THE SHARK GAME PROJECT

Functional Specification

Technical specification

Code Location

I. Functional Specification

# 1. Introduction

### 1.1 Overview

The Shark Game Project involves the development of a game which will be embedded into a site. The game is targeted towards adults and children. The aim of the game is to crop the sharks in the displayed images and tag them which would then be stored in a database. The game is being developed to be cross platform and should be functional on many devices e.g. phones, tablets and PCs.

# 1.1.1 Home Page

The home page or the first page the player will see will be the Sign In page. The player can either register or sign in using a Facebook/Google+ account. The footer displays the text “SHARK PROJECT” which links to the About page.

<http://localhost/SharkGameProject/MAIN/main_signin.php>

# 1.1.2 Game Page

Cropping

After signing in the player should get a sign in confirmation and be be taken to the Game page. This page will display an image containing a shark. The player's aim is to draw a box around the shark only. This box functions similar to a crop box.

Tagging

After drawing the crop box around the shark, a text box should appear on the cropped image prompting the player to type a tag into it. The player would then type 1 tag only.

There should be an auto-fill functionality which should aid the player in filling when typing the tag. The auto-fill can only help fill in the 14 sharks which this project concentrates on.

Check Mark

The text box should have a check mark (✔) symbol beside it and it should light up after the tag is completed. As the player clicks on the lit up check mark, the next image should appear. This check mark should only light up if the player has typed in either of the 14 shark species.

<http://localhost/SharkGameProject/MAIN/game.html>

# 1.1.3 Shark Guide Page

There is a separate page for the Shark Guide. This guide exhibits information on the 14 different sharks which the players will be tagging. Each shark will be represented by an image and a species name. Each of them will have a features section which should address Dorsal Fin, Tail, Caudal Fin, Side Fins (Pectoral Fins), Anal Fins etc. This should help the players in identifying the sharks that are displayed in the images on the Game page.

<http://localhost/SharkGameProject/MAIN/sguide.html>

# 1.1.4 Leaderboard Page

This page which will display Usernames and Points corresponding to each of the Username.

<http://localhost/SharkGameProject/MAIN/leaderboard.html>

# 1.1.5 About Page

About Page is reserved for future details.

<http://localhost/SharkGameProject/MAIN/about.html>

# 1.1.6 Shark Project Header

The “SHARK PROJECT” is a temporary header which links to the About Page. The aim is to place a logo there in the future.

# 2. Game Strategy

This defines how the game should operate. The gaming needs and requirements for the project have also been established.

# 2.1 Correct Tag

As the players tag the sharks in the images, these tags are stored in a database. In order to figure out which is the correct tag for a shark image, a limit should be met on how many times a said image has been tagged by a specific tag. An example would be, If a shark image has been tagged “Bull Shark” at least ten times then that shark in the image shall be taken as the Bull Shark.

# 2.2 Player Points

The points for each player is calculated based on what the tag they suggest is and what most likely the correct tag is for each image. The points for the players will not be updated until the correct sharks in the images are confirmed i.e. a shark in an image has reached the limit of being tagged “Bull Shark” ten times and hence shall be confirmed as Bull Shark. The player that tagged this image “Bull Shark” a few days ago will receive their points immediately.

**II. Technical Specification**

# 1. Technologies

The following is a list of all the technologies used to develop the game and other elements of the project up till date.

# 1.1 Bootstrap v3.3.5

Bootstrap is a HTML, CSS, JavaScript framework. It is used as the most common basis of front end development. It allows for cross platform compatibility i.e. it scales websites and applications to fit any device; from phones to tablets to PCs.

The pages in the project contain basic Bootstrap elements. A simple example can be found here: [http://getbootstrap.com/getting-started/#template](http://getbootstrap.com/getting-started/" \l "template)

The following link provides a run down on the CSS and Bootstrap elements which provide functionality for the page: <http://getbootstrap.com/css/> This page should be studied before attempting to edit or add any new code.

CDN

A CDN is a content delivery network which provides webpages on servers. Bootstrap has source code which can either be downloaded and then inserted into the project or source code in the form of links which can be inserted via CDNs. CDNs are not reliable since if the server crashes or stops responding, the website will malfunction or experience errors.

It is vital that no Bootstrap CDNs are used in this project. The website in this project need not be dependent on CDNs. Rather the bootstrap pre-compiled code found in the folder '**Bootstrap-3.3.5**' should always be inserted into the current pages and any more pages developed. The Ajax link at the end of a Bootstrap page is the **ONLY** exception.

