|  |
| --- |
| Responsive-web icon |
| Web Portfolio for  Jamie Rodden |
| |  |  |  | | --- | --- | --- | | Arnis Zelcs st. n. 20136310 | 2/20/16 | HND Computing - Software Development | |

Content:

[**Introduction** 2](#_Toc450343925)

[**Client Requirements** 2](#_Toc450343926)

[**Project Requirements** 3](#_Toc450343927)

[**Functional** 3](#_Toc450343928)

[**Non-Functional** 3](#_Toc450343929)

[**Research** 3](#_Toc450343930)

[**Tools and Services for Development** 4](#_Toc450343931)

[**Information Resources and Help** 5](#_Toc450343932)

[**Use Case Diagram** 6](#_Toc450343933)

[**Data Structure Normalization** 7](#_Toc450343934)

[**Tables and Relations** 5](#_Toc450343935)

[**Design** 1](#_Toc450343936)

**Web Portfolio** *for* **Jamie Rodden**

# **Introduction**

Every professional creative person needs more than just a plain text CV for demonstration of his or her talent, skills and knowledge. Nowadays, it is very common to have a web portfolio website where you can clearly demonstrate your work and achievements. Jamie Rodden is a young professional artist and musician living in Aberdeen. Luckily, I had a chance to meet this man at O’Neill’s pub where we are both working just now. I was looking for an idea for my Graded Unit project and Jamie was very interested in a chance of making a web portfolio website for him. His idea was to place some of his work like showreel video, pictures and music on a website, but making just a website wouldn’t cover my Graded Unit project specifications and I decided to make a portfolio website with a control management system. My idea is to make an administration rights to Jamie and make a control panel where he would be able to change content of his website himself without any knowledge of programming.

# **Client Requirements**

Jamie made a brief scratch of a website consisting from a landing page (home page) and another four pages for splitting his content in different groups.

Landing/Home page

This page should consist of a horizontal smooth sliding gallery what should be controlled by the picture menu *(look design section for visual representation)* and main menu for navigation to other pages.

Showreel page

This page should consist of showreel video with a title and description.

Actor CV page

This page should contain an actor’s CV.

Media page

This page should consist of shots from gigs/productions.

Music page

This page should contain the songs (audio files).

# **Project Requirements**

## **Functional**

Browse content on a website

Login as administrator

Providing rights for changing content of a website for administrator

Ability to view on a server

Ability to add new content on a website

Ability to edit content on a website

Ability to delete content on a website

Ability to upload files on a server

Ability to delete files from a server

## **Non-Functional**

Responsive design

Modern, clear and friendly UI

Clear and friendly Administration Panel

Website performance (work with DOM elements)

Sensitive data encryption and decryption

File upload to the server with allowed set of extensions

File deletion on the server

# **Research**

To get some ideas about how a website may look like I have analysed some web portfolios and websites of different people. I would like to mention one very good looking and professional website of one creative photographer - [Graeme Macdonald](http://graemephotography.com/) (<http://graemephotography.com/>). That photographer has a very impressive website and there are many ideas I potentially can use in my project. The only thing I didn’t like about that website is its loading time as there are used a lot of large graphics and animations. It is making a feeling what the website is overloaded and not efficient. I made a decision to use less images of a big size and animations. As well I decided to consider the best practices of making good and balanced products.

Almost every project can be developed in many ways and with use of different tools. After all requirements were defined it was important to make a research and choose appropriate tools for an implementation stage. I read a lot of articles and forum discussions about advantages and disadvantages of different tools to figure out the most appropriate and suitable ones for the current project.

There was necessary to choose a server side language for the project. My choice stopped on PHP for several reasons. Comparing with other alternatives like Python and ASP.NET I noticed what the PHP is more suitable for the project and one of the main reasons was its popularity. PHP has greater community, lots of books and tutorials as it is free and well supported. Another reason is a choice of hosting for the project as PHP hosting is possible to find almost everywhere when Python hosting is harder to find and comparing to Python or ASP.NET, PHP hosting is less-expensive. One more reason for choosing PHP was a development on a local machine as I found easier instructions of how to download and setup WAMP (for windows) server what consists of the Apache web server PHP and MySQL comparing to complex setups of servers for Python and Windows servers. MySQL was also ideal choice for me as I am familiar with it already and I always wanted to get experience working with it.

It was made a decision to choose a HTML, CSS and JS framework for a front-end development of the project. Most of that frameworks are giving an option to create responsive, mobile first web projects easier and faster. Bootstrap is the most popular framework of that category and would be ideal for many kinds of projects, but I made a decision to try another framework as I am already familiar with the Bootstrap. I was impressed with the material design language created and introduced by Google and I really wanted to try building a projects using principles of material design. One of my friends suggested to have a look on Materialize framework and I was really surprised with that framework and what it is offering to developers. I made a personal choice to use Materialize framework for my projects even it is not as popular as Bootstrap, Foundations, MDL (official by Google) or any other framework of that category.

It was made a decision to use GitHub platform for keeping truck of the development process and backups. That wasn’t a difficult choice as GitHub is the largest host of source code in the world. GitHub provides desktop and web-based graphical interface what gives an option to save every step of development, create another versions of the prototype, compare and retrieve the code of every commit.

Some research on the MySQL GUI tools showed me what MySQL Workbench visual database design and development environment is one of the best and widely used tools among the developers. Other similar tools like Navicat for MySQL, HeidiSQL or SQLyog are just an alternatives and preferences of other developers as that tools are quite similar on what they are offering. I decided to use MySQL Workbench during implementation on the local machine. For the purpose of testing and demonstration of the project on a web I decided to use my hosting space on Eco Web hosting, that hosting provider is offering phpMyAdmin browser-based MySQL GUI tool, so I it will be used during the development stage of the project.

## **Tools and Services for Development**

All tools I am going to use are free and can be downloaded on the official websites. GitHub is also providing 5 private projects and unlimited public projects while you are student.

Notepad++ (code editor)

GitHub (Git repository hosting service) – save, backup, create separate branches

WAMP server (includes Apache web server, PHP and MySQL) - local

MySQL Workbench (MySQL scripting editor) – local

phpMyAdmin (MySQL scripting editor) – hosting

A dynamic and data driven website is required for this project. The first task is to make a modern (materialize) and responsive design as any creative person’s web portfolio have to leave a good impression on its visitors.

From the technical side there is necessary to have knowledge of HTML, CSS and JavaScript for the user side implementation. To make implementation faster it was decided to use Materialize framework and jQuery library what should simplify work with website responsiveness, styling of elements, DOM manipulations and will allow to use some of the integrated functionality of the framework. The second task is to make a class based website with dynamic content display fetched from a database. I chose a PHP language and MySQL database for the server side implementation.

