Mälardalen University
School of Innovation, Design and Engineering
Västerås, Sweden

Software Engineering 2 : Project Teamwork - 7.5 hp - DVA313

BLACK RIVER RUN TIME KEEPING

Design Document

Bastien Delbouys
bds18002@student.mdh.se

Zacharias Claesson
zcn16001@student.mdh.se

Sebastian Oveland
sod16003@student.mdh.se

Cécile Cayèré
cce18001@student.mdh.se

Mohammed Abuayyash
mah18005@student.mdh.se

Rikard Gestlöf
rgf16001@student.mdh.se

Johannes Sörman
jsn16009@student.mdh.se

December 6, 2018

Contents

1 Introduction

The aim of this document is to give a complete explanation of the project *Black River Run.* It will give all the necessary information to understand the whole project and development choices made by the working group.

Firstly, this document will give an overview of the project and its context. It will also help to learn more about the client and the end use of the project.
Then, it will answer the question of the design choices of the system in order to return to the client a product that fit his requirements.
Finally, the way the user can interact with the system across all available functions will be explained in detail.

2 Project

2.1 Client

The client of this project is Christoffer Holmstedt from the organisation Västerås Running Club. Some important information about the client are shown below :

- **Name :** Christoffer Holmstedt
- **Organization :** Västerås Running Club
- **Email :** christoffer.holmstedt@gmail.com
- **Phone number :** +46 (0)73 7816126

2.2 Context

Each year, Västerås Running Club organizes a long-distance race called Black River Run in Västerås. Runners have the choice to register to run 100 miles, 50 miles or 20 miles. These different distances consist of a defined number of laps of a 10-mile marked loop.
Before the race, each runner is given a time keeping device called SI-Unit that they will have to carry for the duration of the race.
Along the entire length of a lap, there are 4 stations where the SI-Units worn by the runner can connect remotely and transmit data quickly to indicate the position of the runner to the third party server called OLresult. This is called a virtual punch.
To be able to track how people are doing without attending the race, Västerås Running Club has a website Brrlive (*http://www.brrlive.eu*) where information about the race is displayed.

2.3 Goal

As the previous website is outdated, the goal of this project is to create a new web application that displays the results of the race in progress and shows an history of the previous race. Through this interface, an connected user (administrator) could manage all the data of the system (runner, race, si-unit, ...).

3 Research

The group plans to complete this part later, for the last version of the document. It should include the following content :

- Personas.
- Test on wireframes.
- Test on navigation UI.
- Target group information.
- UI test on target group.
- Test responsive.
- Standards for this kind of website.
- Colors.
- Client's thoughts about UI design.
- Implementations standards (camelCase, lower_case, etc...).
- Maybe analyze of similar websites.

4 System Design

The figure below shows an overview of the global system. It will be used through the entire section in order to get a better understanding of the system.

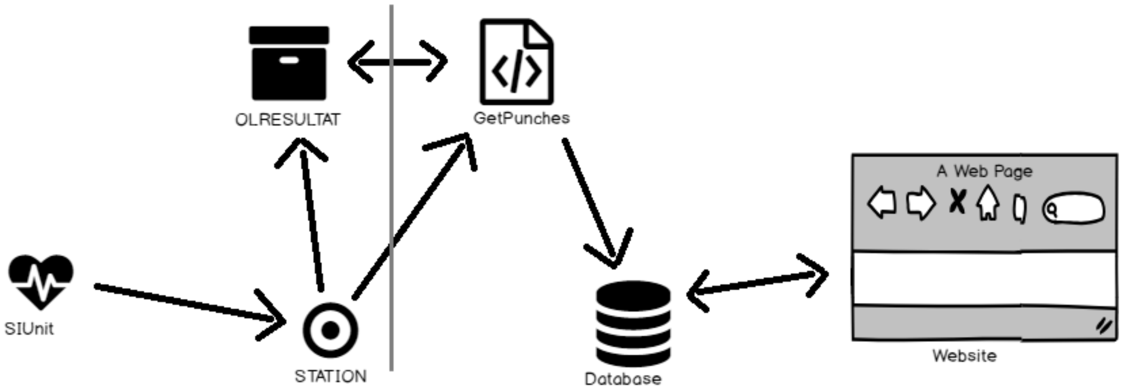


Figure 1: System Overview.

4.1 Main Parts Of The System

The complete system contains a few different parts that communicate with each other. One part is a SI-Unit that the runner carries throughout a race. This SI-Unit is an identifier for the runner. It connects to a station every time the runner passes within a specific distance from it.
Then, the station, which keeps track of which unit passes at which time, takes over.

The station saves a timestamp at the time of the passing, together with an identifier for the runner. It then tries to connect to a third-party web server and upload all stored values. This web server then makes the data available through a web API.

The system fetches this data from the API through a back-end script. This script is called upon after the station has successfully pushed its data to the API. When the data is fetched, it's parsed and formatted to be stored in a database.

