**Group A3 – Music Zombies’ Team Report**

**1. Team Details**

Team A3 aka Music Zombies from the 12AM lab on Tuesdays

**2. Team Members**

|  |  |  |
| --- | --- | --- |
| **NAME** | **PRIMARY ROLE** | **SECONDARY ROLE** |
| James Hamm | JavaScript | Team Leader |
| Tobias Burns | JavaScript | Logo/Image Design |
| Hamza Zahoor | HTML | CSS |
| Jeanette Moran | HTML | CSS |
| Brendan Mullahey | PHP | Databases |
| Patrick Behan | PHP | Databases |

**3. Proposed Application**

Our group has decided to design and implement a website which sells both CD's and digital music, along with music based merchandise. We hope to include music to accommodate many tastes and preferences.

**4. List of Functional and Non-Functional requirements**

Functional

Registration of new users

Confirmation of registration email

Display a design and logo

Ability for users and admins to login

Search bar for making queries about items

Advanced search filtering options

Navigation between site pages

Database for all items

Admin able to edit the database contents through the website

Display of search results and appropriate items from the database

Show all the details associated with a specific product

Include an add to basket button for each product page

Basket for saving wanted items

Checkout where users can see relevant information such as total price and a list of every item.

Purchase button and confirmation of order (including email)

Non-Functional

Consistency

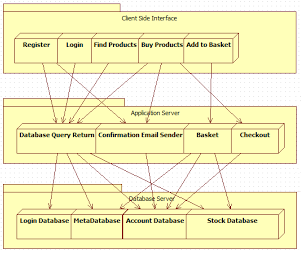
Extensible, easy to add features in the future

Compliance with coding standards, all pages passing a lint test

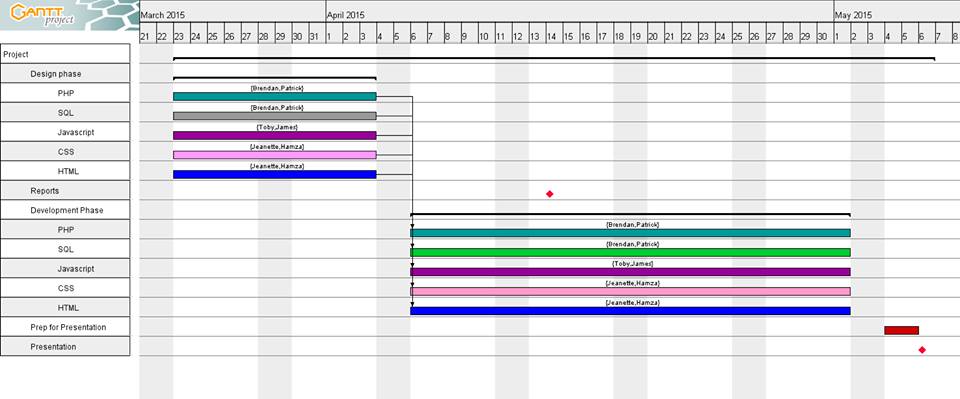
Age Guidance

Ease of use

**5. System Architecture Design**



**6. Gantt Chart**



We have decided to dedicate the beginning of the project time so that each team pair can research and learn more about their chosen field, and we collaborated to design what the overall site would contain. The rest of the time is dedicated to implementing the features we want the website to have, with some time left over to prepare the presentation.

**7. Database design**

Database design: 3 tables

Table users has 6 columns

id serial number to be unique and atomic (auto increment) primary key not null

username variable character

password variable character

email variable character

country variable character

date\_of\_creation variable character

Table music: 6 columns

artist variable character primary key

song variable character

album variable character

genre variable character

year variable character

publisher variable character

Table merch: 3 columns

artist variable character primary key, relates to primary key from music table

item variable character

size variable character

**8. List of database queries**

These are example queries for the database, since there are innumerable possible queries.

