Si Thu Zaw
G5223816T
BTID11701A
T7177793

Rapid Development Systems ICA

Part A – Design Document for Video Tutorial Website

31 August 2018

Teesside University

Rapid Development Systems (**COM2059-N**)

Student ID: T7177793

Student Name: Si Thu Zaw

Lecturer Name: Ms. Manian Latha

Table of Contents

List of Figures

Si Thu Zaw
G5223816T
BTID11701A
T7177793

1.0 Introduction

The following document contains the requirements and planning for the application, *Project: Muses*, including the requirements identified, functionality and content intended, technical and design specifications, prototype user interface mock-ups and security policies.

The main purpose of the application is to provide a platform for sharing videos as tutorials under many categories, which are to be viewable to both visitors and members of the web site. These tutorials will be contributed by the members of the website and they will be encouraged to structure their videos to be educational in nature. The tutorials must fall under ten categories when they are uploaded. The users will also be able to add likes or dislikes and comments to the tutorials to provide better feedback to the other users.

2.0 Requirements

2.1 Content

The following is a content model representing the main types of information the system would need to handle (Figure 1).

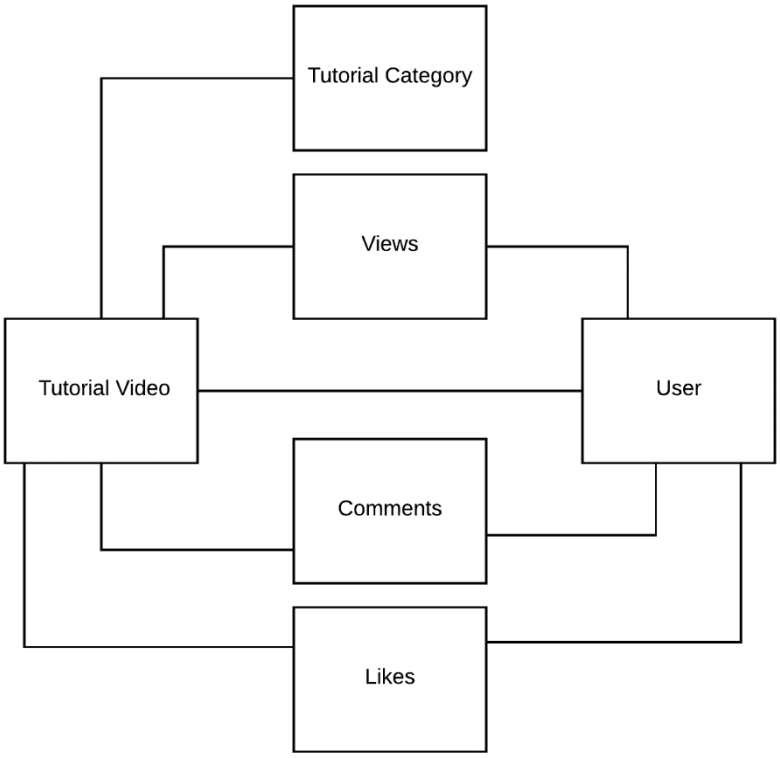

Figure 1: Content Model of Project: Muses

The site will host tutorials in the form of videos that shows the users how to do various things. The content of these tutorials, length, style, delivery, etc, will be determined by the users who uploaded them to their profile. The tutorials will be classified under specific categories to make browsing and searching through them simpler.

The listed categories are:

- Around the House DIY
- Arts and Craft
- Arduino
- Cooking
- Pets
- Modern Internet Phenomenon
- Life Hacks/Skills
- 3d Printing
- Environmental Friendliness
- Gardening

2.2 Functionality

The application's main purpose is to function as a platform to host various tutorials in the form of videos that are uploaded by the users, sorted into various categories, along with supporting functions such as tools for managing videos, feedback mechanism in the form of likes and other administrative tools.

The guests to the website are able to view all the tutorials on the platform and are able to sign up as members to the platform. Once they are registered members, they will be able to upload their own videos, make changes to them after uploading and delete them at their discretion. They will also be able to 'like' videos to help vote for videos that they feel are good.

Administrators of the platform have access to more functions in addition to the functions that the members have such as the ability to remove other members' videos from the platform, managing the users such as resetting their passwords or deleting their accounts from the system.

2.3 Users Roles

The diagram shows types of users that can use the Project: Muses website and the functions they have authorisation to access.

Guest: Users accessing the application that are not logged in with a known account. Guests are able to view videos uploaded by the members on the platform, but they will not be able to like/dislike videos or upload videos.

Member: Users accessing the application that are logged in with a registered account that is assigned the 'member' user role. Members of the website can upload their own tutorials and manage them and like/dislike videos on the platform.

Administrator: Users accessing the application that are logged in with a registered account registered as the 'admin' user role. Administrators are responsible for moderating the content displayed on the application and managing issues related to accounts. They are also capable of using the same functions as the members.

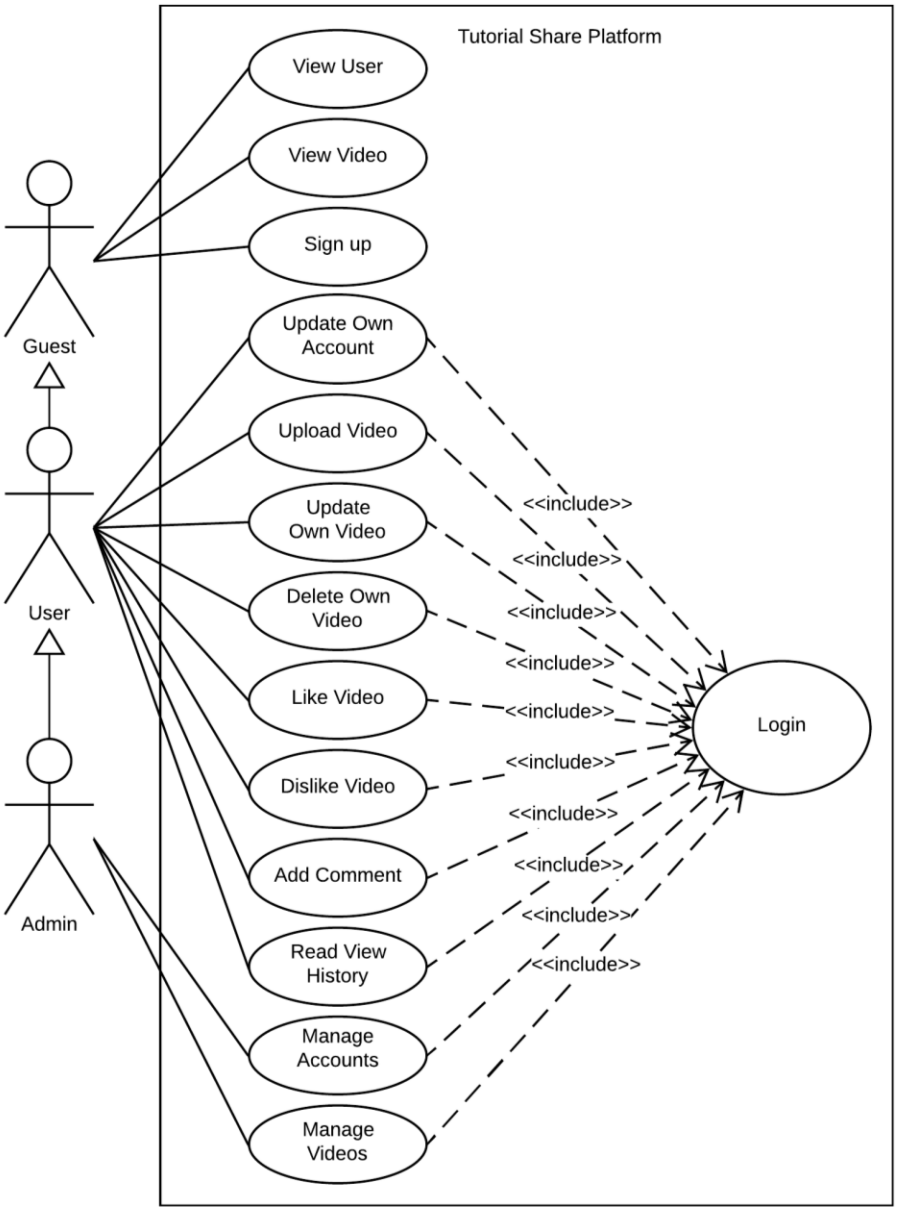

Figure 2: Use Case Diagram for Project Muses

3.0 Structure of Application

3.1 Data Model

The diagram (Figure 3) shows the data model containing the entities used for the database in the tutorial website.

