroll down, enter text, click Submit Comment
Search: enter text in search field on the nav bar, press Submit