SELECT artist FROM music WHERE song = 'far from home';

SELECT music.artist, merch.artist, merch.items FROM music WHERE artist = 'Five Finger Death Punch' CROSS JOIN merch ON music.artist = merch.artist;

SELECT song FROM music WHERE artist LIKE '%dragon%';

SELECT item FROM merch WHERE size = 'l';

SELECT album FROM music WHERE song LIKE 'reise' AND artist LIKE 'rammstein';

SELECT username FROM users WHERE country = 'ireland';

UPDATE merch SET size = xl WHERE item = shirt;

DELETE FROM users WHERE id = 3;

**9. PHP component specification**

config.php: Connects to the sql database. In a seperate file to use the include command in other files.

search.php: Takes input from the search form, queries the input against the music database to search for content, matching against anything like the entered term, then returns and displays any results to the page along with the number of results. If nothing is entered into the form, tells the user to enter a search term.

registration.php: Takes input for username, password, email address, and country, from the registration form, queries the user database and inserts the information into the next available slot in the database. Instead of inserting the password, adds random characters to the password and stores that in place of the password. Also sends an email confirmation upon successful registration. If unable to find the users email address, displays that the email address could not be found. If the email is not sent, displays the email could not be sent.

login.php: Takes input for username and password from the login form, queries the users database and checks the input against the stored information. If the username and password do not match the database, displays that either the username or password is incorrect.

purchase.php: Queries the users database upon a purchase being made, retrieves the email address for that user and sends an email to the user confirming their purchase.

**10. JavaScript component specification**

Slideshow.js: Rotate through pictures to show our products, either automatically, or with user prompt.

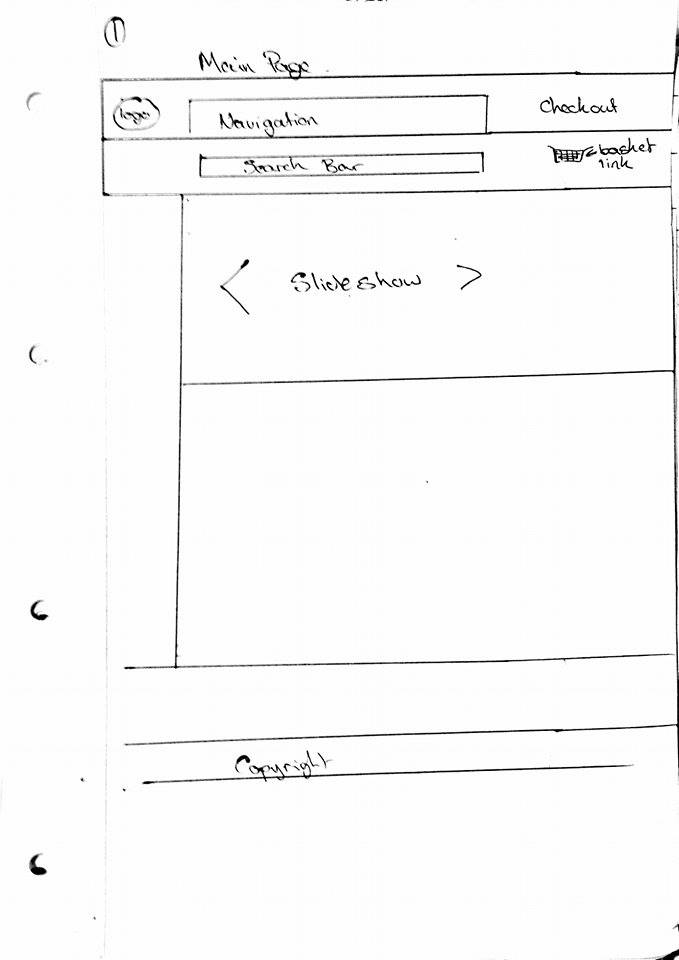
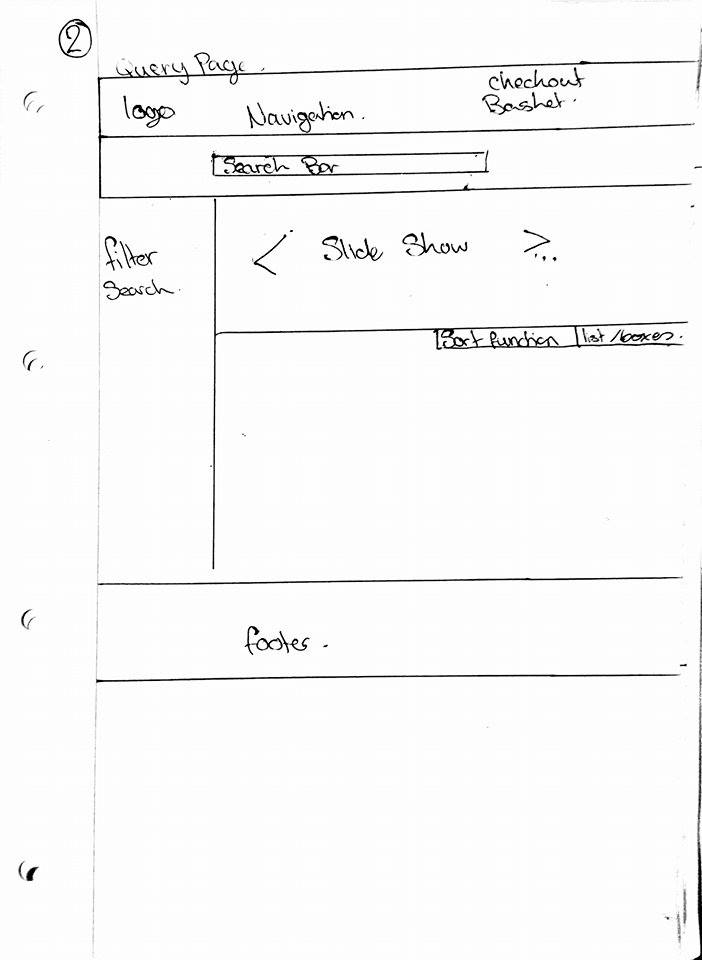
Cookies.js: Store user information throughout a session, such as name, login time, and preferences so the database is not queried every time another internal page is loaded.