Versions

Version 3.3.5 was used at the time of development and is still used but version [4.0.0 alpha](http://v4-alpha.getbootstrap.com/) is now readily available since 19/08/15. v3 will still be kept updated by Bootstrap and hence should be continued in the project's development.

# 1.2 Jcrop v0.9.12

This is a plugin which comprises of jQuery. It is used as the cropping functionality in the project. Jcrop is only placed in the Game page.

This is used to crop the image. It also has the option of displaying the coordinates of the cropped image, adding animations, and many other functions.

This project requires Jcrop to crop the image and store the coordinates of the cropped image in a database. Cropping should be enabled in any platform i.e. phones, tablets, PCs.

Versions

[Jcrop v0.9.12](file:///srv/www/htdocs/SharkGameProject/Jcrop-0.9.12/index.html) was used during development but it does not implement the cropping feature when switched to tablet or phone viewing. For this, [Jcrop v2.0.0](file:///srv/www/htdocs/SharkGameProject/Jcrop-2.0.0/index.html) should be used to allow for cross platform compatibility. In order to understand Jcrop, both versions must be studied before editing or the addition of any code. Both versions are located in their corresponding folders; '**Jcrop-0.9.12**' and '**Jcrop-2.0.0**'.

A folder containing tests which were carried out with v2.0.0 are located in the '**MAIN**' folder called '**testing-cropping-with-Jcrop-v2.0.0**'.

There are 3 things to keep in mind when implementing the Jcrop functionality. The following is the example for Jcrop v0.9.12 on how to implement Jcrop.

**Jcrop v0.9.12: One**

**Jcrop v0.9.12: Two**

**Jcrop v0.9.12: Three**

After the three steps, in order for Jcrop to be operable, the following should be implemented in the <body> section to display the coordinates and to draw a crop box around the shark.

# 1.3 XAMPP/LAMP

[XAMPP](https://www.apachefriends.org/index.html) is a server stack. It provided the creation of a MySQL database using phpMyAdmin. Apache HTTP server is used to host the site. The project was first developed on Windows and hence required XAMPP. It was then moved onto [LAMP](https://bitnami.com/stack/lamp) and shall be continued on this OS.

LAMP is the server stack for Linux. This was used for the creation of the MySQL database using phpMyAdmin and hosted on the Apache server.

Server

The Apache server on Linux has already been set to start automatically on every reboot. The player needs to just access the site. This can be done by clicking or typing the following link: <http://localhost/SharkGameProject/MAIN/main_signin.php>

phpMyAdmin

In order to access the MySQL database, the player needs to access phpMyAdmin. phpMyAdmin allows us to manage our MySQL database online.

The following link can be clicked or typed to see the phpMyAdmin welcome screen: <http://localhost/phpMyAdmin/>

Credentials used to log in to phpMyAdmin to access the 'sharks' MySQL database:

Username: **root**

Password: **sharks**

MySQL Database

phpMyAdmin is used to manage our database. The '**sharks**' database has already been created for this project. It can be viewed here: [http://localhost/phpMyAdmin/index.php?token=0ed3c45bd75415baf1111a36d482b71c#PMAURL-1:db\_structure.php?db=sharks&table=&server=1&target=&token=0ed3c45bd75415baf1111a36d482b71c](http://localhost/phpMyAdmin/index.php?token=0ed3c45bd75415baf1111a36d482b71c" \l "PMAURL-1:db_structure.php?db=sharks&table=&server=1&target=&token=0ed3c45bd75415baf1111a36d482b71c)

[image table](http://localhost/phpMyAdmin/index.php?token=0ed3c45bd75415baf1111a36d482b71c" \l "PMAURL-23:tbl_structure.php?db=sharks&table=image&server=1&target=&token=0ed3c45bd75415baf1111a36d482b71c) will handle all the shark images. It contains the image\_id for each image which will not be similar to any other image. It is the primary key for the image table but a foreign key to the tag table. The file\_location is the location of each image when the images will be stored in a filesystem. tag is a column for the 14 different tags which will be used to tag the shark images.

