Krzysztof Stepien

540108 | 24.02.2019

The Power House Gym

Solution Planning Analysis

Table of Contents

[1 Use case diagrams and fully dressed use case descriptions 2](#_Toc7174827)

[1.1 Website default look and functionality 2](#_Toc7174828)

[1.2 Admins’ use case diagram and fully dressed descriptions 3](#_Toc7174829)

[1.2.1 Use Case 1: Log in 3](#_Toc7174830)

[1.2.2 Use Case 2 : Manage personal trainers 5](#_Toc7174831)

[1.2.3 Use Case 3 : Manage User Data 7](#_Toc7174832)

[1.2.4 Use Case 4: Log out 9](#_Toc7174833)

[2. Database design and entity relationship model 10](#_Toc7174834)

[2.1 Identifying Entities 10](#_Toc7174835)

[2.2 Identifying the attributes 10](#_Toc7174836)

[2.3 Initial relationship matrix 11](#_Toc7174837)

[2.4 Normalization 11](#_Toc7174838)

[2.5 Data that can change over time 12](#_Toc7174839)

[2.6 Arcs and Hierarchical relationships 12](#_Toc7174840)

[2.7 Entity Relationship model 12](#_Toc7174841)

[3 Static Model 13](#_Toc7174842)

[3.1 Textual analysis of the system elements 13](#_Toc7174843)

[3.2 Associations / collaborations 14](#_Toc7174844)

[3.3 User analysis 14](#_Toc7174845)

[4 Dynamic Model 14](#_Toc7174846)

[4.1 Sequence diagrams 15](#_Toc7174847)

[4.1.1 User Log in 15](#_Toc7174848)

[4.1.2 User Log out 16](#_Toc7174849)

[4.2 Activity diagram for the log in use case 17](#_Toc7174850)

[5 View Model and UI Design 18](#_Toc7174851)

[5.1 Navigation system and the tree diagram showing the site map 18](#_Toc7174852)

[5.2 Page Layout: Wireframe diagrams 18](#_Toc7174853)

[5.3 Screen resolution and responsiveness 21](#_Toc7174854)

[5.4 Style guide and accessibility 22](#_Toc7174855)

[5.4.1 Color scheme 23](#_Toc7174856)

[5.4.2 Font choices 23](#_Toc7174857)

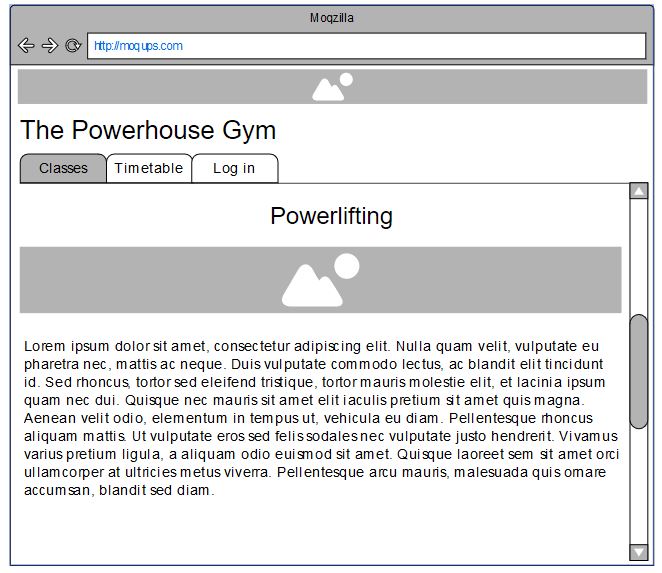
[5.4.3 Accessibility 24](#_Toc7174858)

[References 25](#_Toc7174859)

# 1 Use case diagrams and fully dressed use case descriptions

This section explains the use cases with detail, as previous document (Inception phase report- Graded Unit- K.Stepien) included only brief descriptions. Fully dressed descriptions describe the functionality of every use case with detail and are a base to build and structure the systems included in the project. I will also include initial wireframes for the specific pages of the website, to help understand the use case descriptions and visualise the functionality.

## 1.1 Website default look and functionality

At this stage of this document I would like to include the wireframe for the Home Page of the website, this page will include the basic functionality of the project that have to be handled first and the wireframe will help to understand the use cases described later on in this document.

Initial screen includes three tabs : Classes, Timetable and Log in. Classes and timetable display the information set by personal trainers and are visible to anyone who visits the website. Log in tabs functionality is designed with Admin and personal trainers in mind. It allows to access the restricted normally pages of the website. Admins’ functionality is explained in the sections below.

## 1.2 Admins’ use case diagram and fully dressed descriptions



Figure 0‑1 Admins' Use Case model

### 1.2.1 Use Case 1: Log in

**Primary Actor(s)**: Administrator, Database

**Description**: Main page of the website will present Administrator the option to access the logging screen. After clicking the log in button admin will see 2 input boxes – username and password and button allowing to submit the input. After entering the data ,all the input is validated and when successful admin is redirected back to main screen, but now with access to additional tabs – Management Centre and log out.

**Pre-conditions**: Input is validated and accepted.

**Post-conditions**: Admin is redirected to the home screen with additional submenus mentioned in the description

**Trigger event:** Admin clicks the button to log in.

***Normal Flow***

1. Admin enters his login into the username input box
2. Admin enters the password into the password input box
3. Admin clicks the Submit button
4. Admin is redirected back to home page with additional submenus available

***Extension Flows***

3a. The input is not a valid Username

1. Display the message explaining the error : “Invalid username. Please try again”
2. Username input box is cleared and cursor is positioned in the Username input box

3b. The input is not a valid Password

1. Display the message explaining the error : “Invalid password. Please try again”
2. Password input box is cleared and cursor is positioned in the Password input box.