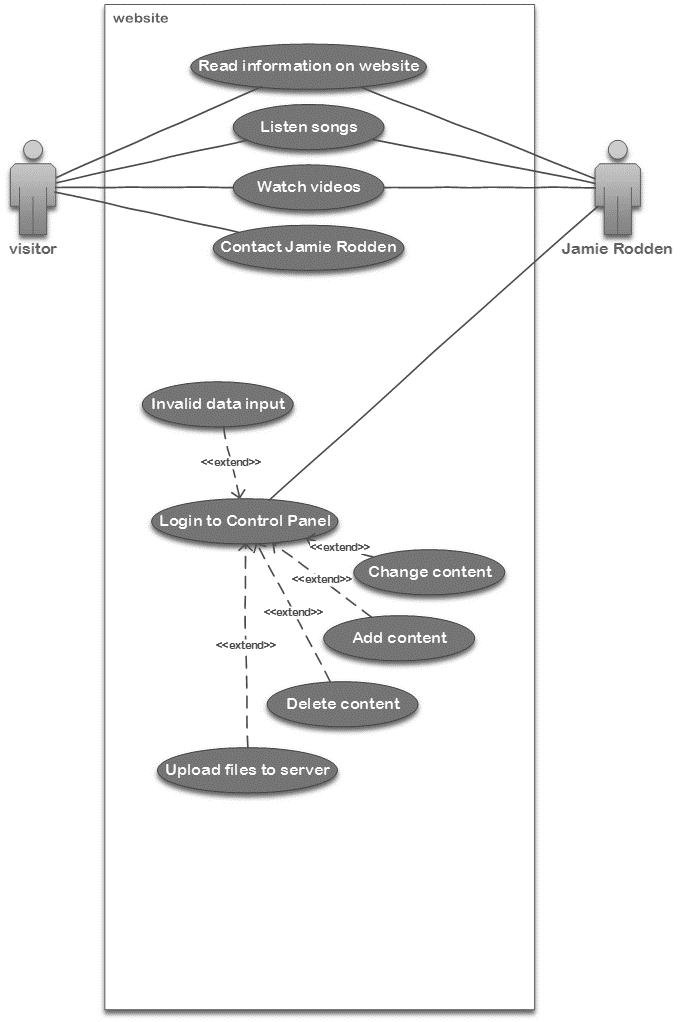
As addition, I decided to write a horizontal sliding gallery jQuery plugin for demonstration of my ability to write my own plugins and make a few nice animations in JavaScript. As well I decided to write my own ORM (Object-relational mapping) tool, based on PHP language and PHP MySQLi driver, with a role to simplify database manipulations.

## **Information Resources and Help**

During implementation I will face with some of the problems and will need to use external information resources.

One of the best resources is a *w3school.com website*, where it is possible to find a solution to a problem related to HTML, CSS and other languages and techniques. For more serious and not standard problems I am going to use a *stackoverflow.com* website as it is a huge resource where professionals may answer on you questions, find a bug in your code and give an advice. Of course, it is more reliable to read an information on official websites, as there is displayed the latest documentation. I am going to read about the PHP functionality on their official website as many functions may be deprecated and replaced with improved ones with a new version of language release. As well I will read a great “PHP and MySQL Web Development” book written by Luke Welling and Laura Thomson. During the implementation it is very important to follow recommendations of other developers, as inappropriate use of some of the elements and techniques, may lead to the slow performance. Especially it is very important for the large scale projects. I am going to rely on good performance practices, like proper use of iterations, code reusability and appropriate use of selectors in jQuery (to reduce DOM tree manipulations), there is a performance section on jQuery official website. For achieving a good and modern design I am going to use Materialize framework as it is quite similar to Bootstrap I am familiar with already. All the information related to this framework I can find on their official website. Also I am going to use my knowledge about MySQL I got last year in college.

# **Use Case Diagram**



# **Data Structure**

## **Normalization**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| usrID | usrID | usrID | usrID | **user** |
| usrName | usrName | usrName | cv\_ID |  |
| usrPsswd | usrPsswd | usrPsswd | usrName |  |
| cv\_ID | cv\_ID | cv\_ID | usrPsswd |  |
| name | cv\_name | cv\_name |  |  |
| equity | cv\_equity | cv\_equity | cv\_ID | **cv** |
| phone | cv\_phone | cv\_phone | cv\_name |  |
| email | cv\_email | cv\_email | cv\_equity |  |
| height | cv\_accents | cv\_accents | cv\_phone |  |
| chest | cv\_skills | cv\_skills | cv\_email |  |
| waist | cv\_height | cv\_height | cv\_accents |  |
| Inside\_leg | cv\_chest | cv\_chest | cv\_skills |  |
| eyes | cv\_waist | cv\_waist | cv\_height |  |
| hair | cv\_Inside\_leg | cv\_Inside\_leg | cv\_chest |  |
| build | cv\_eyes | cv\_eyes | cv\_waist |  |
| playing\_age | cv\_hair | cv\_hair | cv\_Inside\_leg |  |
| training\_id | cv\_build | cv\_build | cv\_eyes |  |
| training | cv\_playing\_age | cv\_playing\_age | cv\_hair |  |
| accents |  |  | cv\_build |  |
| skills | usrID | usrID | cv\_playing\_age |  |
| film\_tv\_id | music\_ID | music\_ID |  |  |
| film\_year | music\_alloc | video\_ID | usrID | **media** |
| film\_role | music\_title | img\_ID | music\_ID |  |
| film\_production | music\_path |  | video\_ID |  |
| film\_director | music\_descr | music\_ID | img\_ID |  |
| film\_company | video\_ID | music\_alloc |  |  |
| theatre\_id | video\_alloc | music\_title | music\_ID | **music** |
| th\_year | video\_title | music\_path | music\_alloc |  |
| th\_role | video\_path | music\_descr | music\_title |  |
| th\_production | video\_descr |  | music\_path |  |
| th\_director | img\_ID | video\_ID | music\_descr |  |
| th\_company | img\_alloc | video\_alloc |  |  |
| music\_ID | img\_path | video\_title | video\_ID | **video** |
| music\_alloc | training\_id | video\_path | video\_alloc |  |
| music\_title | training | video\_descr | video\_title |  |
| music\_path | film\_tv\_id |  | video\_path |  |
| music\_descr | film\_year | img\_ID | video\_descr |  |
| video\_ID | film\_role | img\_alloc |  |  |
| video\_alloc | film\_production | img\_path | img\_ID | **images** |
| video\_title | film\_director |  | img\_alloc |  |
| video\_path | film\_company | cv\_ID | img\_path |  |
| video\_descr | theatre\_id | film\_tv\_id |  |  |
| img\_ID | th\_year | theatre\_id | cv\_ID | **experience** |
| img\_alloc | th\_role | training\_id | film\_tv\_id |  |
| img\_path | th\_production |  | theatre\_id |  |
| ai\_images | th\_director | film\_tv\_id | training\_id |  |
| ai\_videos | th\_company | film\_year |  |  |
| ai\_music | ai\_images | film\_role | film\_tv\_id | **films** |
| ai\_films | ai\_videos | film\_production | film\_year |  |
| ai\_theatre | ai\_music | film\_director | film\_role |  |
| ai\_training | ai\_films | film\_company | film\_production |  |
|  | ai\_theatre |  | film\_director |  |
|  | ai\_training | theatre\_id | film\_company |  |
|  |  | th\_year |  |  |
|  |  | th\_role | theatre\_id | **theatre** |
|  |  | th\_production | th\_year |  |
|  |  | th\_director | th\_role |  |
|  |  | th\_company | th\_production |  |
|  |  |  | th\_director |  |
|  |  | training\_id | th\_company |  |
|  |  | training |  |  |
|  |  |  | training\_id | **training** |
|  |  | cv\_ID | training |  |
|  |  | ai\_images |  |  |
|  |  | ai\_videos | cv\_ID | **counters** |
|  |  | ai\_music | ai\_images |  |
|  |  | ai\_films | ai\_videos |  |
|  |  | ai\_theatre | ai\_music |  |
|  |  | ai\_training | ai\_films |  |
|  |  |  | ai\_theatre |  |
|  |  |  | ai\_training |  |