[tag table](http://localhost/phpMyAdmin/index.php?token=0ed3c45bd75415baf1111a36d482b71c" \l "PMAURL-25:tbl_structure.php?db=sharks&table=tag&server=1&target=&token=0ed3c45bd75415baf1111a36d482b71c) will handle all image tags, either tagged by the player or the ones we have already tagged beforehand. tag\_id is the id for each tag, we are using 14 tags. tag\_id is also the primary key. Username corresponds to the player who will be tagging. tag is the column for the 14 tags available. verified\_tag is the tag which we would have already tagged the image with. 2oordinates are the two values; (x1, y1), (x2, y2), which will correspond to the cropped part of the image by the player. The coordinates column needs to be double checked. image\_id is the foreign key to the tag table. user\_id is the foreign tag to the user table.

[user table](http://localhost/phpMyAdmin/index.php?token=0ed3c45bd75415baf1111a36d482b71c" \l "PMAURL-28:tbl_structure.php?db=sharks&table=user&server=1&target=&token=0ed3c45bd75415baf1111a36d482b71c) contains the information of the player. user\_id is the primary key for the user table and also a foreign key for the tag table. The username is the name the player registers and uses. password represents their password they will log in with.

The user table already has 2 players inserted. The following credentials can be entered into the sign in form on the main\_signin.php to sign into the site.

Username: **saira** | Password: **1234**

Username: **barlas** | Password: **4321**

[verified\_tag table](http://localhost/phpMyAdmin/index.php?token=0ed3c45bd75415baf1111a36d482b71c" \l "PMAURL-30:tbl_structure.php?db=sharks&table=verified_tag&server=1&target=&token=0ed3c45bd75415baf1111a36d482b71c) is the table for all the images we will have pre-tagged and put into the game. The verified\_tag is the unique id for each pre-tag of an image. It is also the primary key. image\_id corresponds to the images that have been pre-tagged. It is also a foreign key to the image table.

[temptag table](http://localhost/phpMyAdmin/index.php?token=3ba1627be1e46affb3d2034aafa67663" \l "PMAURL-3:tbl_structure.php?db=sharks&table=temptag&server=1&target=&token=3ba1627be1e46affb3d2034aafa67663) corresponds to the an experimental tagging system which was being implemented. It is a table which has an id for the image. A name for the tag that would be inserted for the image. The top is the coordinate for the start of the tag box and the left is another coordinate for the start of the crop box. These are basically the top left coordinates for the corner of the tag box. When the cursor clicks on the image, the coordinates would be the place where the cursor is situated and hence a tag box would appear.

This table can be inserted or removed based on future development. The tests can be seen in the '**MAIN**' folder called '**testing-tagging**'. Unfortunately this did not function when it was implemented. The following tutorials were followed to create this tagging system: [Part 1](https://www.youtube.com/watch?v=qLbrkSeBan4) | [Part 2](https://www.youtube.com/watch?v=lzgJNNtgvPw) | [Part 3](https://www.youtube.com/watch?v=ug4RMo131fg) | [Part 4](https://www.youtube.com/watch?v=drzKh9fuXgE) | [Part 5](https://www.youtube.com/watch?v=CqwbvjONHJA).

# 1.4 PHP-Login

[PHP-Login](https://github.com/fethica/PHP-Login) is the sign in system using PHP and Ajax. This involves PHP being used with Bootstrap HTML and CSS to allow the player to sign in and sign out. The PHP connects to the '**user**' table of the '**sharks**' database.

Ajax involves updating the page without the page having the need to reload. Ajax was used because it came with the PHP-Login system which was used in this project. There is no need to know Ajax for this project.

This system also shows a confirmation page to the player when they have signed in successfully. The sign out does not appear on that page. If the player decides to click on a different page e.g. 'About', the 'sign out' button will not appear on that page or any other pages for that matter. The 'sign out' button can only reappear again if the player decides to click on the 'sign in' page, which the site will then prompt them to sign out.

This system also involved using a JavaScript file which checked the player credentials and gave an error message if the credentials were incorrect or involved an error.

III. Code Location

# 1. Unix Login

Username: **root**

Password: **sharks**

# 2. Locating the code

1. Click on *Activities* which is situated on the top left hand side of the screen.
2. This will open up a panel of icons.
3. Open *Files* which is the fourth icon on the panel.
4. Click on *Computer*.
5. Follow: *srv/www/htdocs/SharkGameProject.*

The folder *SharkGameProject* contains all the code and anything extra for this project. Alternatively the full code is also located on: <https://github.com/sjbarlas/SharkGameProject> and can be downloaded by clicking on “Download ZIP”.