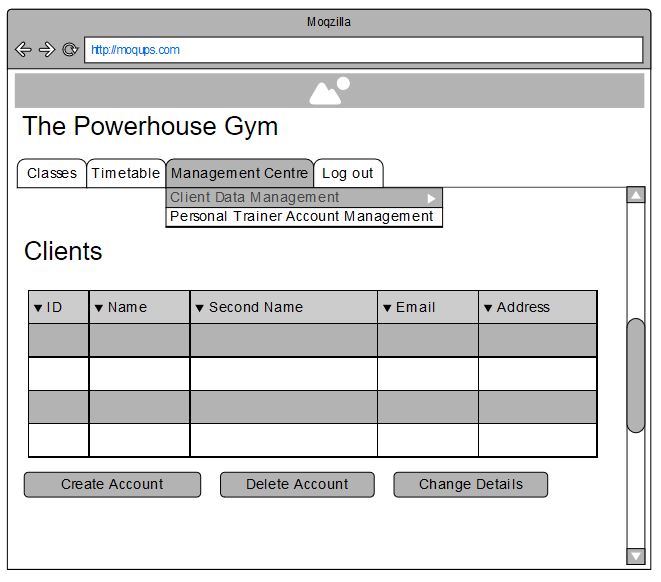


Figure 1-0‑2 Admin Management Screen

This screen shows how the Client Data Management page and Personal Trainer Account Management will look like. Their functionality is similar, the only difference between those two screens is the data displayed in the list

### 1.2.2 Use Case 2 : Manage personal trainers

**Primary Actor(s)**: Administrator, Database

**Description**: After logging in Admin have access to Management Centre submenu (Figure 1-2), which after expanding on will allow to access the Personal Trainers Account Management. This opens a new page that displays the personal trainers details and allows to create the account , delete the account and change details via the buttons located on the top of the page.

- Create the account button displays the form with input boxes for name, second name, address and email address of personal trainers. On the bottom of the form there is a Submit button which will save the input to the database.

- After clicking on the personal trainer on the list and choosing to delete the account admin will see the confirm dialog validating the decision. Success will erase the chosen data from the database

-After clicking on the personal trainer on the list and choosing Change the details button admin will be able to change personal trainers details using the form with input boxes and submit button.

**Pre-conditions**: Admin have logged in correctly and Main page displays the Management Centre and Log out tabs in the menu.

**Post-conditions**: Correctly displaying the Personal Trainers Account Management page.

**Trigger event:** Admin chooses the Personal Trainers Account Management from the drop down menu.

***Normal Flow***

1. Admin chooses Personal Trainers Account Management from the drop down menu located in the main page.
2. Website displays the management page with the list of Personal Trainers and buttons : Create account , Delete account, Change details
3. Clicking Create Account displays the form with input boxes mentioned above, the Submit button and Cancel button
4. Clicking Delete Account with one of the trainers highlighted in the list will delete the account info from the database.
5. Choosing Change Details displays additional form with input boxes , Submit button and Cancel button.

***Extension Flows***

1. The list of trainers cannot be accessed from the database
2. Displays the error message “Couldn’t’ access the data, please refresh the page.”

3a. Invalid input for any of the boxes after clicking the Submit button

1. Displays the error message ”Please enter the valid input”
2. Border of the input box with invalid input changes colour to red

3b. If input is validated and accepted by the system the message is displayed: “Success! Account created.”

3c. If PT already exists in the database system displays the message: “Personal Trainer ‘name’ already exists in the database”.

3d. Clicking the Cancel button displays the message : “Are you sure you want to cancel?”

4a. After clicking the Delete button without choosing the personal trainer from the list, system displays the error message : “Please choose the account to delete!”

4b. Choosing the account and pressing the Delete button displays the Yes/No message : “Are you sure you would like to delete this account?”

5a. If personal trainer is not chosen from the list and Change Details button is clicked, error message is displayed: “Please choose the personal trainer from the drop down menu first.”

5b. Invalid input for any of the boxes after clicking the Submit button

1. Displays the error message ”Please enter the valid input”
2. Border of the input box with invalid input changes colour to red

5c. Clicking the Cancel button displays the message : “Are you sure you want to cancel?”

### 1.2.3 Use Case 3 : Manage User Data

**Primary Actor(s)**: Administrator, Database

**Description**: Just like with the Personal Trainer use case after logging in Admin have access to Management Centre submenu, which after expanding will allow to access the Manage User Data tab. This opens a new page with the list of users including their name, second name, id , address and email address. This page also includes buttons Add New User , Delete Account and Change User Details.

- Create the account button displays the form with input boxes for name, second name, address and email address for the new user. On the bottom of the form there is a Submit button which will save the input to the database and Cancel button that will go back to the User Management .

- Deleting the account will allow admin to choose the User from the list and press delete button which on success will erase the chosen data from the database

- To change the details of the User admin have to highlight the user on the list and then press the Change Details button. This will display the form with input boxes and Submit, Cancel buttons.

**Pre-conditions**: Admin successfully logged in to the website and Main page displays the Management Centre and Log out tabs in the menu.

**Post-conditions**: Correctly displaying the User Data Management page.

**Trigger event:** Admin chooses the User Data Management from the drop down menu.

***Normal Flow***

1. Admin chooses User Data Management from the drop down menu located in the main page.
2. Website displays the management page with the list of Users and buttons : Create account , Delete account, Change details
3. Clicking Create Account displays the form with input boxes mentioned above, the Submit button and Cancel button
4. Clicking Delete Account with one of the Users highlighted in the list will delete the account info from the database.
5. Choosing Change Details displays additional form with input boxes , Submit button and Cancel button.

***Extension Flows***

1. The list of Users cannot be accessed from the database
2. Displays the error message “Couldn’t’ access the data, please refresh the page.”

3a. Invalid input for any of the boxes after clicking the Submit button

1. Displays the error message ”Please enter the valid input”
2. Border of the input box with invalid input changes colour to red

3b. If input is validated and accepted by the system the message is displayed: “Success! Account created.”

3c. If Customer/ User already exists in the database system displays the message: “User ‘name’ already exists in the database”.

3d. Clicking the Cancel button displays the message : “Are you sure you want to cancel?”

4a. After clicking the Delete button without choosing the personal trainer from the list, system displays the error message : “Please choose the account to delete!”

4b. Choosing the account and pressing the Delete button displays the Yes/No message : “Are you sure you would like to delete this account?”

5a. If Customer is not chosen from the list and Change Details button is clicked, error message is displayed: “Please choose the Customer from the drop down menu first.”