## **Tables and data types**

AI – Auto Increment

PK – Primary Key

counters

| **Column** | **Type** | **Comments** |
| --- | --- | --- |
| cv\_ID | int(11) | PK |
| ai\_images | int(11) |  |
| ai\_videos | int(11) |  |
| ai\_music | int(11) |  |
| ai\_films | int(11) |  |
| ai\_theatre | int(11) |  |
| ai\_training | int(11) |  |

cv

| **Column** | **Type** | **Comments** |
| --- | --- | --- |
| cv\_ID | int(11) | AI, PK |
| cv\_name | varchar(255) |  |
| cv\_equity | varchar(255) |  |
| cv\_email | varchar(255) |  |
| cv\_accents | varchar(255) |  |
| cv\_skills | varchar(255) |  |
| cv\_height | varchar(255) |  |
| cv\_chest | varchar(255) |  |
| cv\_waist | varchar(255) |  |
| cv\_inside\_leg | varchar(255) |  |
| cv\_eyes | varchar(255) |  |
| cv\_hair | varchar(255) |  |
| cv\_build | varchar(255) |  |
| cv\_playing\_age | varchar(255) |  |

experience

| **Column** | **Type** | **Comments** |
| --- | --- | --- |
| cv\_ID | int(11) | PK |
| film\_tv\_ID | int(11) | PK |
| theatre\_ID | int(11) | PK |
| training\_ID | int(11) | PK |

films

| **Column** | **Type** | **Comments** |
| --- | --- | --- |
| film\_tv\_ID | int(11) | PK |
| film\_year | year(4) |  |
| film\_role | varchar(255) |  |
| film\_production | varchar(255) |  |
| film\_director | varchar(255) |  |
| film\_company | varchar(255) |  |

images

| **Column** | **Type** | **Comments** |
| --- | --- | --- |
| image\_ID | int(11) | PK |
| image\_title | varchar(255) |  |
| image\_descr | varchar(500) |  |
| image\_path | varchar(255) |  |
| image\_group | varchar(255) |  |
| image\_folder | varchar(255) |  |

media

| **Column** | **Type** | **Comments** |
| --- | --- | --- |
| usr\_ID | int(11) | PK |
| music\_ID | int(11) | PK |
| video\_ID | int(11) | PK |
| image\_ID | int(11) | PK |

music

| **Column** | **Type** | **Comments** |
| --- | --- | --- |
| music\_ID | int(11) | PK |
| music\_title | varchar(255) |  |
| music\_descr | varchar(500) |  |
| music\_path | varchar(255) |  |
| music\_group | varchar(255) |  |
| music\_folder | varchar(255) |  |

theatre

| **Column** | **Type** | **Comments** |
| --- | --- | --- |
| theatre\_ID | int(11) | PK |
| theatre\_year | year(4) |  |
| theatre\_role | varchar(255) |  |
| theatre\_production | varchar(255) |  |
| theatre\_director | varchar(255) |  |
| theatre\_company | varchar(255) |  |

training

| **Column** | **Type** | **Comments** |
| --- | --- | --- |
| training\_ID | int(11) | PK |
| training | varchar(255) |  |

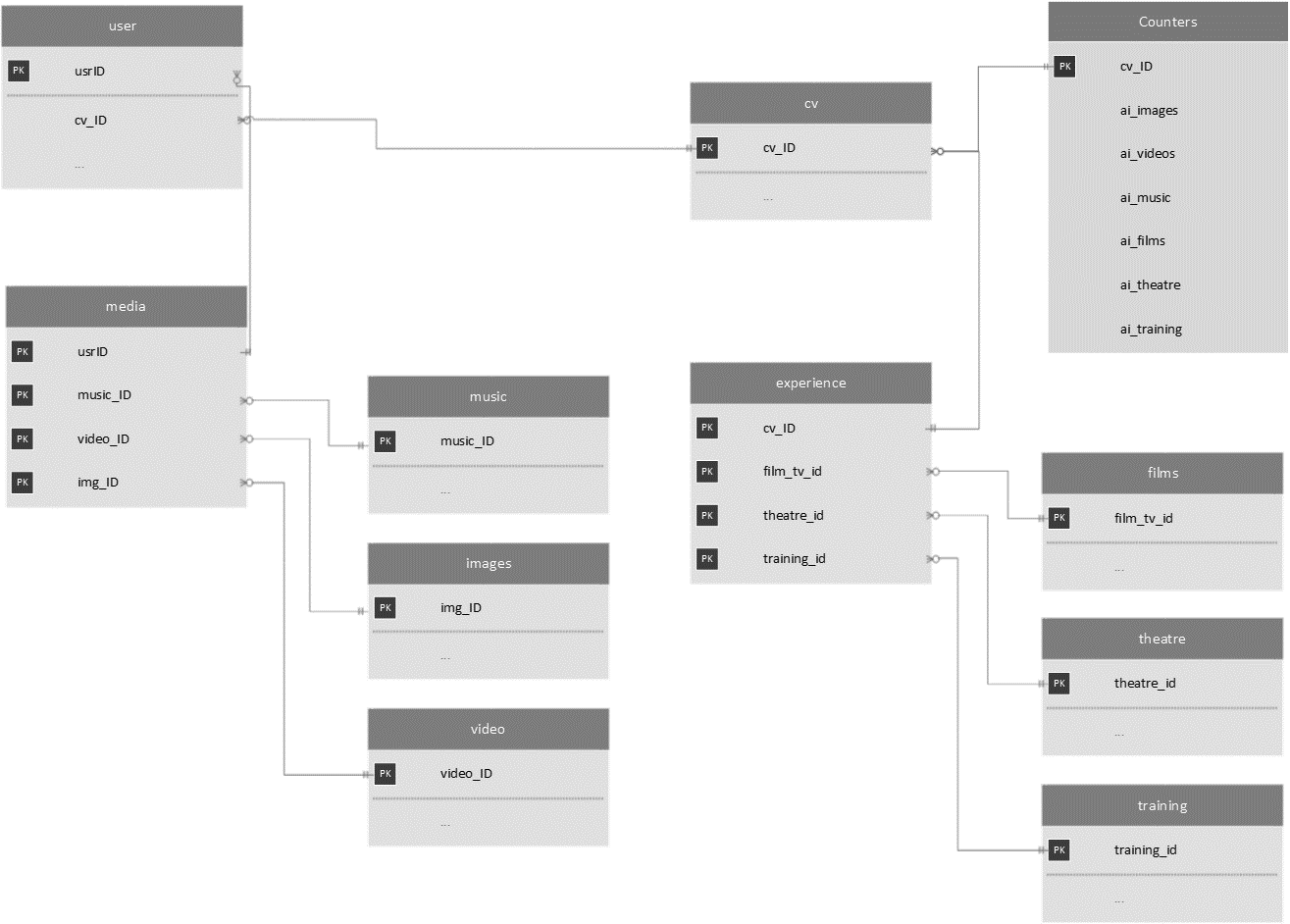
user

| **Column** | **Type** | **Comments** |
| --- | --- | --- |
| usr\_ID | int(11) | AI, PK |
| cv\_ID | int(11) |  |
| usr\_name | varchar(255) | Unique |
| usr\_psswd | varchar(255) |  |