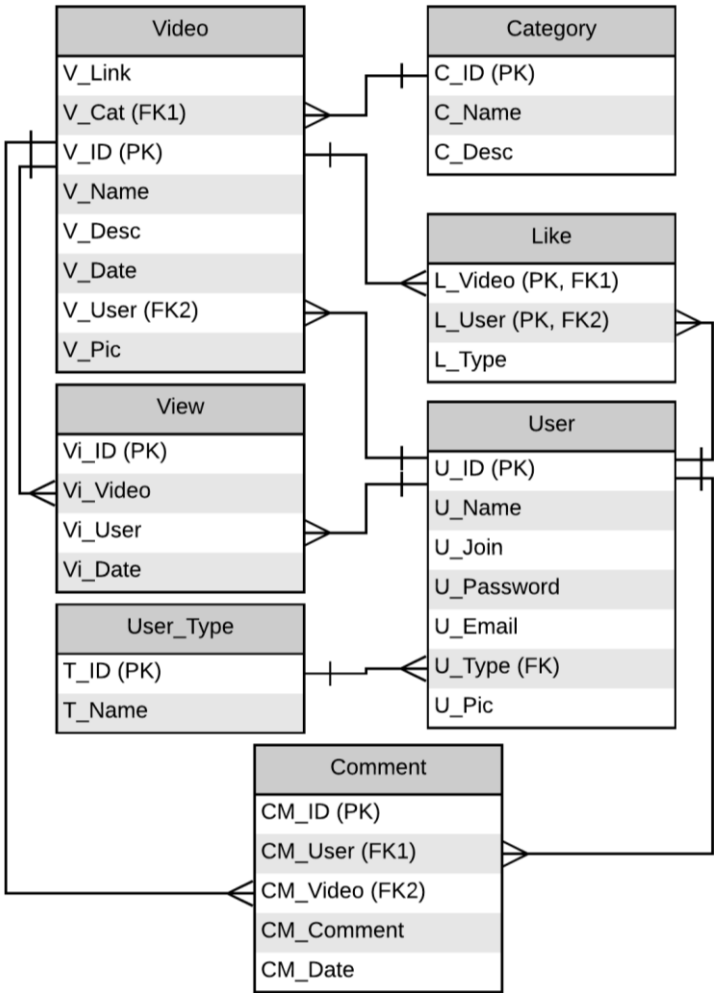

Figure 3: Data Model of Project Muses

3.2 Data Dictionary

The data dictionary shows the details of the attributes in the data model shown above. It contains the data types, field lengths, required fields and the description for the attributes.

Entity	Attribute	Data Type	Field Length	Required	Description
Video	V_Link	Nvarchar	200	Yes	Holds the video file link as a string.
Video	V_Cat	Int	1	Yes	Indicates the category the video belongs to.
Video	V_ID	Int	5	Yes	Unique ID of video.
Video	V_Name	Nvarchar	100	Yes	Name of video.
Video	V_Desc	Nvarchar	500	Yes	Description field of video.
Video	V_Date	Datetime	-	Yes	Date of upload of video.

Video	V_User	Int	1	Yes	Indicates the type of user on the system.
Video	V_Pic	Image	-	No	Holds the binary file containing the thumbnail of the video.
Category	C_ID	Int	1	Yes	Unique ID of category.
Category	C_Name	Nvarchar	50	Yes	Name of category.
Category	C_Desc	Nvarchar	500	Yes	Describes the content for the category.
Like	L_Video	Int	5	Yes	Indicates the video liked.
Like	L_User	Int	5	Yes	Indicates the user liking the video.
Like	L_Type	Int	1	Yes	Indicates whether the like is a like or dislike.
View	V_ID	Int	5	Yes	Unique ID of a View.
View	V_Video	Int	5	Yes	Indicates the video viewed.
View	V_User	Int	5	Yes	Indicates the user viewing the video.
View	V_Date	Datetime	-	Yes	Date of viewing video.
User_Type	T_ID	Int	5	Yes	Unique of user types.
User_Type	T_Name	Nvarchar	20	Yes	Name of user type.
User	U_ID	Int	5	Yes	Unique ID of users.
User	U_Name	Nvarchar	50	Yes	User name of the user. Used for logging in and for display.
User	U_Join	Datetime	-	Yes	The date the user signed up on the platform.
User	U_Password	Nvarchar	50	Yes	Stores the password of the user for login.
User	U_Email	Nvarchar	50	Yes	Stores the email address of the user.
User	U_Type	Int	5	Yes	Indicates the type of user.
User	U_Pic	Image	-	No	Holds the binary file containing the avatar of the user.
Comment	CM_ID	Int	5	Yes	Unique ID of the comment.
Comment	CM_Video	Int	5	Yes	Indicates the video commented.
Comment	CM_User	Int	5	Yes	Indicates the user commenting on video.
Comment	CM_Comment	Nvarchar	500	Yes	Contains the contents of the coment.
Comment	CM_Date	Datetime	-	Yes	Stores the date and time the comment was posted.

3.3 Site Map

This diagram (Figure 4) shows the navigation paths to the various pages of the Project: Muses website.

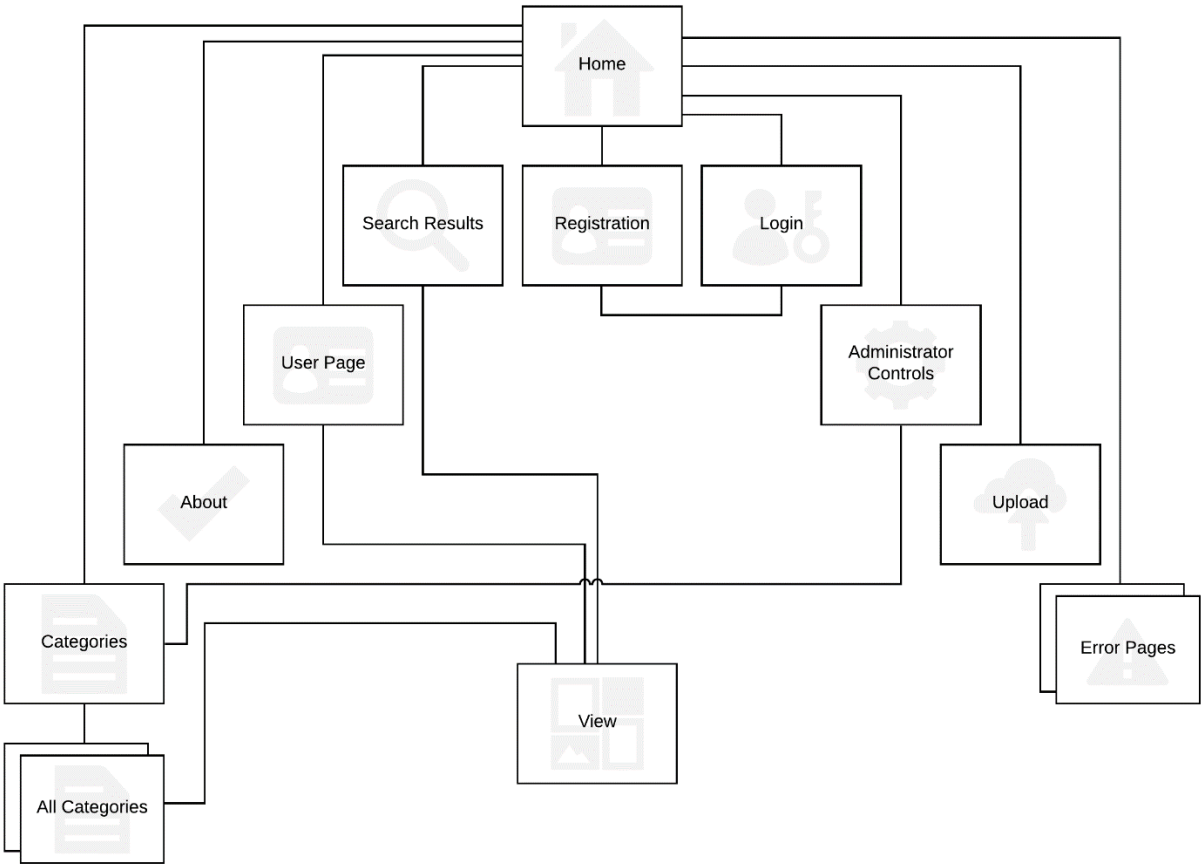

Figure 4: Site Map of Project Muses Application

4.0 Data Processing

Listed below are explanations of queries used to retrieve the data displayed on the webpages for various functions.