5b. Invalid input for any of the boxes after clicking the Submit button

a.Displays the error message ”Please enter the valid input”

b.Border of the input box with invalid input changes colour to red

5c. Clicking the Cancel button displays the message : “Are you sure you want to cancel?”

### 1.2.4 Use Case 4: Log out

**Primary Actor(s)**: Administrator, Database

**Description**: Logging out from the system will display the Main page visible for everyone, Log out and Management Centre tabs will not be visible, instead the Log in tab is displayed.

**Pre-conditions**: Admin Logs in to the system

**Post-conditions**: Admin is redirected to the home screen without access to management system, have to Log in again to access it.

**Trigger event:** Admin clicks the button to log out.

***Normal Flow***

1. Admin clicks the button to Log out
2. System loads the main page of the website.

***Extension Flows***

1. System displays the message Yes/No: “Are you sure you would like to log out?”
2. On clicking yes – system loads the default main page
3. On clicking no – no action, message box disappears

# 2. Database design and entity relationship model

## 2.1 Identifying Entities

Entities are the objects that information about have to be stored in the database. I have analysed the brief during the Inception Planning (document called ”Inception phase report- Graded Unit- K.Stepien”). Entities in the database are represented as columns or tables.

Entities that will require the data stored in the data base are: **Customer**, **Personal** **Trainer**, **Admin**, **Classes**, **Exercises, Reports**.

## 2.2 Identifying the attributes

This section identify the attributes of the entities, this will be the data stored in the database. Some of the attributes might be used as a unique identifiers of the entity, I will underline the candidate unique identifiers. If the entity won’t have the possible natural identifiers the artificial unique ID will be produced.

**Customer** – email, first name, second name, phone number, address, measurements and results of the exercises

**Administrator** – email , first name, last name, phone number , address

**Personal Trainer** - email , first name, last name, phone number , address, class type

**Classes** – type of the class, timetable

**Exercises –** description, which class they belong to

**Reports** – Personal trainer notes, data included under the previous entities and calculations (for example progress).

**Address** is shared by three entities in the system and consists of many strings, therefore it should be stored as separate entity.

## 2.3 Initial relationship matrix

This table is a representation of what the data relationships in the database might look like.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Customer** | **Administrator** | **Personal Trainer** | **Classes** | **Exercises** | **Reports** | **Address** |
| **Customer** |  |  | Is trained by | Attends | Does | Receives | Lives at |
| **Administrator** |  |  | Manages data of |  |  |  | Lives at |
| **Personal Trainer** | Trains |  |  | Designs | Designs | Produces | Lives at |
| **Classes** | Are attended by |  | Are designed by |  | Include |  |  |
| **Exercises** | Are attended by |  | Are taken care by | Are included in |  |  |  |
| **Reports** | Are produced for |  | Are produced by | Are based on performance in | Include data about |  |  |
| **Address** | Address of | Address of | Address of |  |  |  |  |

## 2.4 Normalization

Normalization is the process included into the design of the database, it’s point is to prevent the unnecessary redundancy in the database. There are three main rules that apply to this process :

In **the first normal form**, only single values are permitted at the intersection of each row and column; there shouldn’t be any repeating groups stored.

To normalize a relation that contains a repeating group, the repeating group should be removed and two new relations formed.

The unique key of the new relation is a combination of the unique key of the original relation plus an attribute from the newly created relation for unique identification.

***Second normal form*** states that all the attributes have to be fully dependant on the entity and its unique key, if any of the attributes can be a single entity, it should be stored in the separate table in the database.

***Third normal form*** states that no non-unique ID attribute can be dependent on another non-unique ID attribute. Each attributes must depend directly on the unique ID. All attributes that are not dependent on the UID must be removed. For example, attributes that can be derived from data contained in other fields and tables must be eliminated.

## 2.5 Data that can change over time

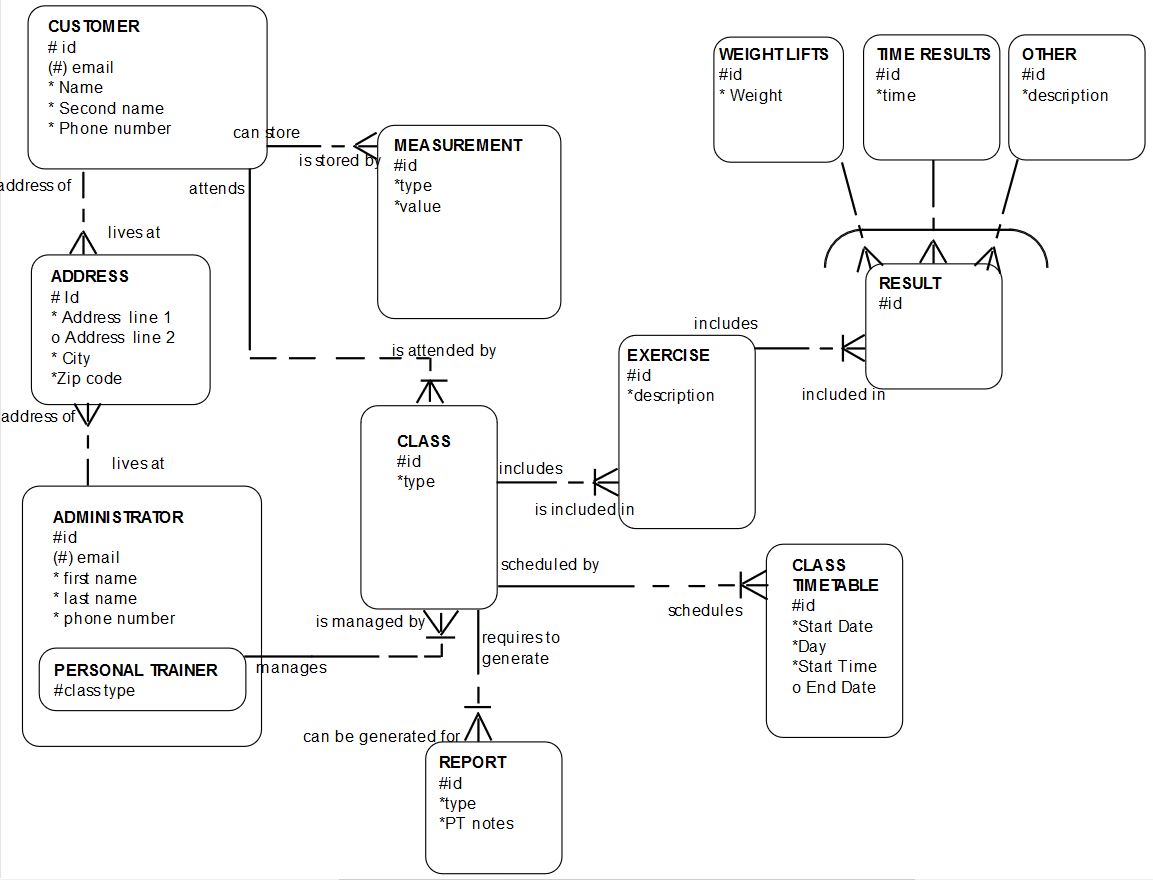
Only example of the data changing over time are class timetables. Keeping the normal forms in mind I will create another entity to store the data related to that issue.