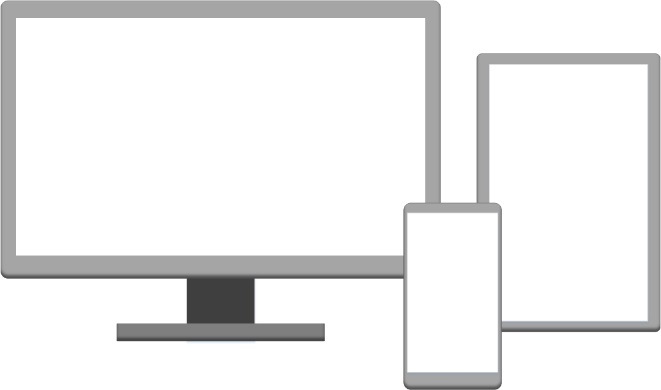
video

| **Column** | **Type** | **Comments** |
| --- | --- | --- |
| video\_ID | int(11) | PK |
| video\_title | varchar(255) |  |
| video\_descr | varchar(500) |  |
| video\_path | varchar(255) |  |
| video\_group | varchar(255) |  |
| video\_folder | varchar(255) |  |

## **Tables and Relations**



# **Design**



“Web Portfolio for Jamie Rodden” project is going to be consisted of a website (available for any visitor) and a content management system (available for administrator/owner of the website). The main idea of the project is to make a responsive design what would allow to use the website and content management system on any device with internet connection (devices like PC, tablet or smartphone). It will look like a modern standard website on large screens and like a mobile app on small screens. The design is going to contain some elements based on material design principles.

Layouts

I decided to use a navigation bar (with a logo and all the links to another pages) on top of the screen for large screens and a navigation bar (with a logo and a toggle button, what is going to open and close the side panel with a menu items) for small screens. This is the most popular approach for positioning of elements on the web and every user should be familiar with that already. Links of the navigation bar (large screens) and the toggle button (small screens) are pulled to the right side as many users use the right hand to navigate on the website.

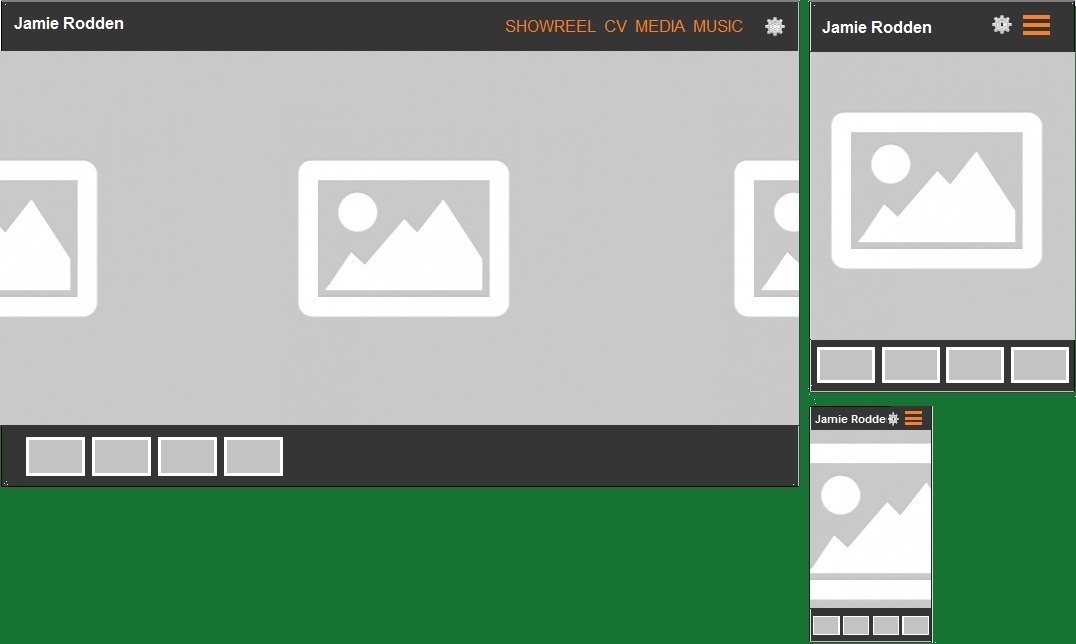
Layouts for the control panel are different as the purpose of the control panel is also different from the main website available for users. A navigation bar is used to display the logo what demonstrates what it is a control panel section and a log out button (on the right), what gives an option to log out from the control panel and go to the main website. As well there is a toggle button (on the left) in the navigation bar, what open and close a side panel consisting from the main menu items of the control panel (on small screens). The main menu of the control panel is positioned on the left side and the next to the right is the main content field (on large screens). That approach is used as we are reading from the left to right and the first action administrator need to do is to select a section he want to work on. After the option was selected user is going to work in the middle and right part of the page (in the main content field). Many people are holding a mouse in the right hand, so to work in the middle and right side of the screen should be very comfortable while the menu, positioned on the left, is not going to disturb. Action buttons of the control panel are placed under the navigation bar in a submenu (on large screens) and in a toggle button on the bottom right corner (on small screens). The idea of the elements positioning was taken from the standard positioning of elements what you can see on windows platforms (for the large screens) and from the android google apps (for the small screens).

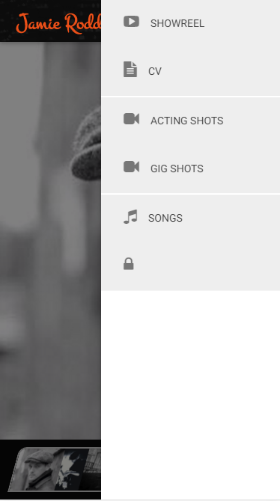
Colours and effects

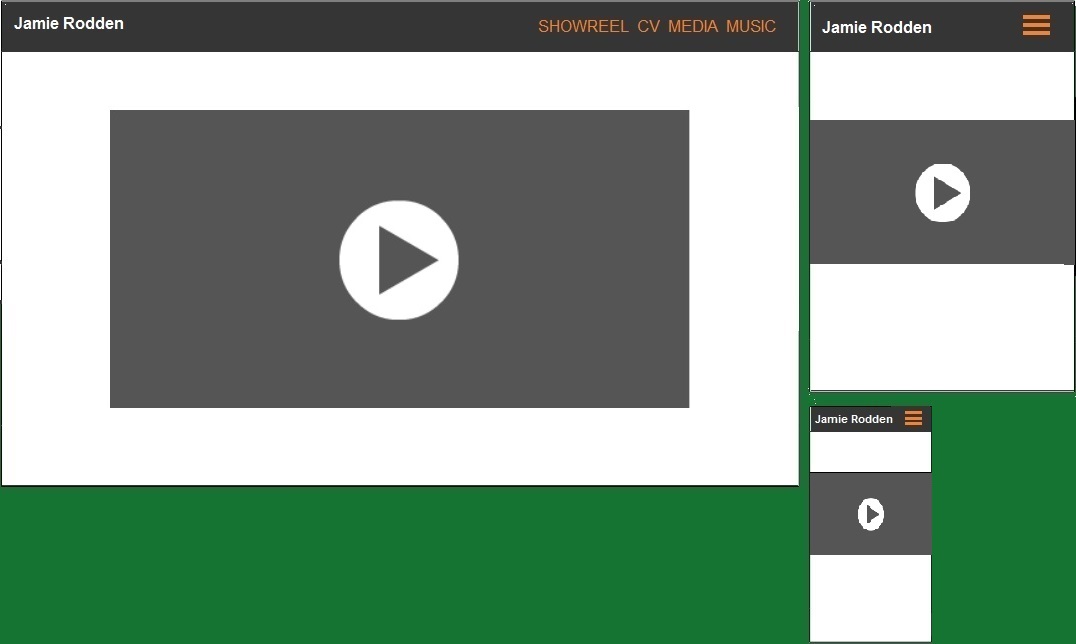
The part of the website available for the users is going to be black and white as it was one of the requirements of the client for the design. The only element what is going to be in red is a logo of the website as it will take an attention of the users a clearly demonstrate what is this website about.