4.1 Select All Categories

This displays the Categories page containing all ten categories. The thumbnail, name and description of the individual categories are shown to the user.

4.2 Select All Videos from a Category

This displays the specific Categories page to the user containing videos under the chosen category. The query selects the videos from the Video table by Inner Joining with Category table to get the videos from the chosen category. The output shows the ID, name, description, upload date, category name and the owner of the tutorial.

4.3List New Videos

On the home page under the slider, the user is shown a few recently uploaded tutorials. This is done by selecting the top three videos from the Video table that is sorted by date uploaded. It is intended to bring the most recent tutorials to the viewers and to direct some traffic to all tutorials at one point.

4.4 Create New Tutorial

In the Upload page, the users logged in as Members or Administrators are able to upload videos to the platform. The user enters the name, description, YouTube embed link in the text fields and selects the relevant category the video falls into in the drop-down list.

Once the user clicks submit, the server generates the date and time the video is uploaded, stores the current logged in user ID and the owner of the video as a foreign key linked to the user table. This information is all stored into the Video table to form a complete video record.

4.5 Edit Tutorial

In the profile page of the user, the details of the uploaded tutorial such as its name, description and the category can be updated in the ‘Uploaded Tutorials’ list. This list is created by joining the User and Video tables via the user ID key. The edit is only allowed if the logged in user is the owner of the tutorial or an administrator.

4.6 View Tutorial

The View page is used to show all tutorials in playable video format. The previous page the user was on before clicking on the video link, passes the ID of the video to the view page. The page retrieves the relevant video entry from the Video table and outputs the video itself, the name of the video, description, number of likes and dislikes and the comments the video has received. This is done by using Inner Joins on Video, Like, Comment and User tables on common user and video IDs.

In the Like table, the IDs of the video and the user form composite keys. This means that one user can only like or dislike a video once. A view record is also inserted every time the page is loaded with the IDs of the user and the video to keep records of user video views and to collect the number of views a tutorial has received.

4.7 Delete Tutorial

To delete a tutorial, the user must be logged in as the owner of the tutorial to be deleted or as an administrator. When a delete is executed, tables that contain child records from videos also

perform deletes on records such as like, views and comments. This is due to cascade delete being enabled on the child tables.

4.8 View Member List

The member list is shown on the administrator control panel as a data grid. It retrieves all columns from the User table such as ID, avatar, name, join date, email, user type and last logged in date.

4.9 Register New Member

This is done on the Registration page of the website. The user must enter the username, password, password confirmation and their email address. When the user clicks submit, the ID and the join date of the user are generated and entered along with the rest of the information.

The database also checks whether the username or the email address already exists in the database before entering the record. If any already exists, the user is prompted to enter different username or email address.

4.10 View Member Details

The details of the members can be seen as the owner of the profile or as another user. The user profile page shows the attributes of the user and the tutorials they have uploaded. If the user is the owner of the profile, they would also be able to edit the user or tutorial details on the page.

4.11 Edit Member Details

The member information can be edited on the profile page as the owner of the profile or as an administrator. For the members, they are only able to edit the email address and the password. The administrator is able to edit all fields including the ID and the username.

4.12 Delete Member

The owner of the profile or the administrator can delete a member account from the website. When the delete is executed, the child records such as associated tutorials, likes, comments and view records are cascade deleted from the database.

4.13 Like Tutorial

From the view page, the members can add a like or a dislike to a tutorial to indicate to other prospective viewers of the tutorial of the quality of the tutorial. This action adds a record into the like table including the video ID and the user ID as the primary key to make the like unique and a like type to indicate whether it is a like or a dislike.

4.14 Add Comment

A comment can be added from the view page when the user is logged in as member or administrator. This adds a comment with Video and User linked by foreign keys to indicate who commented on which video.

4.15 Edit Comment

Existing comments can be edited as the owner of the comment or as administrator from the video the comment was posted to. This updates the CM_Comment attribute.

4.16 Delete Comment

Existing comments can be deleted as the owner of the comment or as administrator from the video the comment was posted to. This deletes the associated comment record.

Prototype User Interface

Mood Board

The general idea of the design of Project: Muses is to invoke a visual feeling of homeliness and comfort while being slightly bubbly to be fun and inspiring. Inspiration for the colours is taken from pictures mostly depicting modern interior design to emulate the clean, modern look on the website.

Figure 5: Mood Board Pictures for Design Inspiration

Design Specifications:

With the inspiration derived from the mood board, the following table describes the details for the design of Project: Muses.

No.	Attribute	Values
1.	Font Name	Space Mono
2.	Header Colour	#1b87c1 (Strong Blue)
3.	Header Font Size	23px
4.	Header Font Colour	#FFFFFF (White)
5.	Sidebar Colour	#e68639 (Bright Orange)
6.	Sidebar Link Font Size	12px
7.	Sidebar Font Colour	#FFFFFF (White)
8.	Content Font Size	18px
9.	Content Font Colour	#000000 (Black), #284775 (Dark Moderate Blue)
10.	Overall Background Colour	#f3f1ec (Light Greyish Orange)
11.	Hover Accent Colour	#a52a2a (Dark Red)

All colour values and names derived via: www.colorhexa.com

Navigation

The users of the application will be able to use various forms of links to navigate through the website. These hyperlinks can be embedded in many visual elements such as text (Figure 6), buttons (Figure 7) and images (Figure 8) located throughout the application such as on the main page content, top menu bar, the sidebar on the desktop or the dropdown menu for the mobile devices.

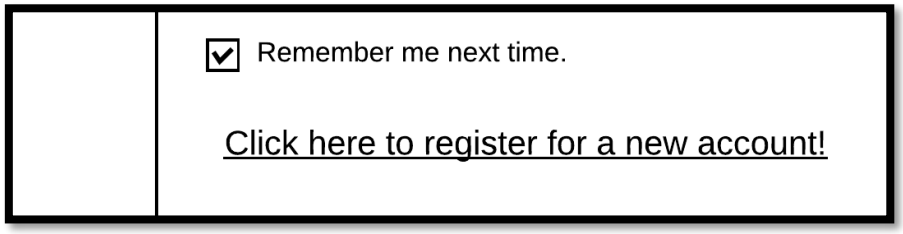

Figure 6: An Example of Text Links

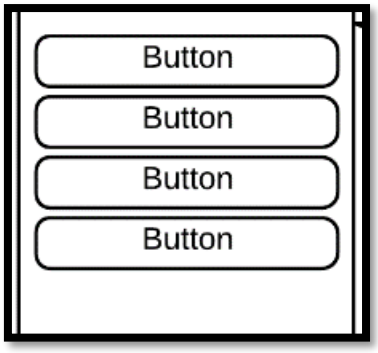

Figure 7: An Example of Button Links

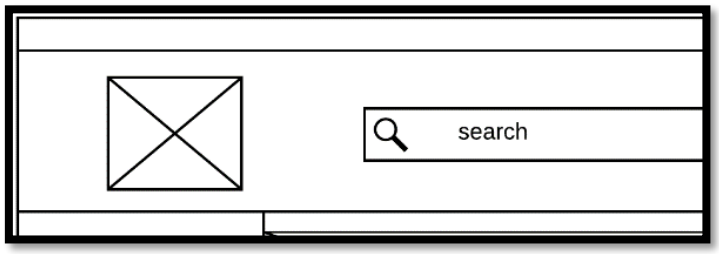

Figure 8: An Example of an Image Link

Various C# method is also used in navigation as part of scripted actions such as directing the user to the personalised home page after a successful log-in or after a navigation error.

User Interface Mock-ups

Master Page Template

The master page template is used to form the basics of all web pages within the website. This master page provides the top and side navigation bars to hold the links to various pages of the website. The empty section is allocated to the content placeholder to allow pages to add their interface to the template (Figure 9)Figure 9: Master Page Template of Project: Muses.