## 2.6 Arcs and Hierarchical relationships

Arcs in ERD are used to represent when to indicate that two or more relationships are mutually exclusive (XOR). In this project only entity that can be represented in that way is **Result**. Results might be stored as an integer for weight, as timed result or just overall description of the progress.

Hierarchical relationship will be represented with **Administrator** and **Personal** **Trainer** relationships. Those entities are both the employees of The Power House gym and share similar attributes, the only difference is a class type of the personal trainer.

## 2.7 Entity Relationship model



# 3 Static Model

Static model’s purpose is to show the main components of the system and their relationships – the logical structure of the system. It is usually achieved using the class diagrams that highlight the software components and how they relate to each other, methods and classes in the object oriented system.

This system however will be developed using the PHP language, which is hard to describe using the standard methods and diagrams. I will describe the structure using textual analysis, it will include different PHP files that will be building the system, Java Script files and CSS style sheets.

## 3.1 Textual analysis of the system elements

|  |  |
| --- | --- |
| **File** | **Description of the purpose** |
| index.php | Displaying the home page of the system. This is a base of the system that will load initially when user enters the website address in the browser |
| classes.php | Displays the classes descriptions |
| timetable.php | Displays the timetable of the classes. |
| login.php | Displays the log in page |
| clientDataManagementCentre.php | Displays the User data management centre (see section 1.2.3 on page 7) |
| personalTrainerManagement.php | Displays the PT management centre (section 1.2.2 on page 5) |
| pageElements.php | This file will contain the code that will be displayed repeatedly on the pages. Elements like menu or footer will be contained in this php file and enclosed in functions in order to easily display the text that repeats on every page or for example menu that have to be different when user is logged in or logged out.  functions :  displayMenu() – will display menu with dynamic elements depending on the state of the website or if user is logged in/ logged out  writeCommonStyles() – this function will contain all the links to the stylesheets used in the website  displayFooter() – displays the footer that will stay static throughout the website |
| session.php | Manages access to the site through a secure session. If the user is not logged in the system will not allow to access the Management Centre. |
| database.php | Manages the connection to the database.  Functions:  connectToDB() – establishes connection to the database used by the system  closeConnection() – closes the connection to the DB  sanitazeString() – trims the Strings so it is safe to use in the sql query. |
| processLogIn.php | Manages the log in functionality |
| processLogOut.php | Manages the log out functionality (section 1.2.4 page 9) |
| changeDetails.php | Displays the form that will cover the functionality to update the details of the user or personal trainer |
| delete.php | Functionality to delete the personal trainer or user |
| createNew.php | Functionality of adding the new user to the database |
| validateForm.js | Initial client side validation of the user input |
| Menu.css | Style sheets for the menu |
| Responsive.css | Style rules that will provide the functionality for the responsiveness of the system. (see section 5.3) |
| Style.css | Style rules for the overall look of the website |

## 3.2 Associations / collaborations

The system can be divided in two sections based on functionality – section that is responsible for displaying the data on the screen and section that provides support for the first section. Files : index.php, classes.php, timetable.php, login.hp, clientDataManagementCentre.php and personalTrainerManagement.php are displaying the text and generating the HTML that will be used by the end user of the system. All of those files will require style sheets menu, responsive and style CSS files, form validation from validateForm.js Java Script file and pageElements.php file to display the menu and elements of the different pages. Also session and database files are used on every page to provide necessary security and functionality. Other files are used for specific use cases in the system.

## 3.3 User analysis

Users that will be using the website system are mentioned in the use case descriptions (see Document : Inception phase report- Graded Unit- K.Stepien page13 part 1.1.5). It is a set amount of users – personal trainers and an administrator – employees of the Power House gym. They all use Windows based personal computers or Android smartphones to interact with the website.

# 4 Dynamic Model

Dynamic model of the system shows how the different parts of the system interact with each other and how the data is passed between the different modules. To achieve clear view of the system activities few kinds of diagrams are used: for example sequence, communication and activity diagrams.

## 4.1 Sequence diagrams

Sequence diagrams show how the data is transferred between different components over time. This stage of the design shows the general interaction of different pages and functions.