Master.js: Will hold small scripts such as expanding a search bar or site wide alerts such as cookies not enabled or email was sent.

Validator.js: Validates any form information such as email addresses, names, quantity, etc before it is turned into a database query

**11. User interface design**

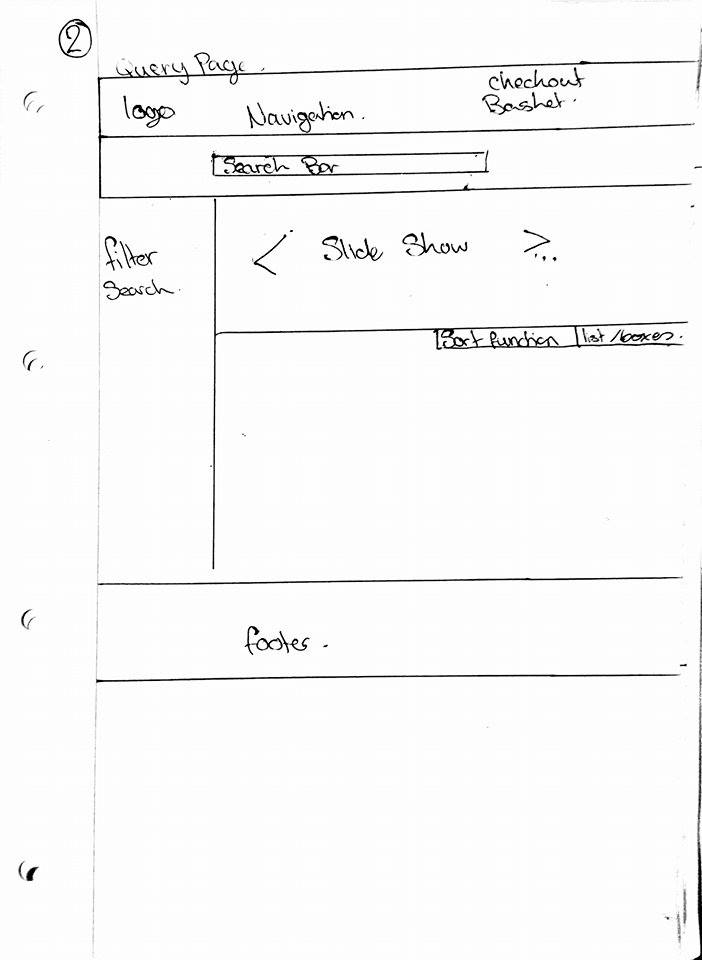
1. **Homepage**



This page will appear when the website is opened. It allows the user to navigate the website.