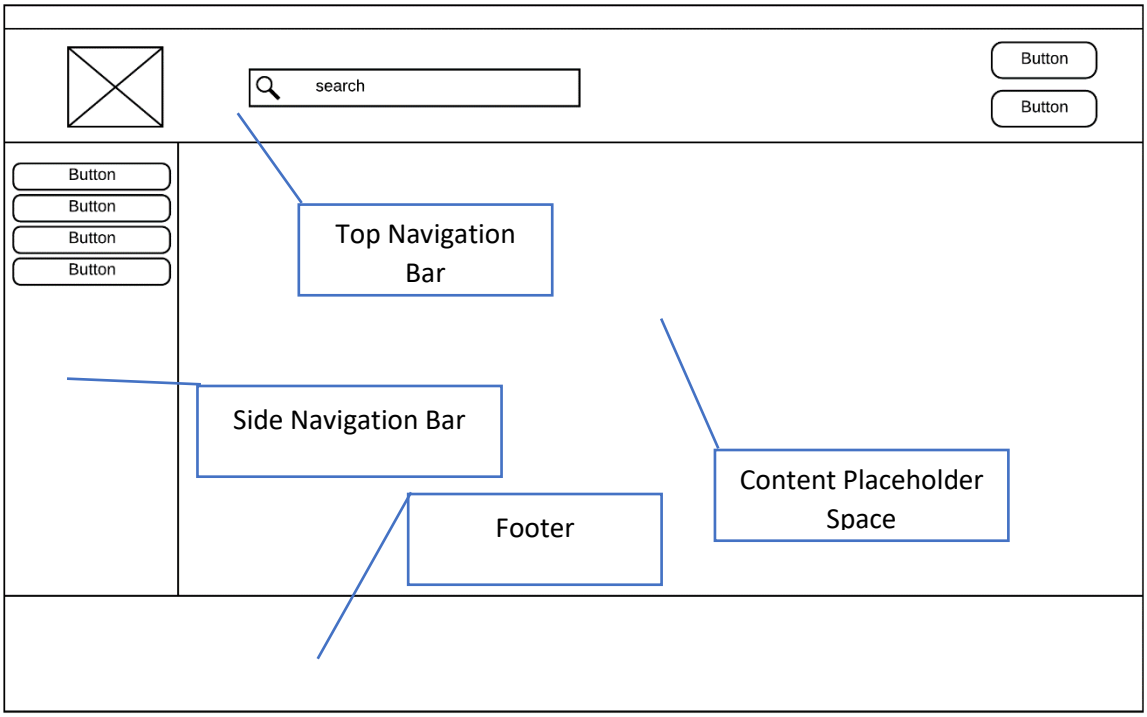

Figure 9: Master Page Template of Project: Muses

Home Page

This page serves as the landing page for all visitors of the website and redirections. It contains a jQuery slider to promote content on the website and a few highlights of recently uploaded tutorials below it (Figure 10).

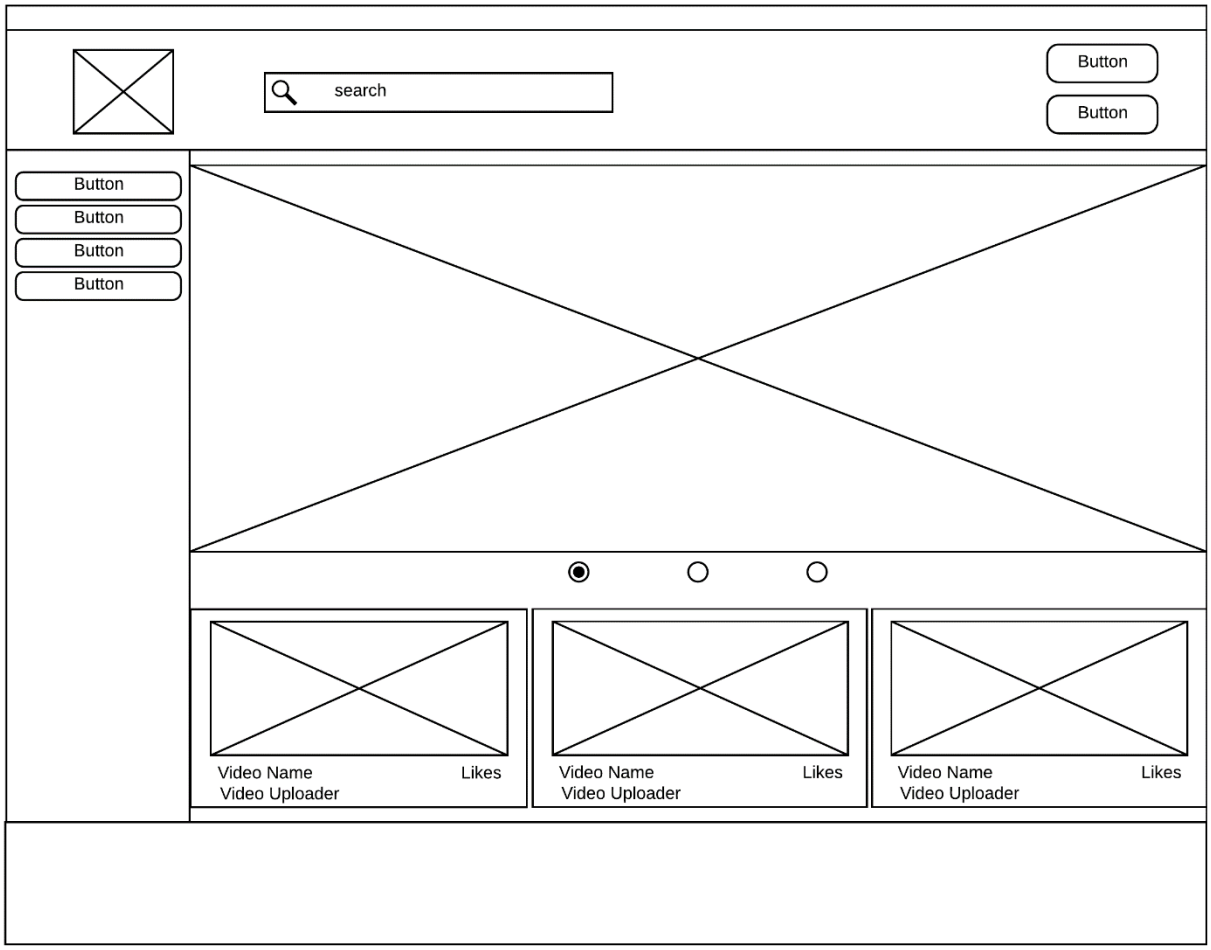

Figure 10: Home Page of Project: Muses

Login Page

This page provides the interface for users of the website to login with their credentials. The login interface provides the text fields to enter username and password, a checkbox for remembering the users' login credentials and a link for those who wish to register a new account (Figure 11).

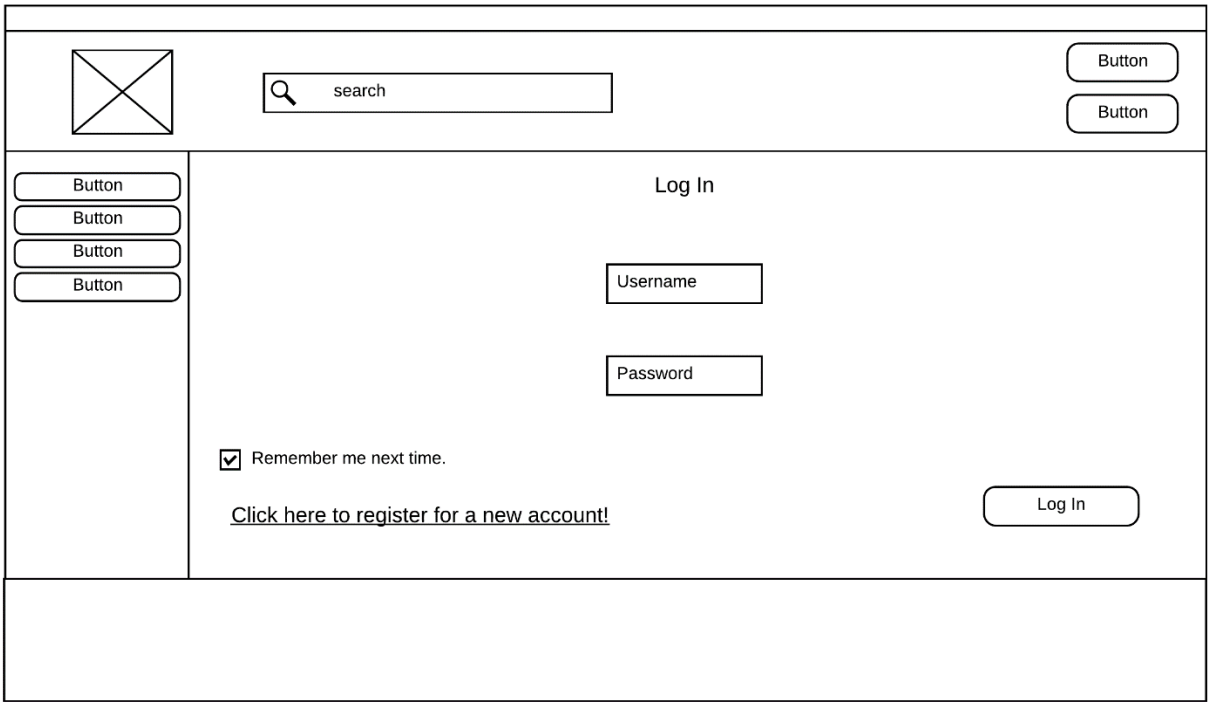

Figure 11: Login Page of Project: Muses

Registration Build

This page provides the interface for guests of the website to register for their own accounts. The registration interface provides text fields to enter username, password and the email address (Figure 12).