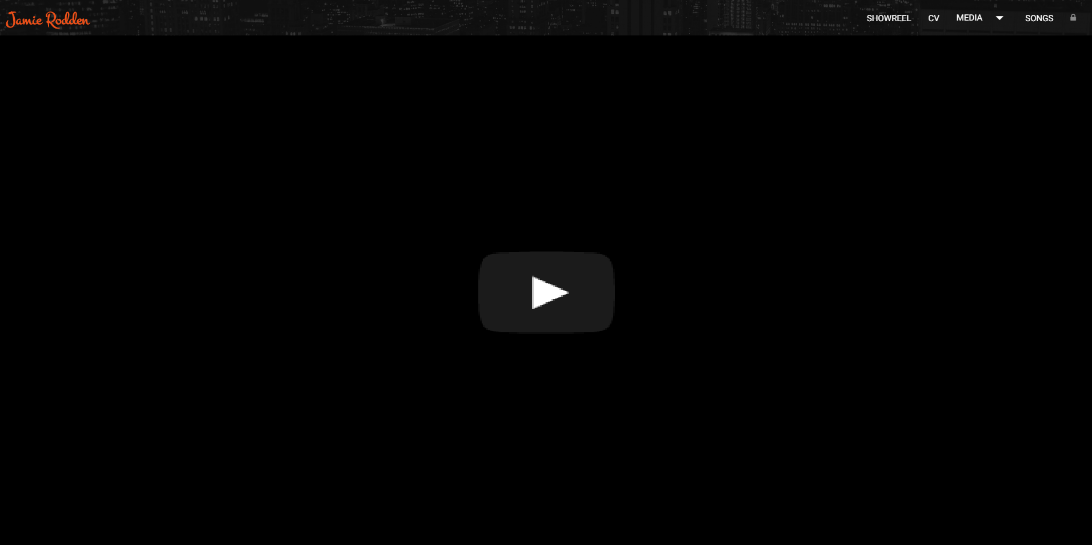
Main colours of the content management system are going to be teal and orange as I found that combination of colours very interesting and not disturbing for the eyes. Also transparency, colours with the meaning (like red – warning, blue – information etc.) and shadow effects will be applied on some of the elements making a feeling of a layer system. That would help the user to be more confident during the usage of the application and will help splitting the content into logical meanings.

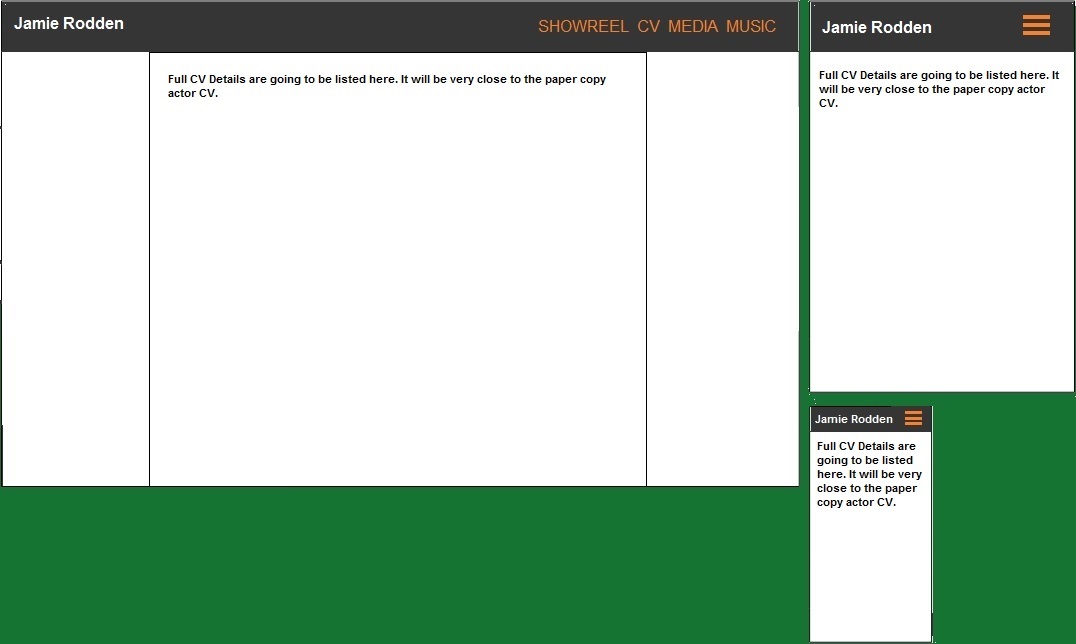


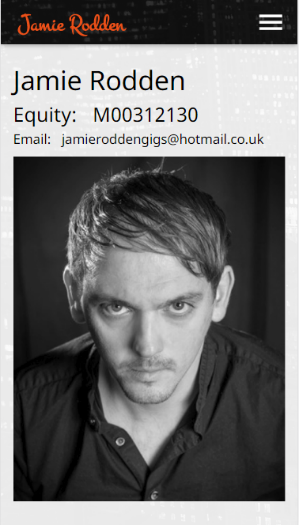
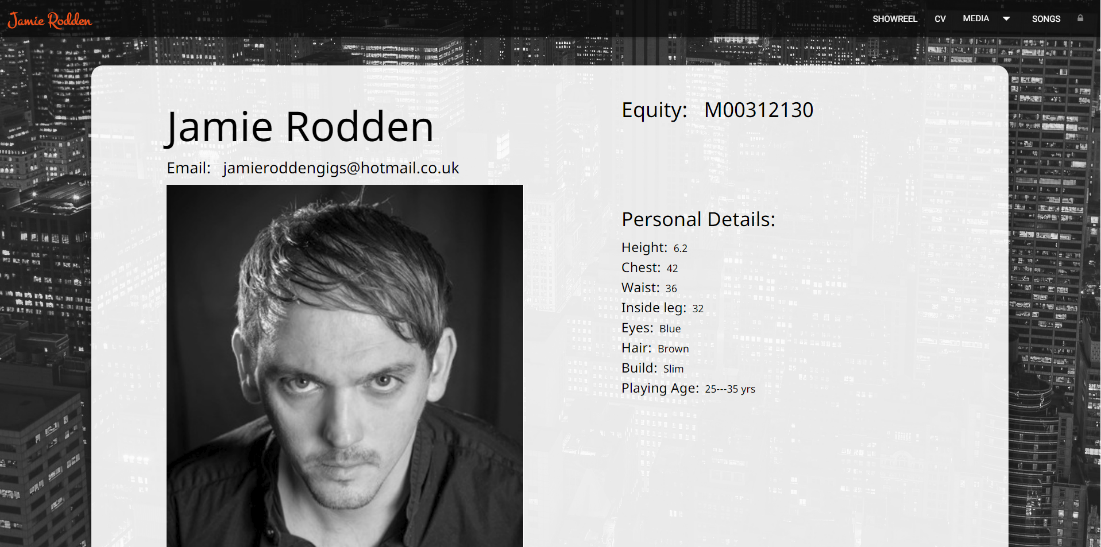
Landing (home) page with horizontal sliding image gallery. User is able to tap and scroll images horizontally on touch devices or use the minified images under the gallery to scroll images by clicking on them. Also user can go to the pages displayed in the navigation menu on the top. On small screens the menu can be accessed by clicking on menu toggle button. On a gear icon click a modal with a login form will pop up and will let log in into the content management system of the website.