Opening up *Sublime Text* from the panel, which is the third icon, should already open up the *SharkGameProject* contents and the source code.

The folder *MAIN* in the *SharkGameProject* contains all the primary source code needed for this project.

# 3. Folder Contents

The contents of each folder shall be examined individually. A standard folder tree can be viewed here: [https://github.com/sjbarlas/SharkGameProject#contents](https://github.com/sjbarlas/SharkGameProject" \l "contents)

# **3.1 SharkGameProject/**

* README.md
* Saira Barlas - Presentation.ppt
* SharkGameProject\_Documentation.pdf
* SharkSpecies.txt

SharkSpecies contains the 14 shark species that should be included in the Shark Guide page.

The other files are there for informative purposes.

SharkGameProject also contains the folders for the whole project which are examined below.

# **3.2 Bootstrap-3.3.5/**

3.2.1 css/

* bootstrap.css
* bootstrap.css.map
* bootstrap.min.css
* bootstrap.min.css.map
* bootstrap-theme.css
* bootstrap-theme.css.map
* bootstrap-theme.min.css
* bootstrap-theme.min.css.map

bootstrap.css is the file that contains all of the css code for bootstrap. bootstrap-theme.css is for bootstrap themes but this file is not needed for this project. Other files are either .map or .min files which again are not needed. It is better to use .css files rather than .min so the code can be later observed for error and edit purposes. It is NOT recommended to edit any bootstrap files, instead a new .css file should be created for any new code.

3.2.2 fonts/

* glyphicons-halflings-regular.eot
* glyphicons-halflings-regular.svg
* glyphicons-halflings-regular.ttf
* glyphicons-halflings-regular.woff
* glyphicons-halflings-regular.woff2js

All the font files that come with bootstrap. These have not been touched throughout the development of the project.

3.2.3 js/

* bootstrap.js
* bootstrap.min.js

The .js and the .min files. .js file should be used.

# **3.3 Database/**

* DB-info.txt
* README.md
* sharks.mwb
* sharks.sql
* sharks2.sql
* sharksERD.png

sharks.sql is the primary file used to create the '**sharks**' database in phpMyAdmin for the LAMP stack.

sharks2.sql was created when sharksERD was used to generate the SQL in [MySQL Workbench](https://www.mysql.com/products/workbench/). sharks2.sql should not be used for this project. It is there for informative purposes.

sharks.mwb is the MySQL Workbench file which was used to create the ERD. Do not use this .mwb file to generate the SQL since it will not be compatible with the LAMP stack. It is compatible with the XAMPP stack.

**All in all, sharks.sql is the file that should be used for the duration of the development!**

DB-info.txt contains information on float. This information was kept in regards to coordinates and how to set up the database in regards to them. This file has been kept for future informative purposes.

# **3.4 Jcrop-0.9.12/**

* MIT-LICENSE.txt
* README.md
* Index.html

Index.html contains the code for the site by the original author.

3.4.1 css/

* Jcrop.gif
* jquery.Jcrop.css
* jquery.Jcrop.min.css

Jcrop.gif creates the animated borders for the crop box. The other 2 files are neede for Jcrop.

3.4.2 demos/

* crop.php
* non-image.html
* styling.html
* tutorial1.html
* tutorial2.html
* tutorial3.html
* tutorial4.html
* tutorial5.html
* demo\_files/
  + demos.css
  + main.css
  + pool.jpg
  + sago.jpg
  + sagomod.jpg
  + sagomod.png

These files should aid in how to start with Jcrop and displays the functionality Jcrop offers.

3.4.3 js/

* jquery.Jcrop.js
* jquery.Jcrop.min.js
* jquery.color.js
* jquery.min.js

These are the .js files in relation to Jcrop. jquery.min.js is the original jQuery file needed in order for any Jcrop to work.

# **3.5 Jcrop-2.0.0/**

* Gruntfile.js
* MIT-LICENSE.txt
* README.md
* bower.json
* index.html
* package.json

This folder is similar to the other version but contains extra files if used with other software. These other files will never be needed in this project e.g. 'Gruntfile.js', 'bower.json', 'package.json'.

3.5.1 css/

* Jcrop.css
* Jcrop.gif
* Jcrop.min.css

The .css and .min files are similar v2.0.0 code. The .gif is contains a much updated crop box than the previous version.