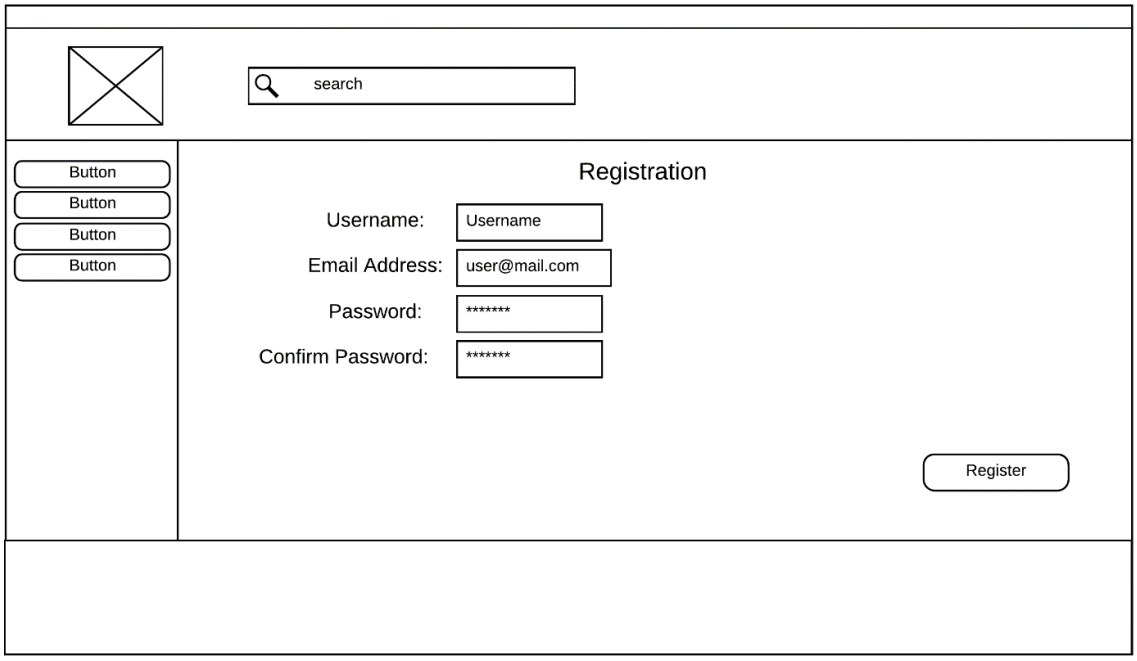

Figure 12: Registration Page of Project: Muses

Search Results

This is the page which returns the results containing tutorials when a user uses the search field on the top navigation bar. The page contains a data grid containing the name, description, upload date, category and the number of likes on a video (Figure 13).

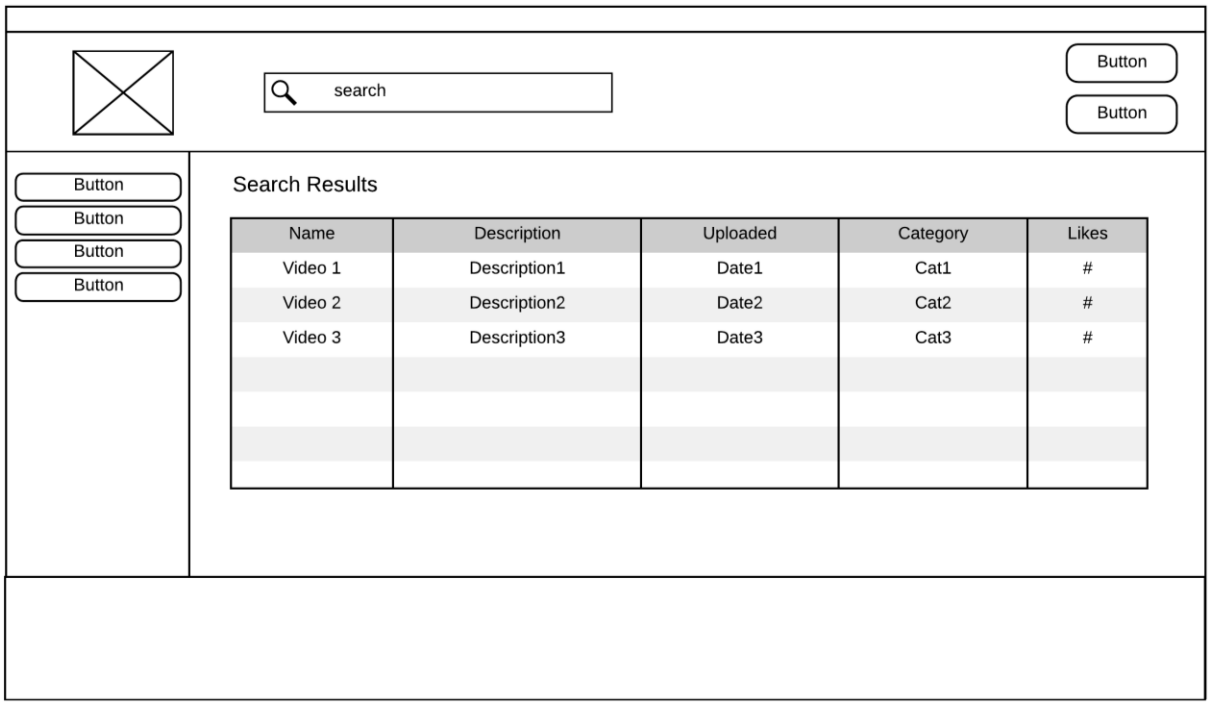

Figure 13: Search Results Page of Project: Muses

Tutorial Categories Page

This page shows all available categories on Project: Muses website. The categories are represented by images representing their topics (Figure 14). Clicking on the images brings the users to the video list page of respective categories.