Updated and improved version of the design prepared for the second submission.

Showreel page is going to consist from a single showreel video and available menu on the top of the page. User can play, pause the video embedded from the YouTube. Navigation menu is also accessible.

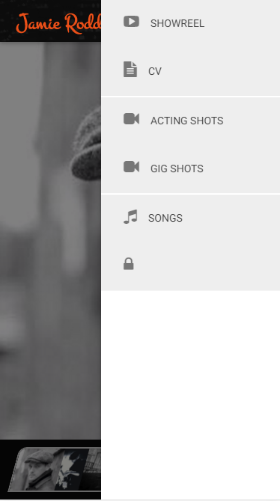
Updated and improved version of the design prepared for the second submission.

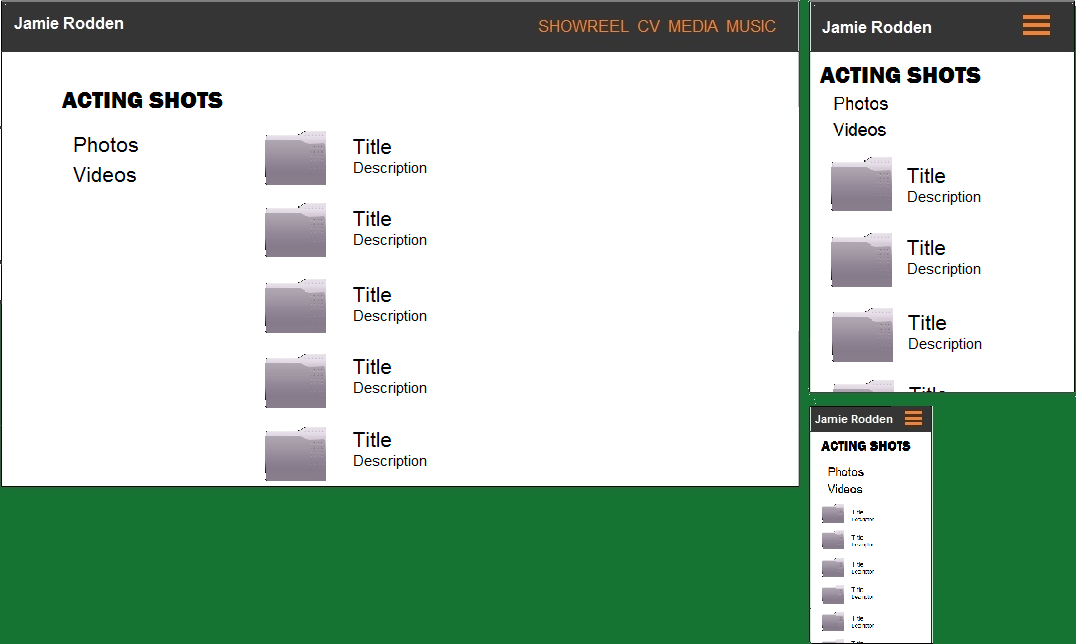
CV page is going to consists of a full actor’s CV with all the relevant information provided by the client. User can read the information of CV and navigate to other pages using the menu available on top.

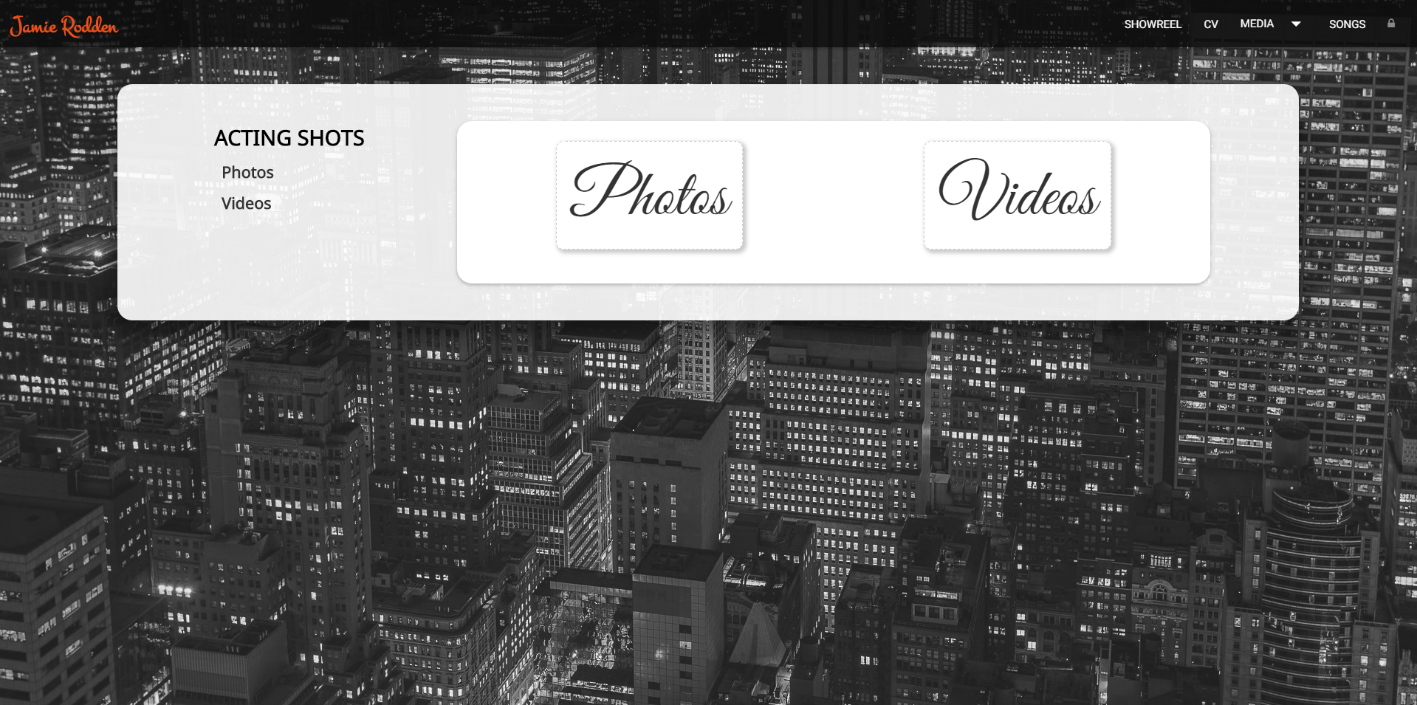
Updated and improved version of the design prepared for the second submission.