### 4.1.1 User Log in

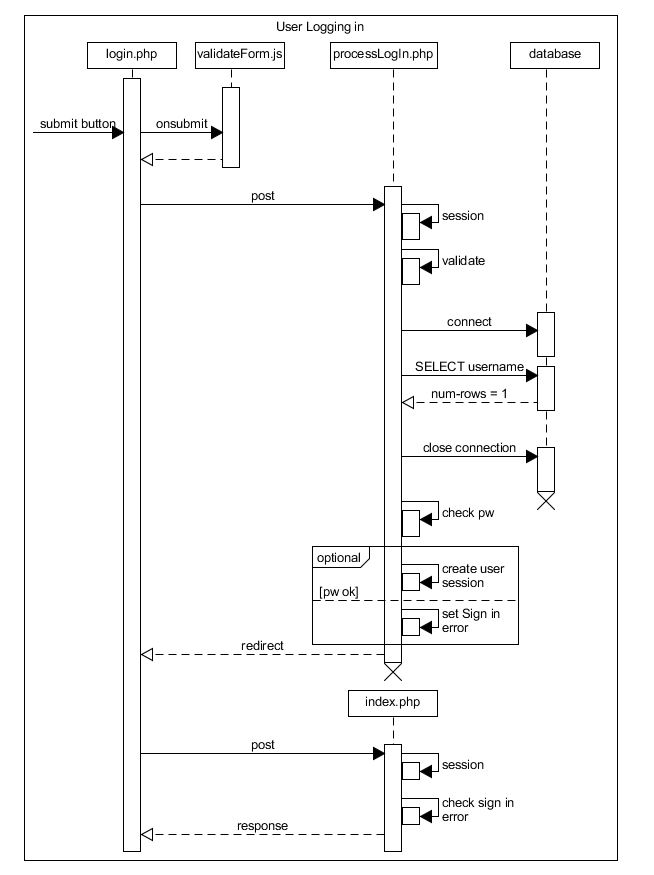


Figure 4‑ user sign in sequence diagram

This sequence is triggered by user entering the data into the input boxes on the log in page and pressing the submit button. On submit the data is initially validated on the client side by the validateForm.js script. Next the data is processed by processLogIn.php, on server validation is triggered and connection to db is opened. Script checks if the username entered is present in the database and closes the connection. Next password check, if valid the new user session is created, if not the error is displayed. If there were no errors encountered, user is redirected to the main screen of the page with new session started and logged in.

### 4.1.2 User Log out

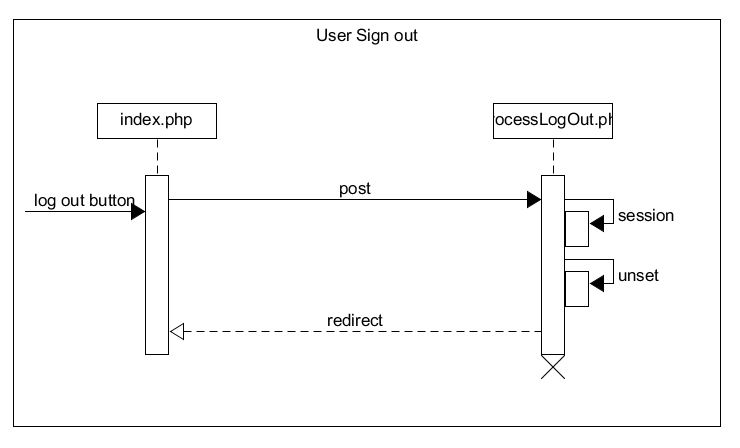


Figure 4‑ user log out sequence diagram

Log out use case is triggered when user clicks on the Log out button from any screen( it is available when user is logged in). Session is unset and destroyed, and user is redirected to the index page with no session set.

## 4.2 Activity diagram for the log in use case

This section represents the activity diagram for the Log in use case. It represents the flow of activities that this use case consists of. The flow of operations can be broken and redirected back to start if the validation is not successful, it can be either when validating through JavaScript on client side, on the server, or when the connection to the database is unsuccessful. When all the validation processes pass, the session is created for the user and he/she is redirected back to the index page with new menu and functionality available.

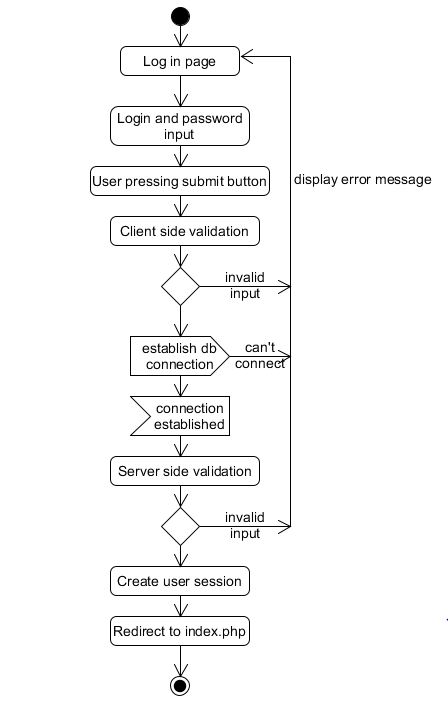


Figure 4-3 Log in activity diagram

# 5 View Model and UI Design

## 5.1 Navigation system and the tree diagram showing the site map

Every page will have a horizontal menu bar in the top left of the page, under the main banner and logo allowing the user to navigate through different subpages of the site. This menu includes the Classes, Timetable, Log in and Management centre drop down menu available after user is logged in. Also after logging in the option to log out is visible. Drop down menu allows the user to navigate to the My details page, Client Data management Centre and Personal Trainer Account Management which is available to view only for the administrator of the page. Tree diagram below (Figure 5-1) represents the layout of the pages.