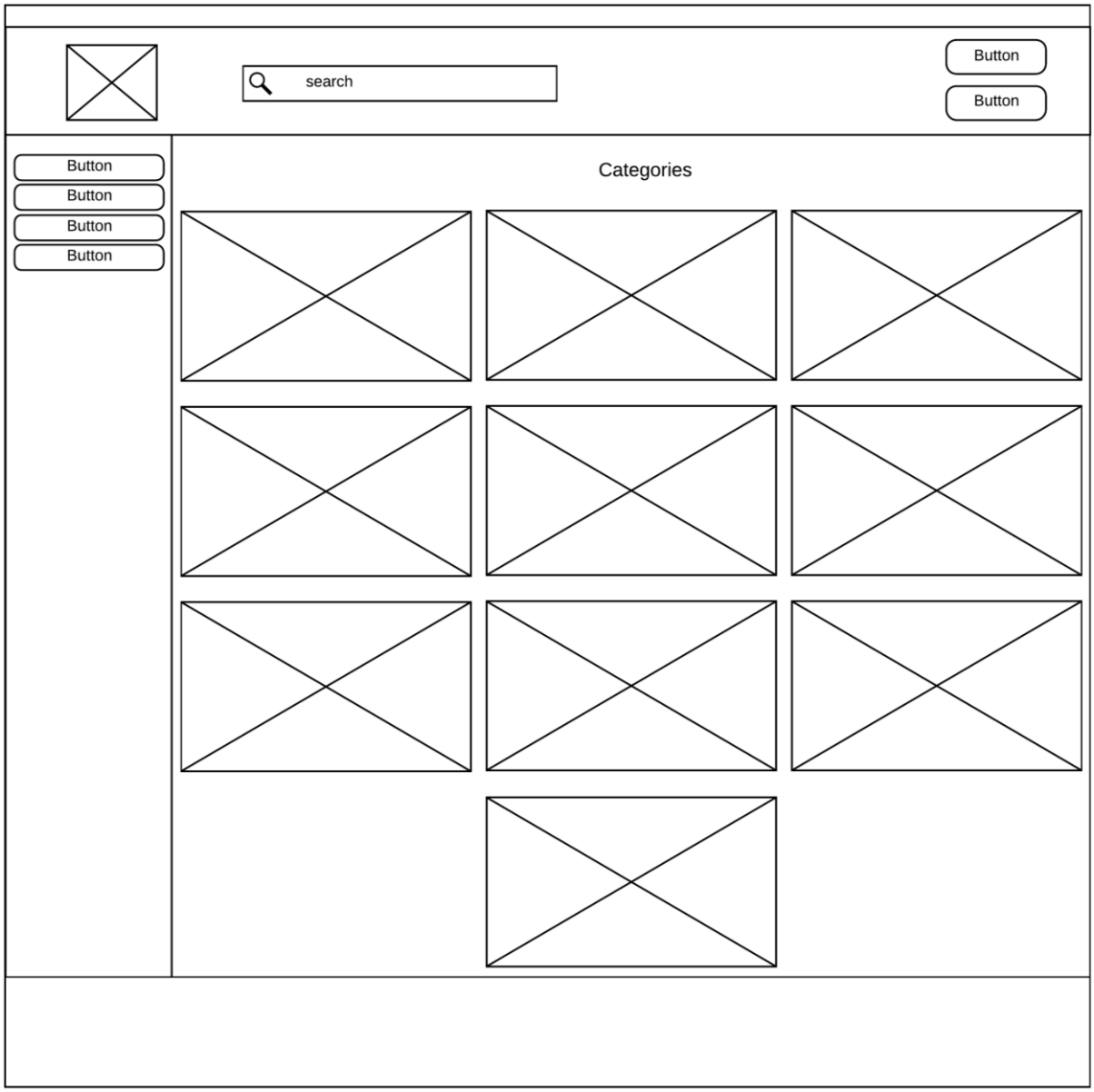

Figure 14: Categories Page of Project: Muses

Video List Page

This page uses a list view to show all videos under a selected category. It shows the videos' information and their thumbnails (Figure 15).

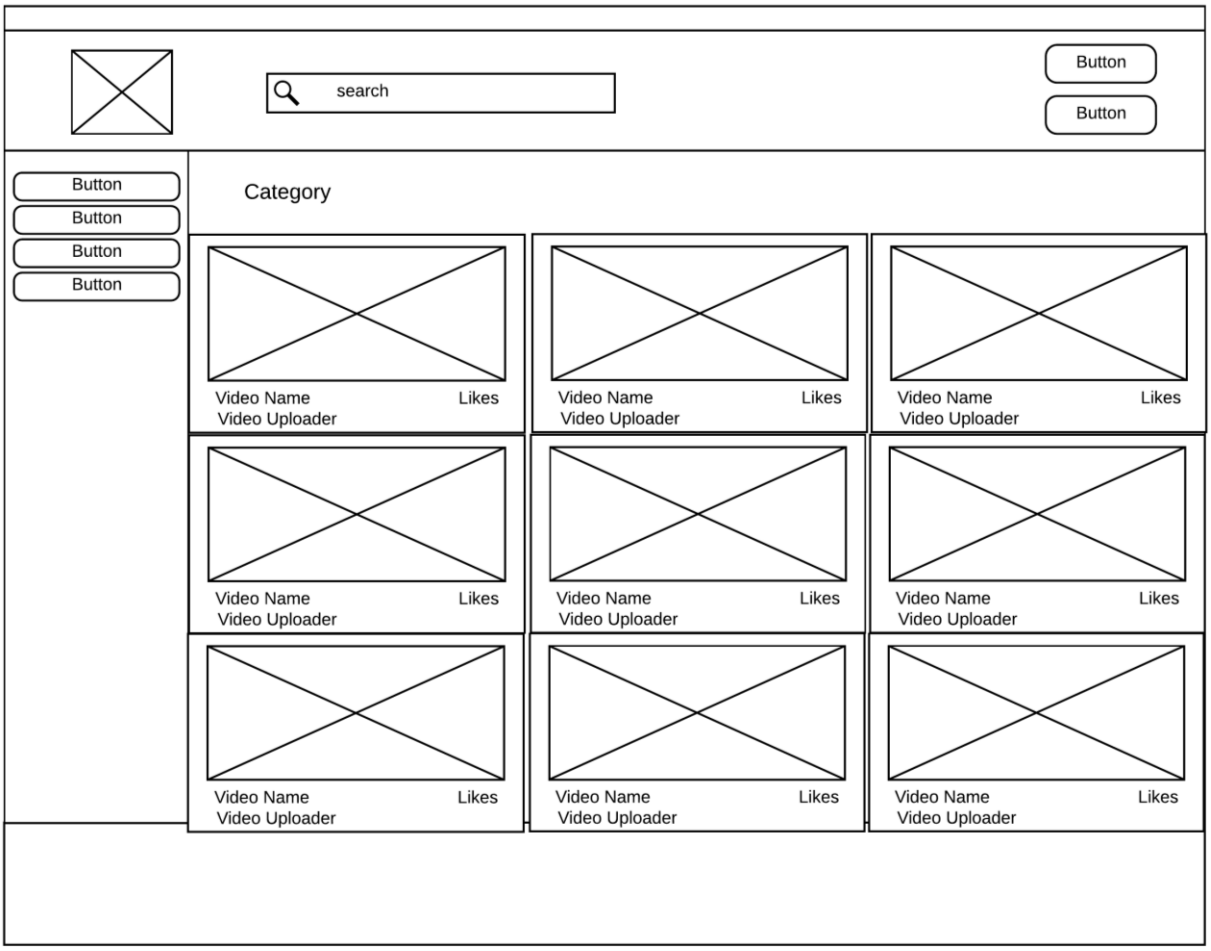

Figure 15: Video List Page for Showing Categories

User Profile Page

This page is used to show a member's information and the tutorials they have uploaded. If the user is the owner of the profile, they would also have the permission to edit the details on the profile (Figure 16).