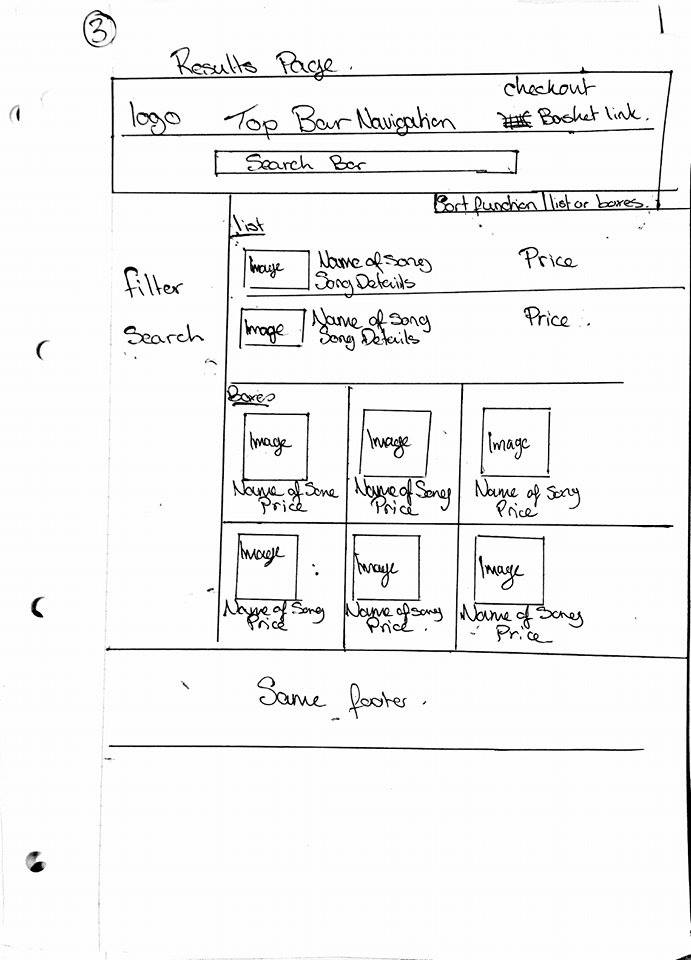
The side bar will show a list of general search options, such as price filters, artist names, genre filters, Year of release, etc. Options selected on the home page will lead the user to the query page. Alternatively the user can use the search bar which will bring them straight to the results page.

1. **Query page**



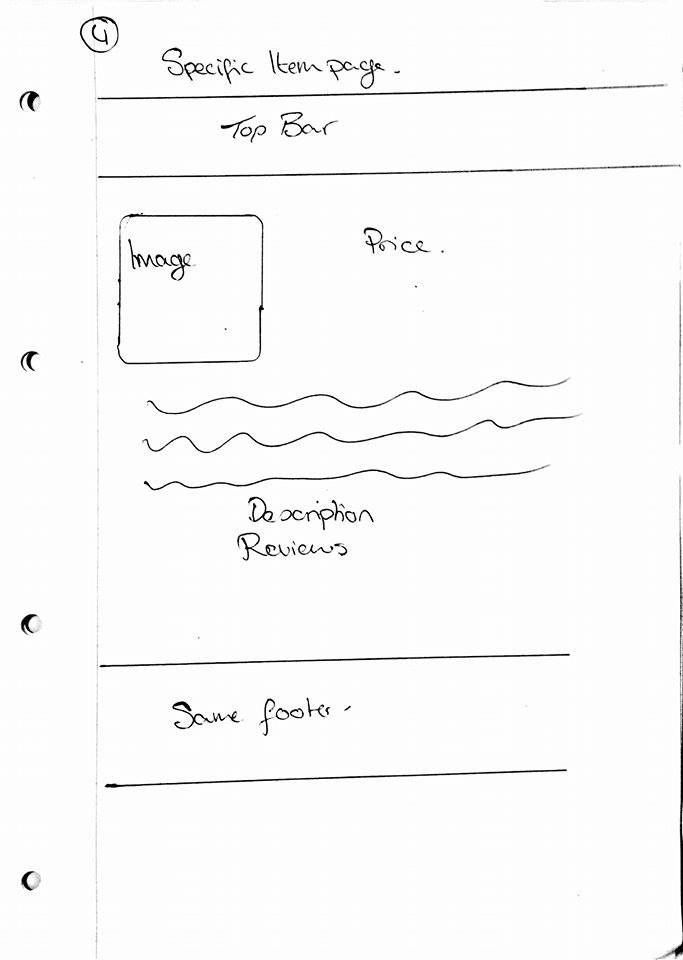
The query page will contain a list of more specific features and search options. It will also only display items which are relevant to the option selected on the home page. When the user searches items on the query page the webpage will display the results on the results page.