3.5.2 demos/

* basic.html
* box-sizing.html
* circle.html
* coords.html
* crop.php
* index.html
* non-image.html
* panel.html
* thumbnail.html
* demo\_files/
  + demos.css
  + main.css
  + pool.jpg
  + sago.jpg
  + sagomod.jpg
  + sagomod.png

These are all tutorials and aids in how to get started with Jcrop. It has more functionality since it's the latest version.

3.5.3 js/

* Jcrop.js
* Jcrop.min.js
* jquery.color.js
* jquery.min.js

All the latest files for Jcrop and jquery.min.js being a newer version of jQuery.

# **3.6 MAIN/**

* README.md
* about.html
* check\_signin.php
* config.php
* game.html
* index.php
* leaderboard.html
* main.html
* main\_signin.php
* sguide.html
* signout.php

The main files that are used to generate the main pages of the site. main\_signin.php shows the form for the player to sign in. check\_signin.php connects the server and the database. config.php contain entries which are needed to connect to the database. index.php is the page that appears when the player has successfully signed in and portrays the message. main.html is seen after index.php. signout.php end the session when the player signs out.

3.6.1 bootstrap/

* css/
  + bootstrap-theme.css
  + bootstrap.css
* fonts/
  + glyphicons-halflings-regular.eot
  + glyphicons-halflings-regular.svg
  + glyphicons-halflings-regular.ttf
  + glyphicons-halflings-regular.woff
  + glyphicons-halflings-regular.woff2js
* js/
  + bootstrap.js
  + npm.js

All the files from bootstrap. npm.js is a file for Node.js which is not used in this project.

3.6.2 images/

* 1.jpg
* 2.jpg
* 3.jpg
* 4.jpg
* ajax-loader.gif

The temporary shark images that were used to test the crop provided by Brian Stone. ajax-loader.gif is the loading icon used when the player clicks the “sign in” button.

3.6.3 jcrop/

* gen-css/
  + demos.css
  + main.css
* jcrop-css/
  + Jcrop.gif
  + jquery.Jcrop.css
  + jquery.Jcrop.min.css
* jcrop-js/
  + jquery.Jcrop.js
  + query.Jcrop.min.js
  + jquery.color.js
  + jquery.min.js
* v2/
  + css/
    - Jcrop.css
    - Jcrop.gif
    - Jcrop.min.css
  + demo\_files/
    - demos.css
    - main.css

* + js/
    - Jcrop.js
    - Jcrop.min.js
    - jquery.color.js
    - jquery.min.js

The 'gen-css/', 'jcrop-css/' and 'jcrop-js' folders all contain code from v0.9.12 which were primarily used in this project. 'v2/' folder contains all the Jcrop code for the updated version which should be used. This folder was using in conjunction with the folder 'testing-cropping-with-Jcrop-v2.0/'.

3.6.4 sharkcss/

* about.css
* game.css
* index.css
* leaderboard.css
* main.css
* main\_signinBox.css
* main\_signinFooter.css
* sguide.css

The .css files for the main pages.

3.6.5 sharkjs/

* signin.js

This file checks if the player sign in credentials are correct or not and checks with the database.

3.6.6 testing-cropping-with-Jcrop-v2.0.0/

* gameTEST.html
* gameTEST2.html
* gameTESTANOTHER.html
* gameTESTANOTHER.php

This folder contains files which were being tested and experimented on using Jcrop 2.0.0.

3.6.7 testing-tagging/

* jquery.js
* tag.php
* taggingSystem.css
* taggingSystem.js
* taggingSystem.php
* taggingSystemConnect.php

These files were used for testing purposes when implementing the tagging system. The 'temptag' table was used with this folder.

# **3.7 PHP-Login/**

* README.md
* checklogin.php
* config.php
* index.php
* logout.php
* main\_login.php

checklogin.php connects to the server and selects the database. config.php has all the credentials which connect the site to the database. index.php displays a confirmation message when the player has successfully signed in. logout.php ends the session when the player signs out. main\_login.php is the page that enables the player to sign in while checking the sign in credentials.

3.7.1 css/

* bootstrap.css
* main.css

Bootstrap files for the system.

3.7.2 images/

* ajax-loader.gif
* screenshot.png

The 'loading' image which is used when the player presses enter.

3.7.3 js/

* bootstrap.js
* login.coffee
* login.js

The .js file for Bootstrap and .js file for the sign in system. .coffee file is never used throughout the project and will not need to be. It is there for informative purposes.