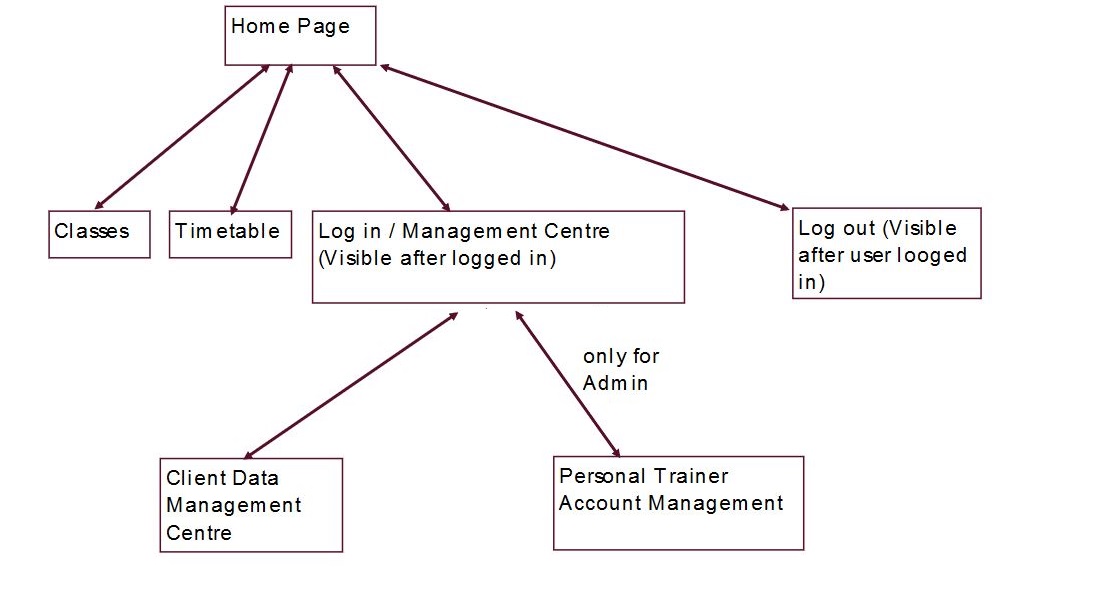


Figure 5-1 site map tree diagram

## 5.2 Page Layout: Wireframe diagrams

Wireframe diagrams are representing a visual guide to the framework of the website. Their purpose is to depict the pages layout, how the pages function together and how the website content is arranged. I have included some of the wireframes in the solution plan document for ease of the use cases explanations, but I will include them in this section for the clarity of the documentation. Wireframes were created using the moqups.com website. (moqups, 2019)

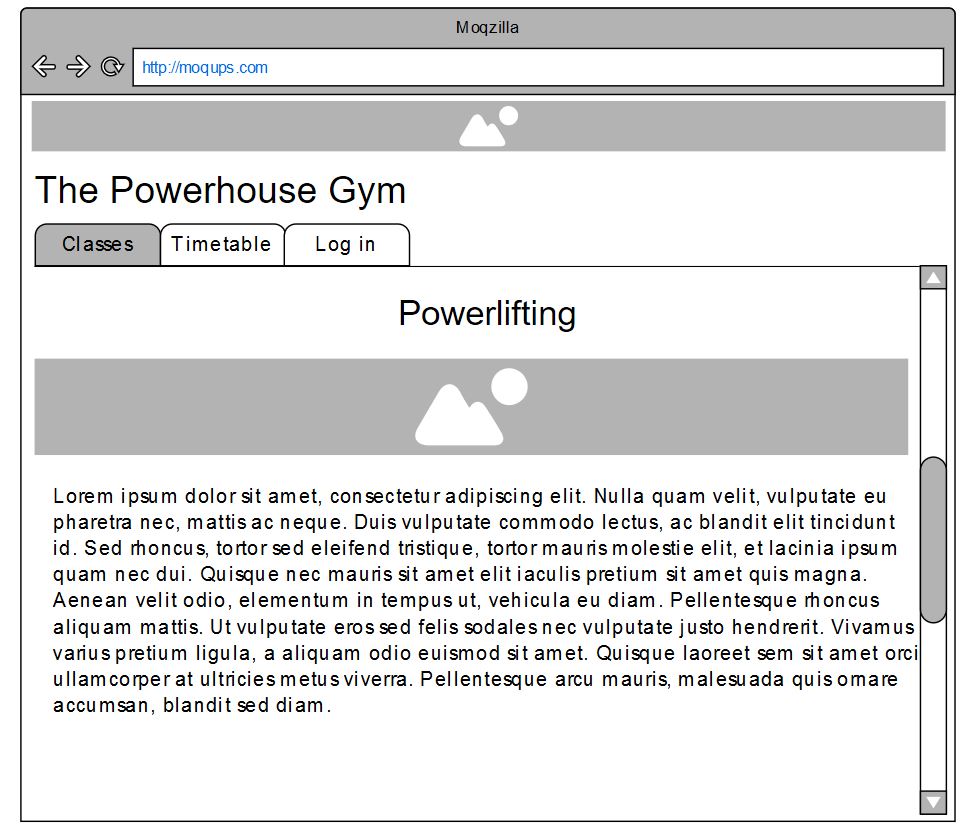


Figure 5-2 Home screen

Home screen of the website will allow the user to access the sub-pages – classes, timetable and log in. If user is logged in another menu is displayed (Figure 5-2).

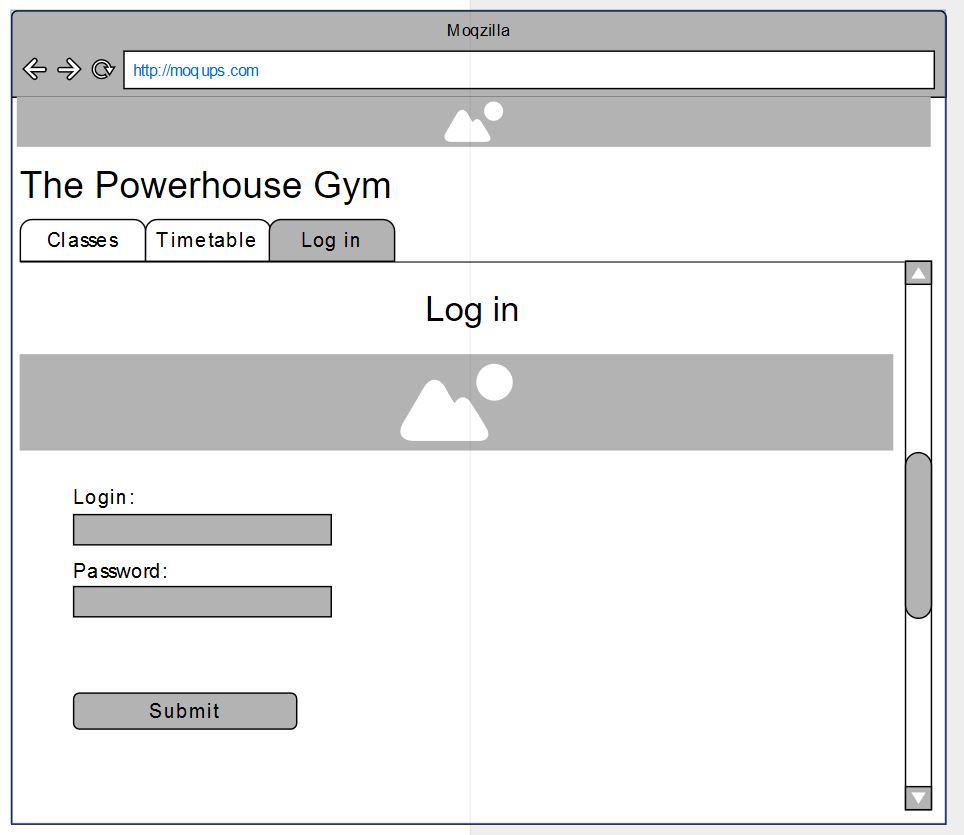


Figure 5-3 user log in screen

After clicking the ‘Log in’ tab user is redirected to the login page which displays the form with input boxes and submit button. Data captured from those input boxes will be validated locally and on the server side and will be used to create the session for the user. This data represents the fields from the database – username and password.