1. **Results page**

****

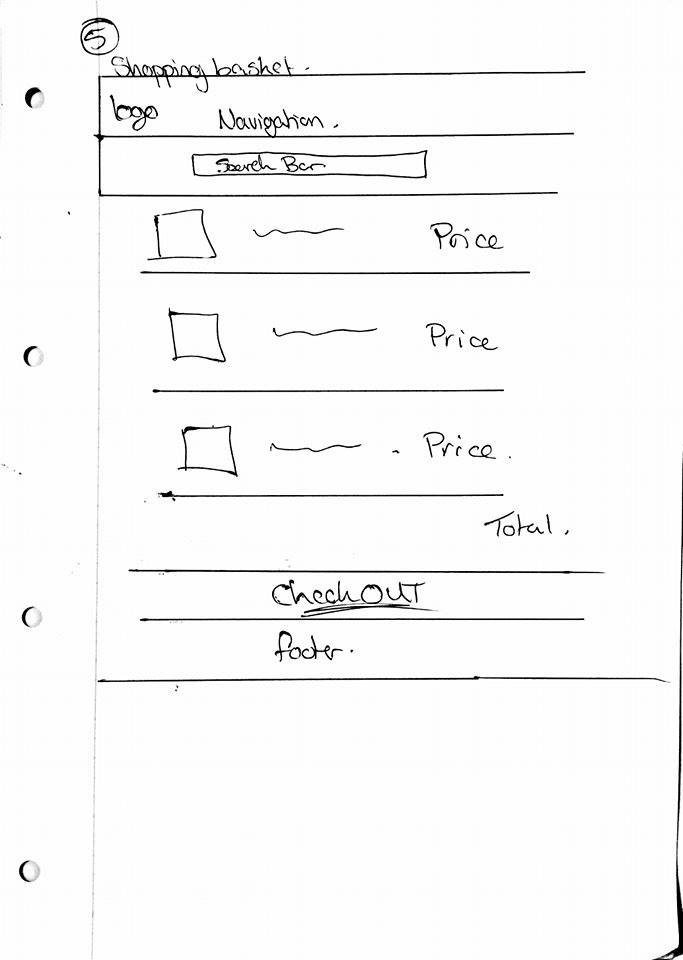
The results page will display a list of items relevant to the users search. Users will be able to select items to go to the items description page.

1. **Items description page**



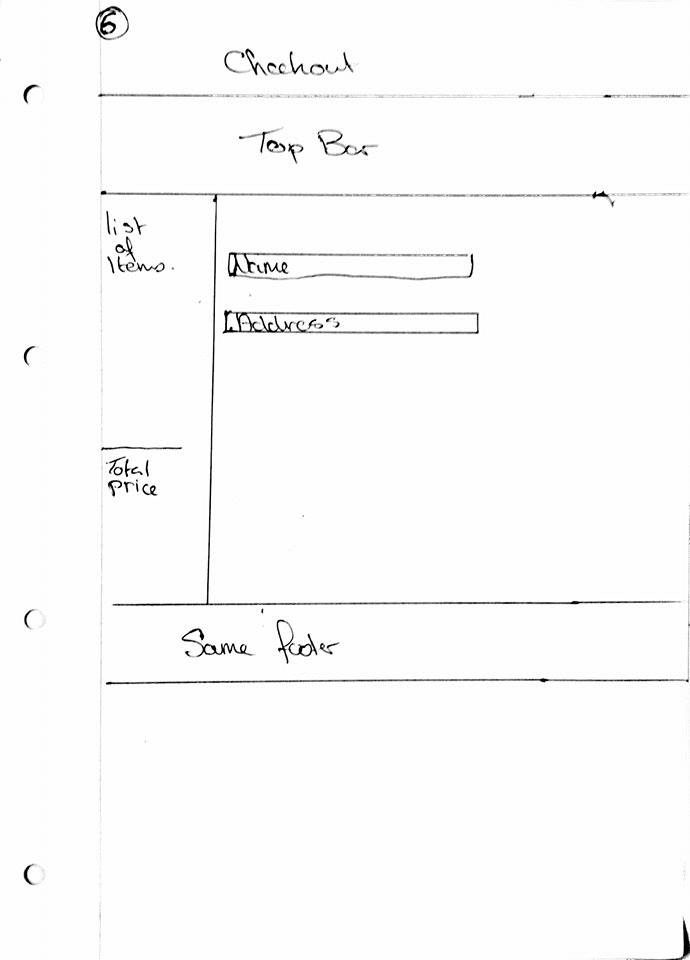
This page will provide a description of the item selected. It will also provide reviews of the item ad an option to add item to shopping basket.

1. **Shopping basket**



The shopping basket will display a list of all items selected by the user as well as the total price of all the items. The user can then select to go to checkout which will bring them to the checkout page.

1. **Checkout/payment page**



In this page the user will fill in all details needed for delivery and purchase of the item(s). The page will also provide a list of the items in the shopping basket as well as the total price of the items.