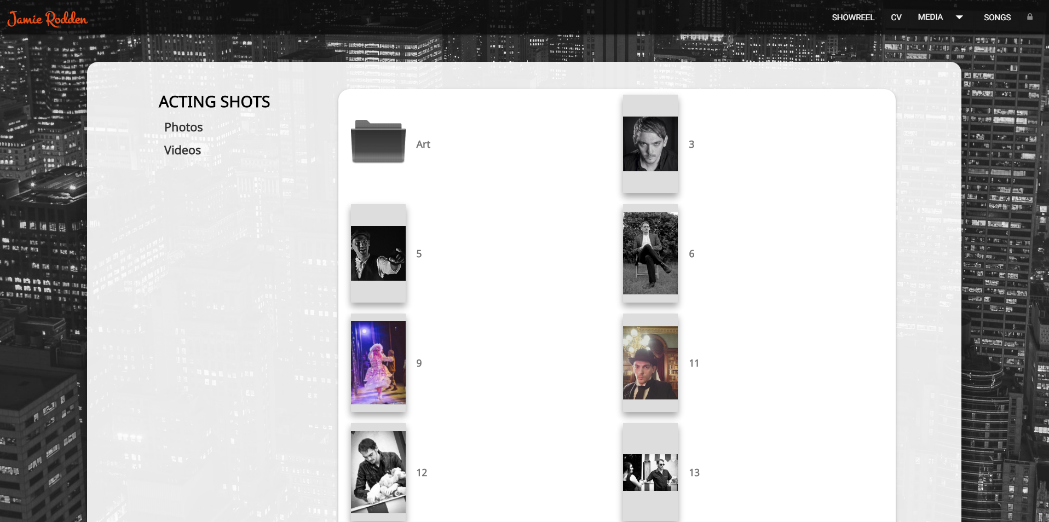
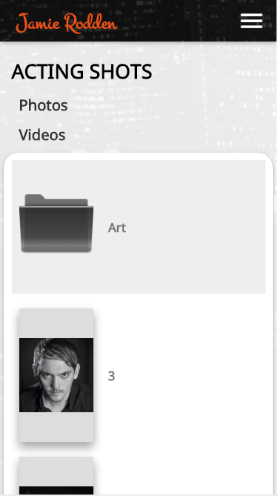
Media page is consisting of two groups (acting shots and gig shots). It is possible to access photos and videos of a group by selecting the big icon. Following pages are going to be identic, only content will be different. Example is shown below.

During the implementation stage some changes were applied for the design. Instead of the having a media page consisting of the 2 links (Acting and Gig shots) it was decided to use a dropdown button in the menu with 2 submenu options (for Acting and Gig shots) for the large screens. It was decided to use Acting and Gig shots as a separate options in a side sliding menu on small screens. That would help to avoid an additional screen and improve a navigation. The dropdown menu is displayed on the image below.

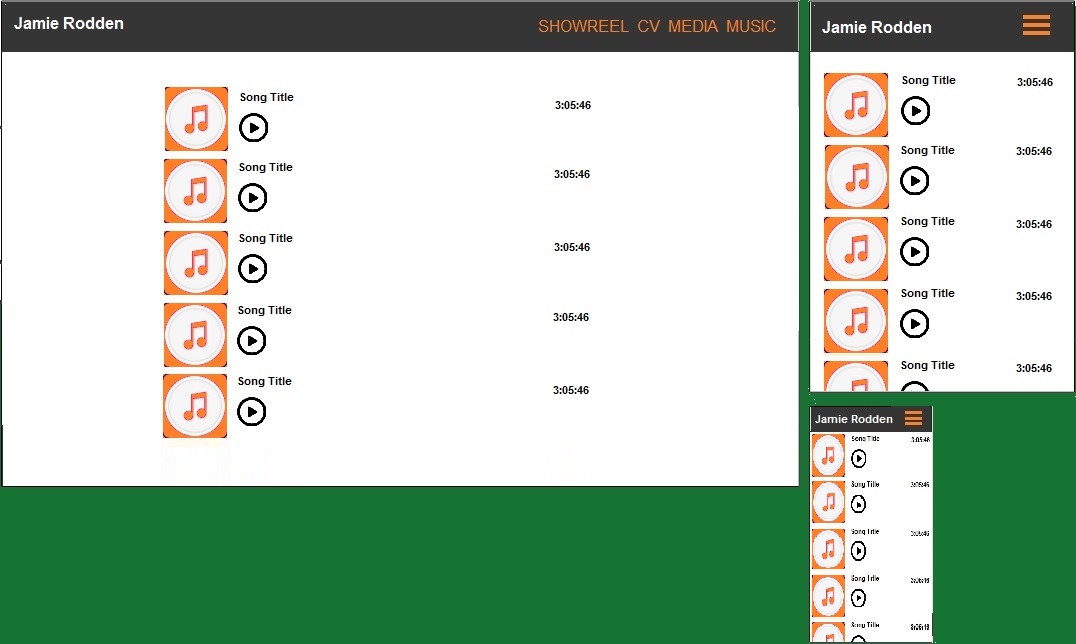


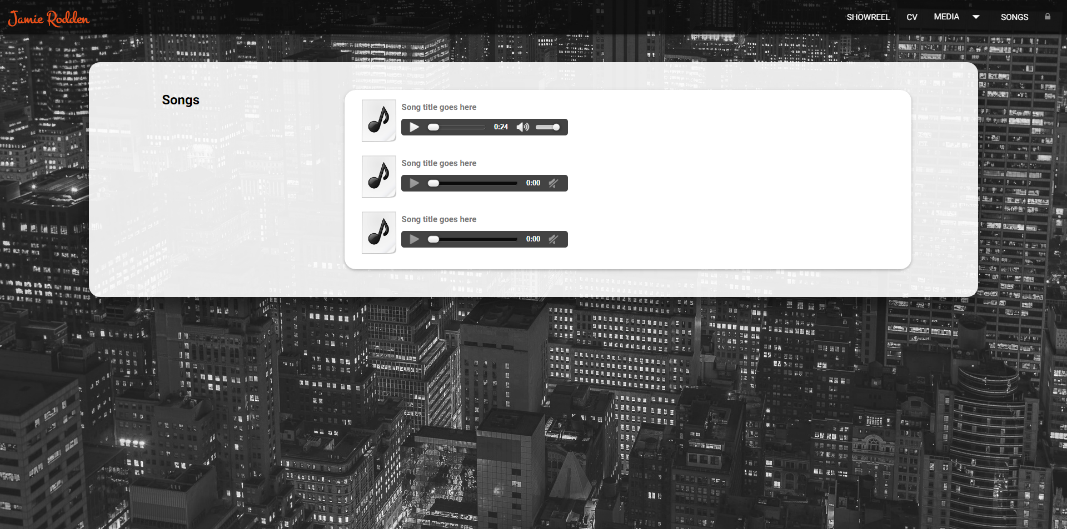
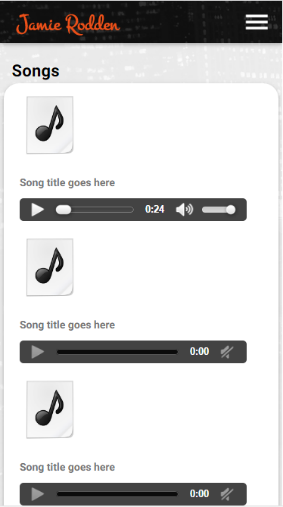
Acting shots and Gig shots pages are going to be absolutely similar. User can navigate to them using the menu on the left side (top on small screens). Albums are displayed on the top and the photos/ videos without the album are going to be listed after the albums.