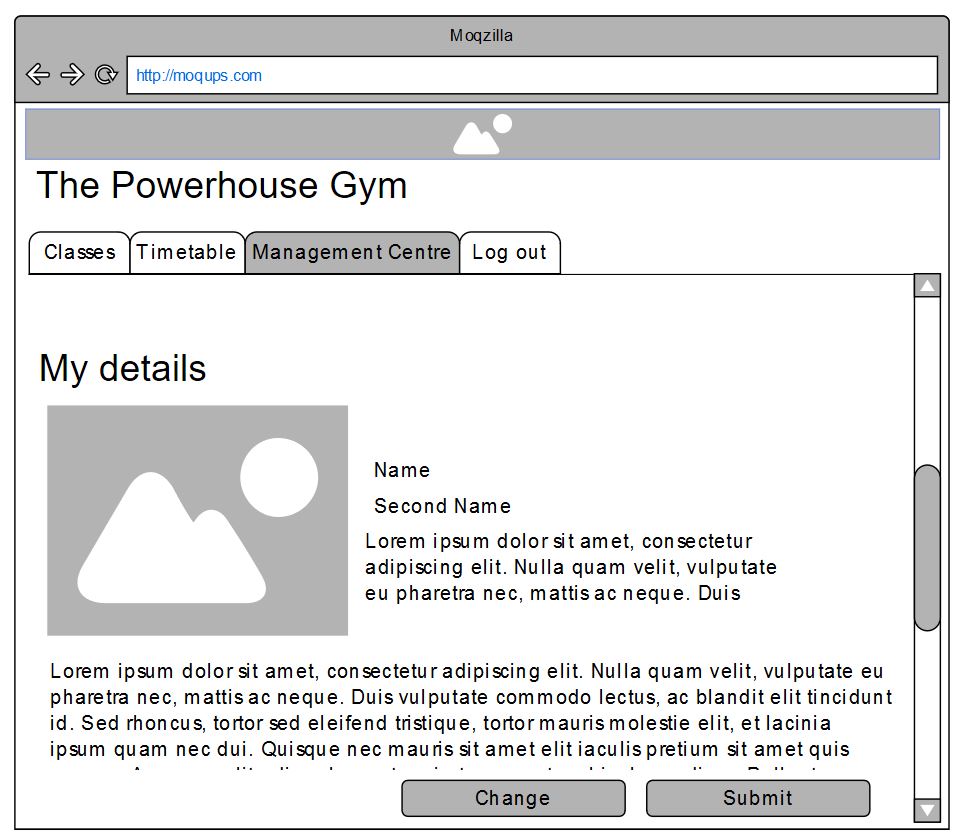


Figure 5-4 User logged in

After logging in user is presented with additional options in the menu. This wireframe represents the ‘My Details ’ page accessible from Management Centre Screen.

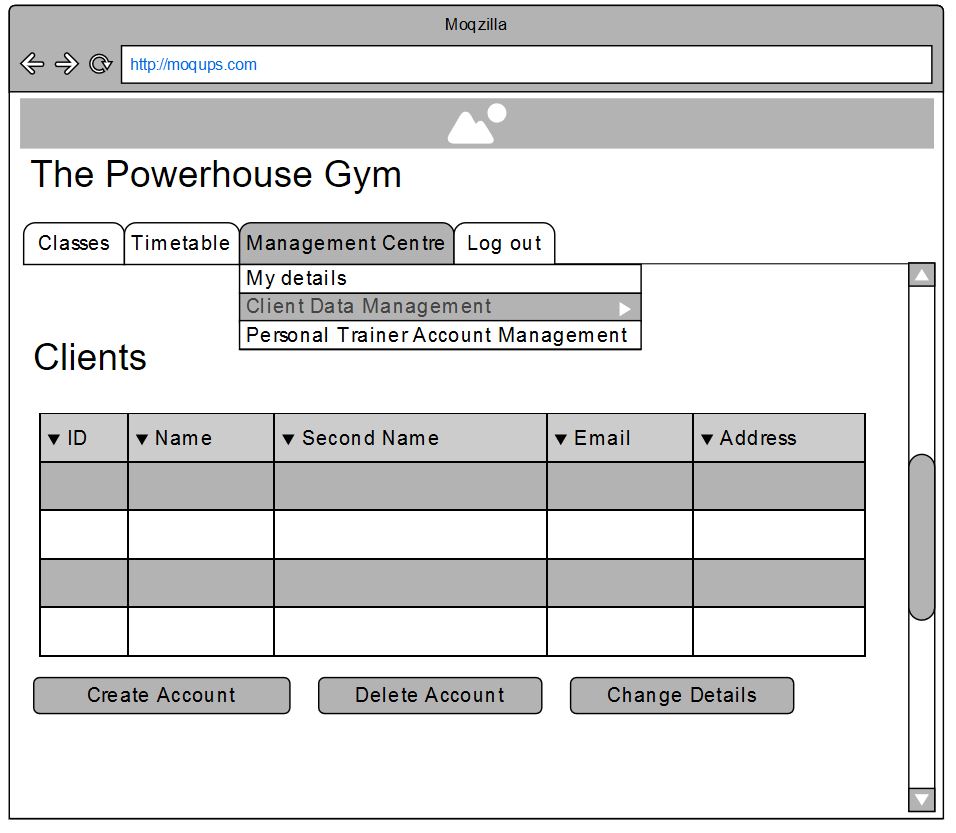


Figure 5-5 Management Centre screen

Figure 5-5 represents the Management Centre page wireframe. While hovering the mouse cursor over this tab additional drop down menu is displayed allowing the user to choose the subpages related to management centre of the system. This wireframe is detailing how the Clients management page will look like. The table in the middle will display the client details and buttons in the bottom of the page will allow the user to access the functionality of the system.