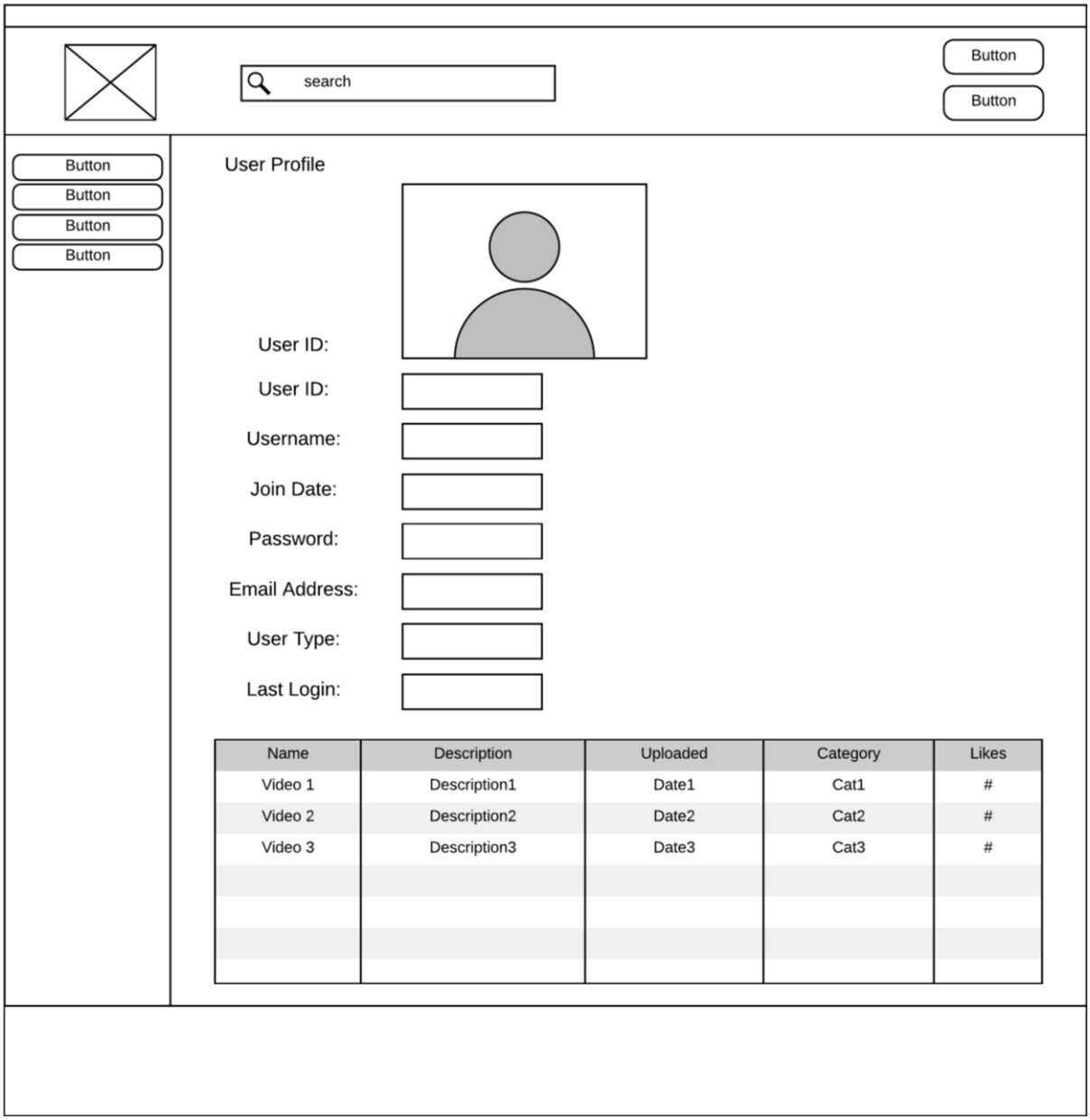

Figure 16: User Profile Page of Project: Muses

View Tutorial Page

This is the general page used to view all videos in a tutorial. It contains an embedded YouTube video player, the description of the tutorial, number of likes and dislikes and the comments submitted by other users (Figure 17).

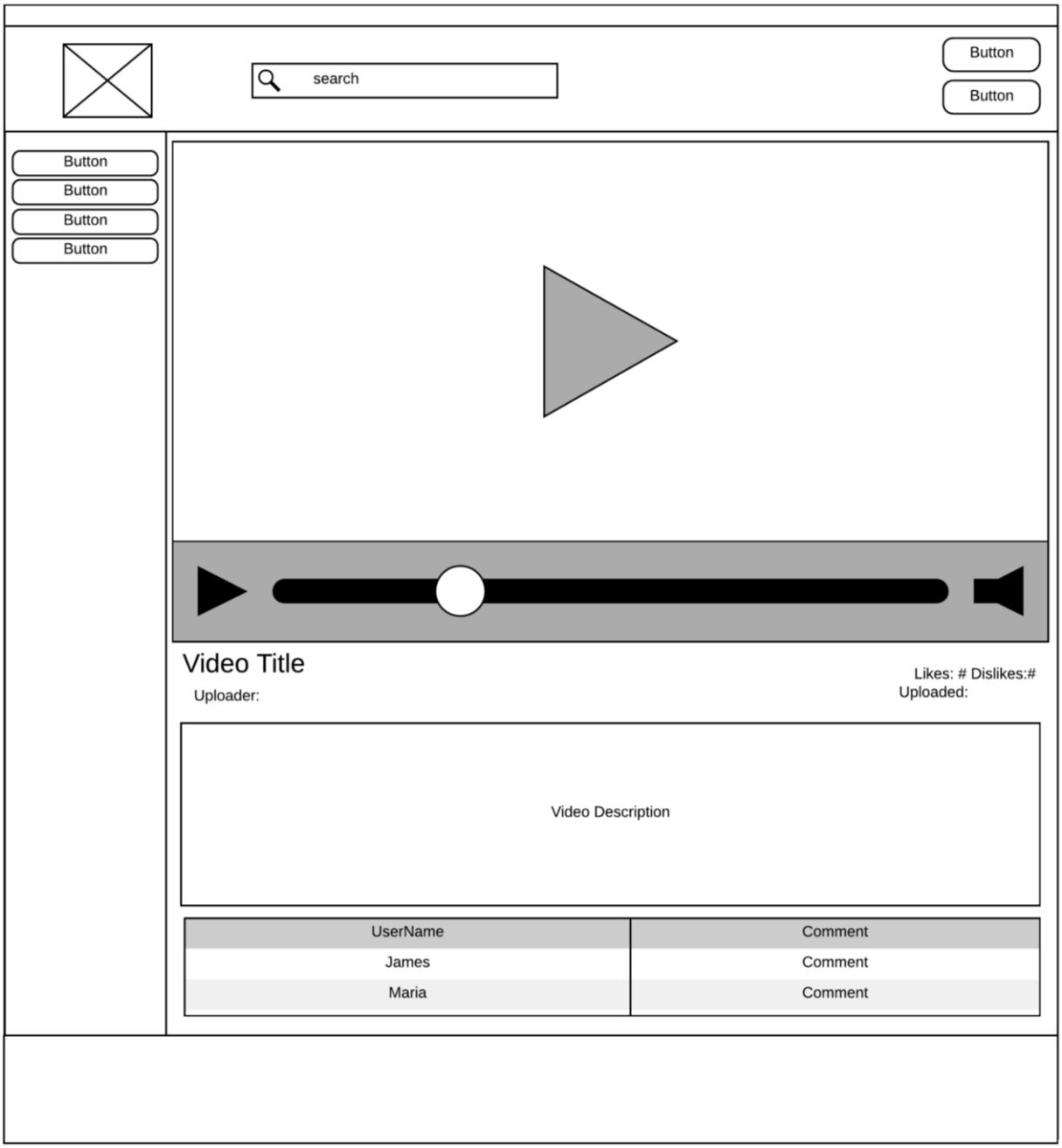

Figure 17: View Page of Project: Muses

Video Upload Page

This page provides the users with facilities for uploading a tutorial onto their profile. It provides fields to enter information about the video and its embed link (Figure 18).

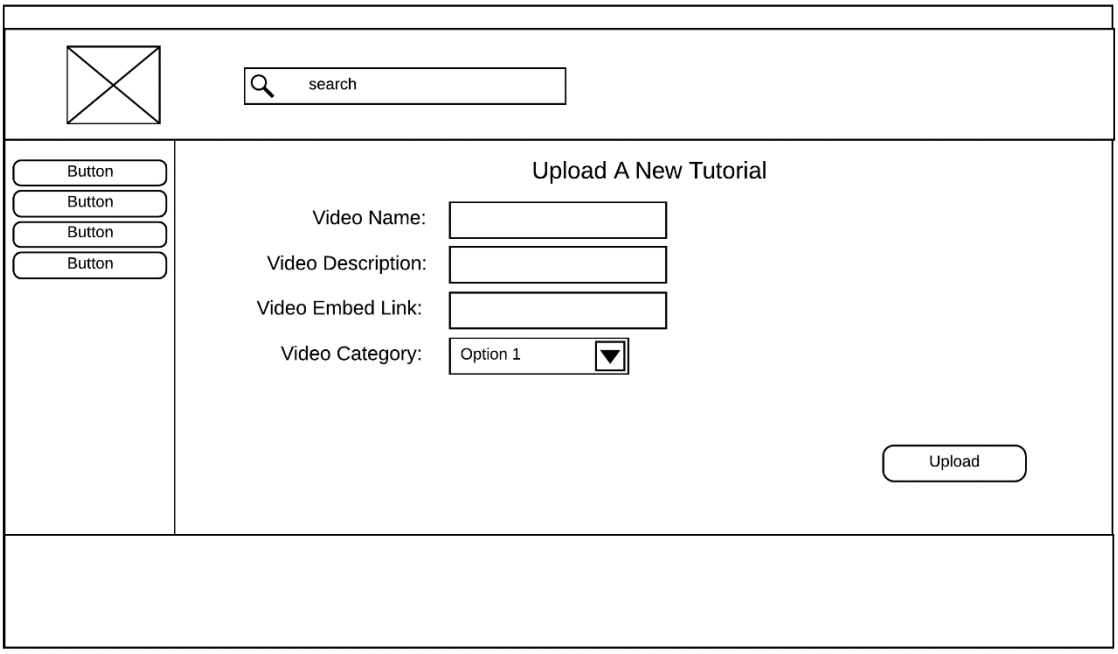

Figure 18: Upload Video Page of Project: Muses

About Page

The about page briefly explains the background and purpose of the Project: Muses website. It also provides links for contacting the developers for feedback (Figure 19).