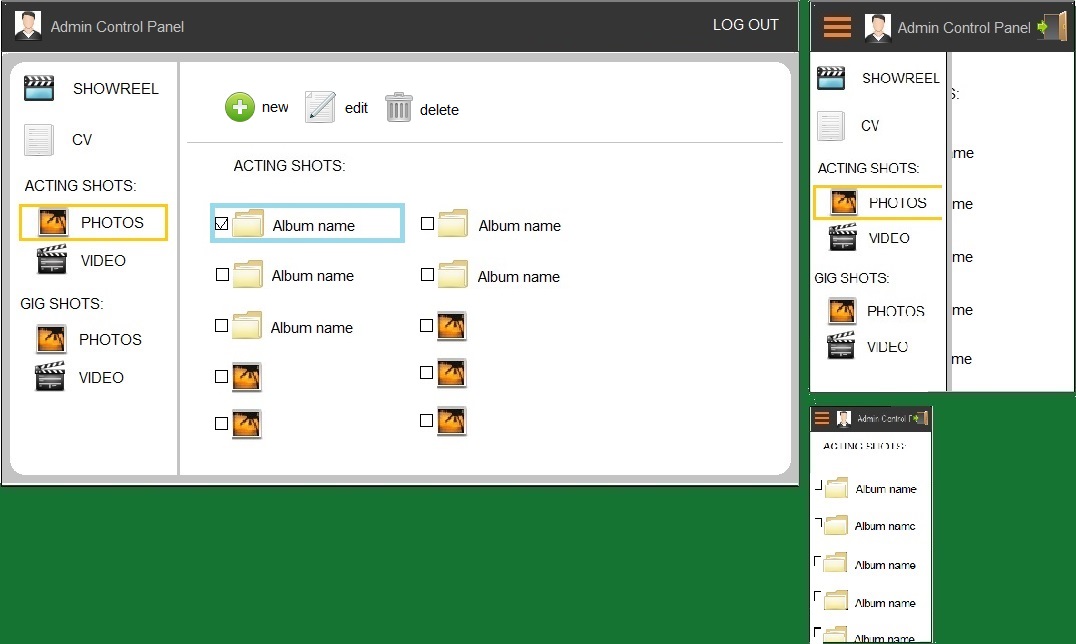
Updated and improved version of the design prepared for the second submission.

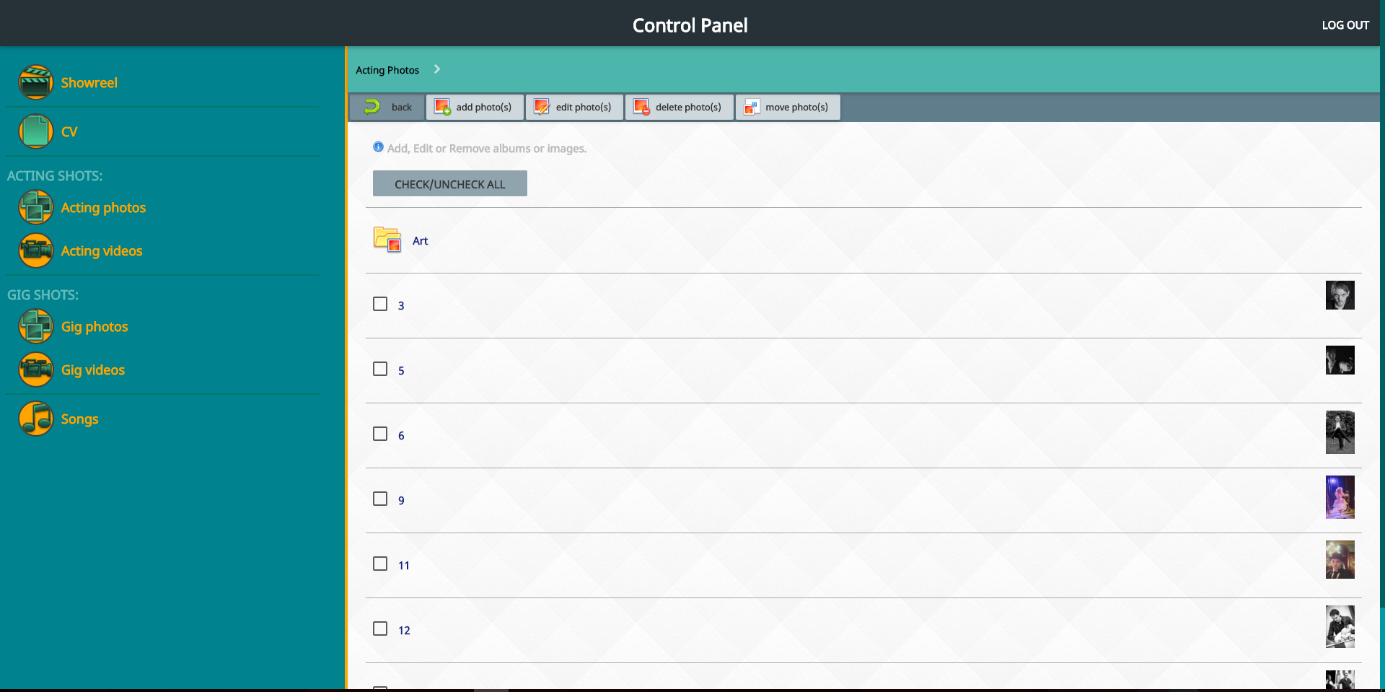


Music page with the list of audio files. User can play and stop audio.

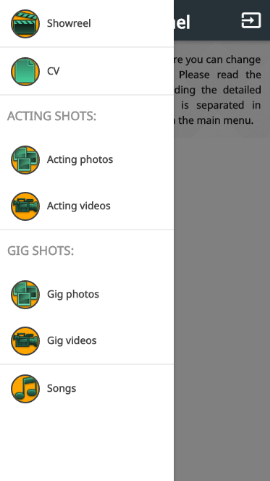
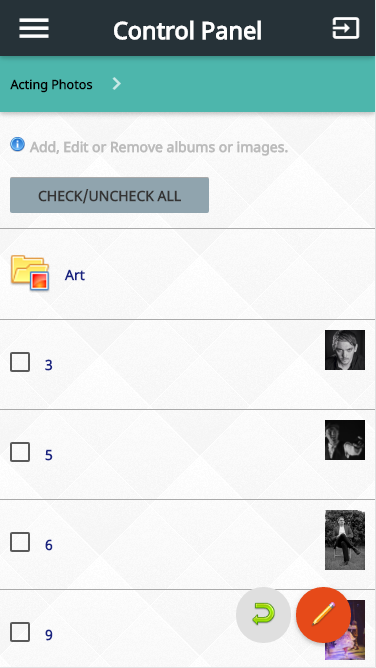
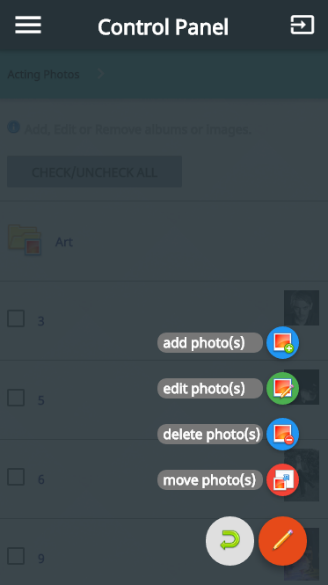


Updated and improved version of the design prepared for the second submission.

 Administration Control Panel. Control Panel allows to change a showreel video, to change and add CV content, change information of the photos and videos as well as upload new files or delete current.

Updated and improved version of the design prepared for the second submission.

Options to upload, edit, delete and move between the albums and the main gallery (no album) are going to be added in the content management system. Click on an album icon will allow to enter and review the content of the album. Every photo/video are going to have the name and description what can be edited or removed.

Mobile view of the Content Management System.

References:

<https://laracasts.com/discuss/channels/general-discussion/is-there-a-specific-reason-why-php-is-hated-by-so-many-developers>

<http://www.databasejournal.com/features/mysql/slideshows/top-10-mysql-gui-tools.html>

<https://en.wikipedia.org/wiki/WampServer>

<http://materializecss.com/>

<http://www.w3schools.com/ajax/>

<http://api.jquery.com/jquery.ajax/>

<http://stackoverflow.com/questions/6974684/how-to-send-formdata-objects-with-ajax-requests-in-jquery>

<http://stackoverflow.com/questions/5392344/sending-multipart-formdata-with-jquery-ajax?lq=1>