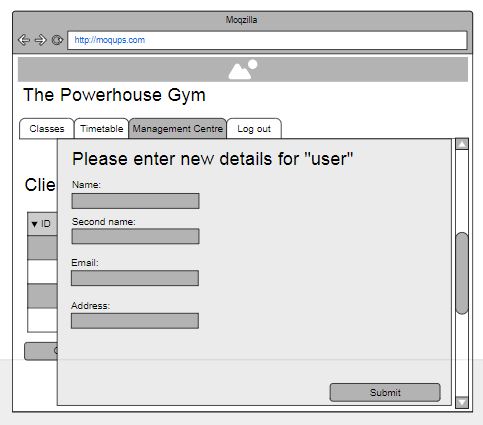


Figure -6 input form

Last wireframe – Figure 5-6 represents the look of the input form that will be used to input the details for the new user or update the details of the existing user. Input boxes from this form will update or insert the data to the database fields in the table users – name, second\_name, email, address etc .

## 5.3 Screen resolution and responsiveness

Site is designed with the responsiveness in mind. Changing the screen resolution will also change the how the elements appear on the page. Menu will be hidden (Figure 5-6) and instead there will be an interactive ‘hamburger’ icon that after clicking will display the menu elements .All the elements will scale with the resolution changes. This technique is necessary as many of the users might want to view and interact with the page via the smartphone or tablet which offer smaller screen sizes. All of the features mentioned above will be developed using CSS and Java Script.

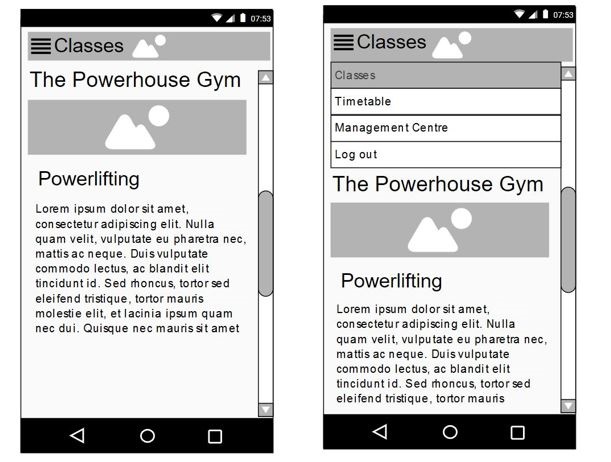


Figure 5‑‑ Left: menu hidden / right: menu extended

## 5.4 Style guide and accessibility

This section is dedicated for style of the system. Font, colour scheme and accessibility information. All the color and accessibility information is designed using the palleton.com website. (palleton, 2019). I have tried to keep the interface simple, purposeful and consistent throughout the website, keeping in mind the principles that are supposed to be used in the UI design. (usability.gov, 2019)

### 5.4.1 Color scheme



Figure 5‑‑ Example of the color scheme

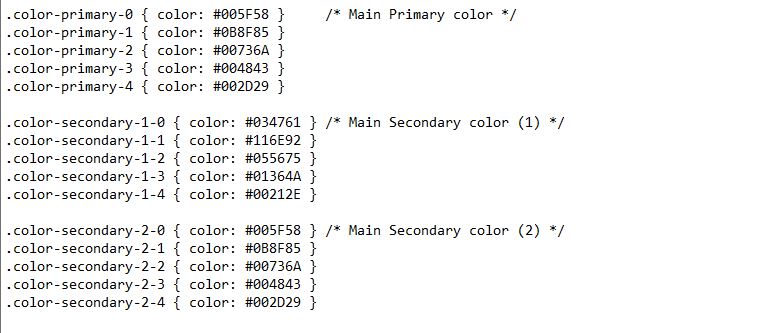
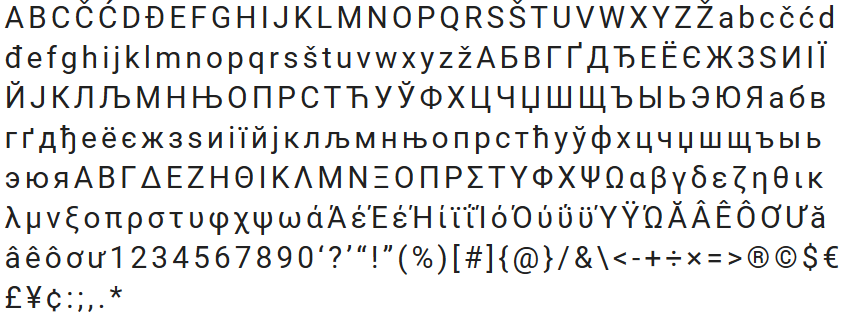


Figure 5‑‑ Website colors in hex codes CSS

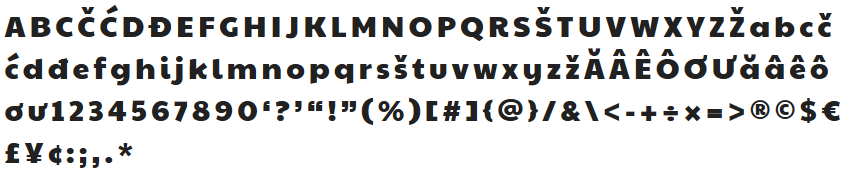
### 5.4.2 Font choices

Main font used on the website will be a standard Roboto font, however headings will be displayed using Paytone One font.

Example of Roboto:



And the Paytone One font:



### 5.4.3 Accessibility

Palleton website offers the simulation that allows to preview how the pages will look like for the colorblind user. There are many color blindness disabilities, I have chosen the most common cases:



Figure 5‑‑ Deuteranomaly (5% of men and 0.4% women affected)



Figure 5‑‑ Achromatopsia (users unable to see colors)

# References

moqups, 2019. [Online]   
Available at: https://moqups.com/

palleton, 2019. [Online]   
Available at: http://paletton.com

usability.gov; Web UI principles, 2019 [Online]   
Available at: https://www.usability.gov/what-and-why/user-interface-design.html