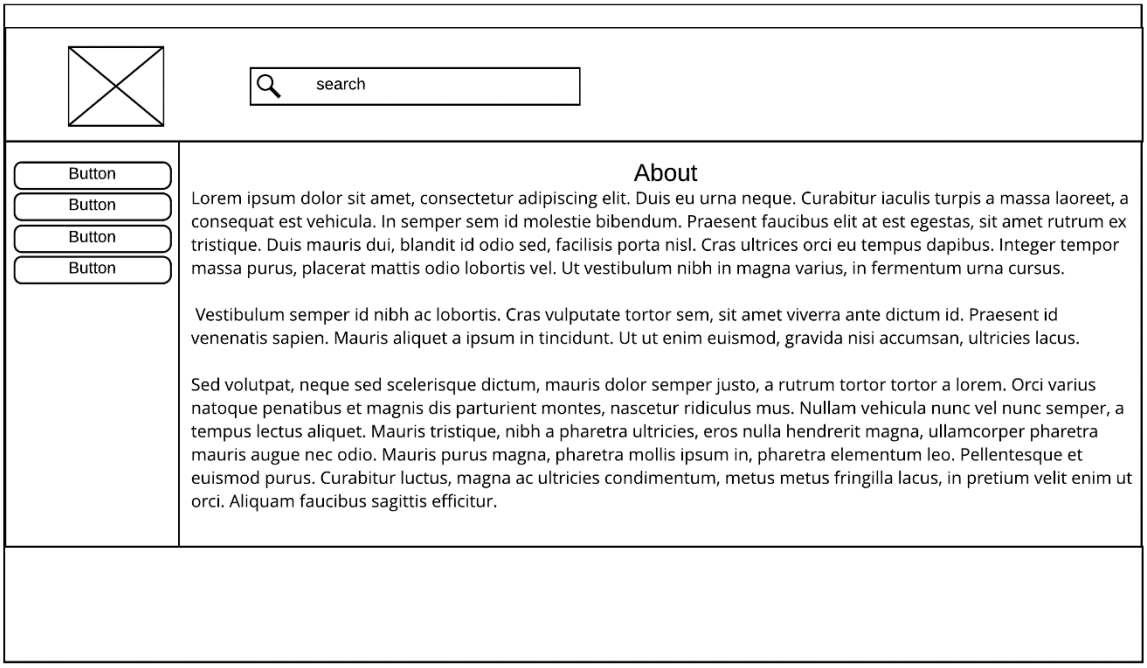

Figure 19: About Page of Project: Muses

Error Page

This page is shown when the application has encountered an error such as the user entering an invalid URL (Figure 20).

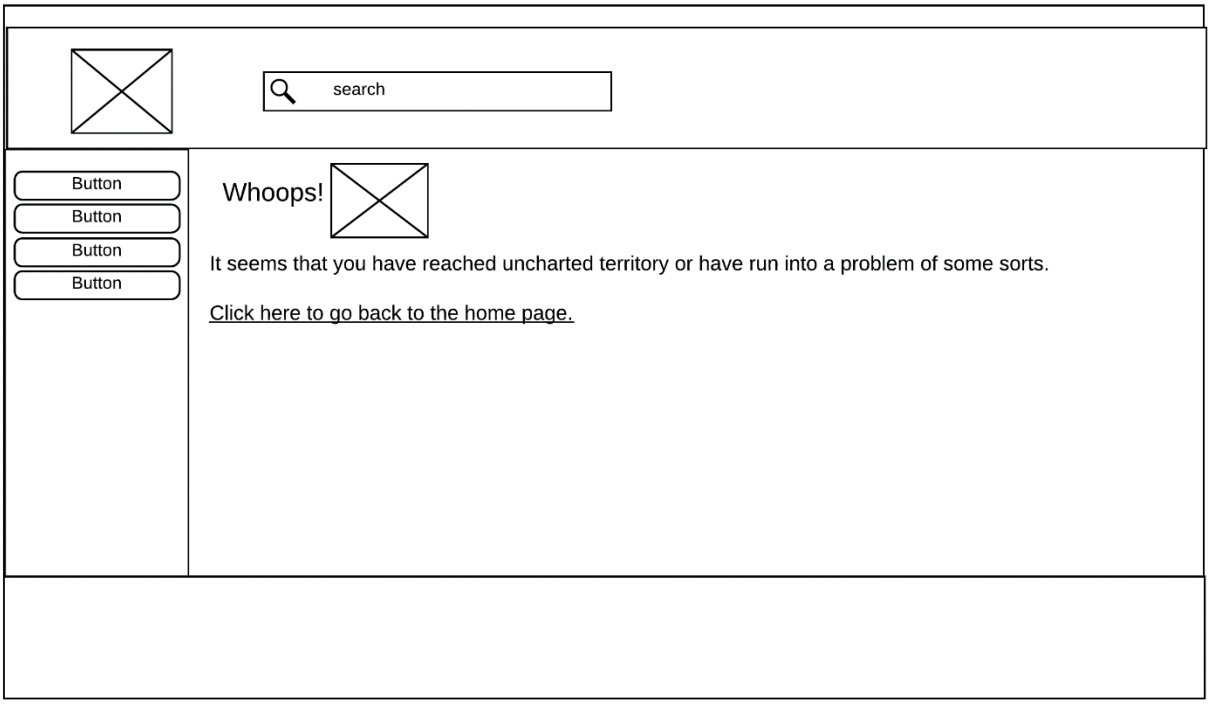

Figure 20: An Error Page for Project: Muses

Security Policies

The following matrix describes the permissions for page access by various user roles (Figure 21).

Page/Users	Guest	Member	Administrator
Home	☑	☑	☑
About	☑	☑	☑
View	☑	☑	☑
Categories	☑	☑	☑
Category Video List	☑	☑	☑
Search Results	☑	☑	☑
Login	☑	☑	☑
Registration	☑	☑	☑
Upload	☒	☑	☑
Error Pages	☑	☑	☑
User Profile	☑	☑	☑
Administrator Controls	☒	☒	☑

Figure 21: Access Matrix Showing Page Access Permissions

The following matrix describes the permissions for the entities in the database by the users (Figure 22).

Entity/Users	Guest	Member	Administrator
User	Create, View	View, Update	Create, View, Update, Delete
Video	View	Create, View, Update, Delete (Own entries only)	Create, View, Update, Delete
View	Create, View	Create, View	Create, View, Update, Delete
Like	View	Create, View	Create, View, Update, Delete
Comment	View	Create, View	Create, View, Update, Delete
User_Type	View	View	Create, View, Update, Delete
Category	View	View	Create, View, Update, Delete

Figure 22: Access Matrix for Entities in Project: Muses

Listed below are the policies associated with functions written into the system to be enforced to ensure security of the application and its users, sorted by category of the functions in the application.

Login

- Username and Password must match.
- Username must not contain special characters or spaces.
- Password must be 6 characters minimum in length and must contain a number.

- After 3 login attempts, message to contact administrator appears.

Creating Users

- User ID must not exist.
- Username must not exist.
- Registration email must not exist.
- Username must not contain special characters or spaces.
- Password must be 6 characters minimum in length and must contain a number.

Access

- User must be logged in to 'like'.
- User must be logged in to upload tutorials.
- User must be logged in to manage videos.
- User must be logged in as an administrator to use administrative functions.

Editing

- User must be the owner of the tutorial to be able to make changes to it.
- User must be the owner of the tutorial to be able to remove it from the platform.

Technology

Various technologies will be used to develop this application. Microsoft ASP.Net is used to create a web application that can dynamically generate pages from the server. This uses C# programming language to write the necessary code required to perform various functions on server-side, such as validation of input, inserting and retrieving data from the server database, etc (Microsoft, 2018).

SQL Server database is used to store and manage data for the application. LINQ to SQL Database technology is used to create the link between the web application and the database. This allows for the website to be updated dynamically with the data from the database (Microsoft, 2017).

The tutorial web application is developed using Microsoft Visual Studio 2017. It allows for creation of web applications rapidly. This is due to the extensive automatic code generation tools and the drag-and-drop interface that Visual Studio provides, which makes making the application very simple and quick (Microsoft, 2018).

While these technologies make developing a web application quick and relatively simple, I found that they also have some drawbacks. Despite the code generation, much of the code still requires being written manually such as including libraries and such. This requires extensive knowledge of all functions included in every pre-set code library and makes the program less beginner friendly. The Visual Studio environment is also not completely bug-free. Much time had to be spent interpreting and finding the sources of error messages or reinstalling modules to ensure that they loaded in the interface itself. While this could be due to my lack of extensive experience with Visual Studio and could be remedied with more familiarity of the environment, there are other development tools such as Mendix which are less buggy in my opinion.

References

Microsoft, 2017. *Code generation features in Visual Studio.* [Online]
Available at: https://docs.microsoft.com/en-us/visualstudio/ide/code-generation-in-visual-studio?view=vs-2017
[Accessed 13 08 2018].

Microsoft, 2017. *LINQ to SQL.* [Online]
Available at: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/sql/linq/
[Accessed 11 08 2018].

Microsoft, 2018. *ASP.NET Overview.* [Online]
Available at: https://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx
[Accessed 12 08 2018].