The stored data is fetched upon request to the web-interface and in turn made available to the end users (everyone that visits the website to check results from the races).

The system handles everything from the API to the web-interface presented to the user that visits the website. This is solely a software-based solution that is hosted on a shared web hosting server, which means that the group only has the code to think about.

The parts that are developed in this project are the system that fetch data from the third-party API, the database that stores everything and the website visible to the end users.

4.2 Parts Collaboration

Black river run race time keeping, is a responsive system that contains many parts, these parts are illustrated in the figure below.

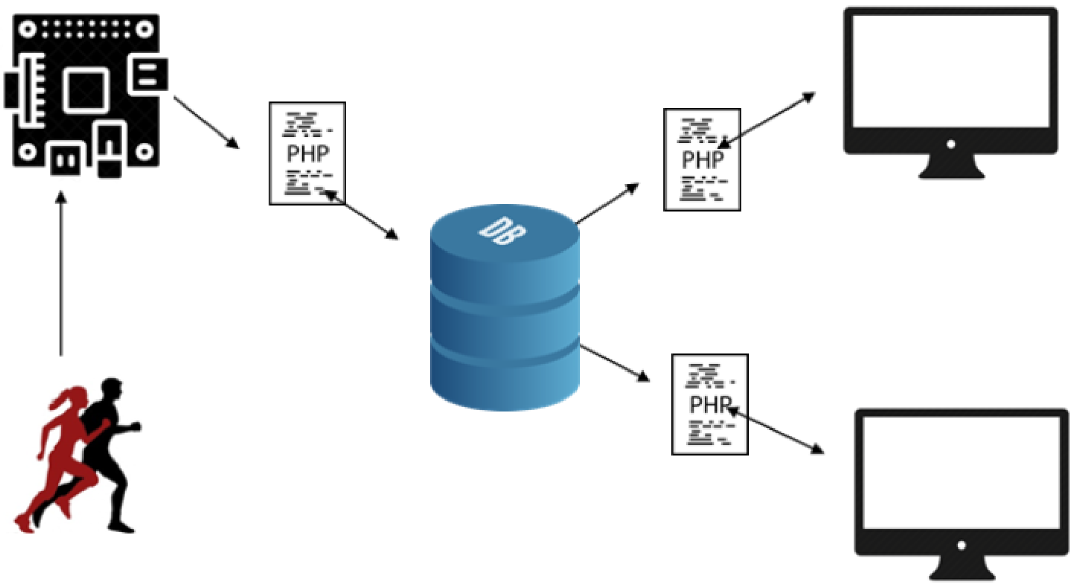


Figure 2: System Parts Collaboration.

All these parts are essential and important to provide the highest quality services. They are listed below :

- Administrator dashboard.
- Guest website.
- Database.
- Raspberry PI.
- Runners controller.

4.2.1 Runners, Raspberry PI And Database Collaboration

Each runner has a controller with a unique identifier. When the race starts the controller sends a signal to the Raspberry PI (stations). Each signal is directly converted to a string which is associated with the runner whose unit initiated it. The string is then retrieved by the PHP script which analyzes this string and extracts the readable data from it. The data are then stored in the database by calling pre-defined stored procedures that the database has.

4.2.2 Website And Database Collaboration

The Black River Run website shows information about the races and runners. The website also shows some other information and features that are illustrated in the requirements part, for this purpose the website use PHP functions to fetch information.

The PHP script calls pre-defined procedures to retrieve data from the database. The data is manipulated and formatted in order to be displayed on the website.

For example, when user want to view *Damer, 50 Miles* runners, the user can select this race from menu. The PHP script proceeds to directly call pre-defined procedures stored in the database, using the ID of *Damer, 50 Miles* element. After that the requested data will be shown on the website.

4.2.3 Administrator Dashboard And Database Collaboration

The Black River Run webpage for administrators has a similar set of functions like the guests, but with some extra functions. These extra functions allow, for example, adding, modifying and removing data.

The administrator page is applying the same methods as the guest website to retrieve data from the database.

For example, when the administrator wants to edit a specific runners name, the first step is to go to this runner's profile by selecting the runner from the runner list. After that, the administrator can select update mode which allow him/her to edit the runner data. At this point a php-function passes the new data to the back-end, which will call pre-defined procedures from the database. This function will be executed and then the administrator is informed with a success or fail message.

4.2.4 External System

The figure above shows the overall flow of how the system works. The external systems are the : Stations, SI-Unit and OLresult. Everything to the right of the gray line belongs to the part of the system that the group is developing, and the things to the left are the external systems. The getPunches box in figure is a PHP script that is located on the groups webserver (this will be the Västerås löparklub webserver when the system will be completed) and it is used to gather data from the third party server. This is the connection between the groups system and the external system. The stations are registered as Radio Online Control units (ROC) at the third party server OLresultat.

4.3 System Structure

4.3.1 Database

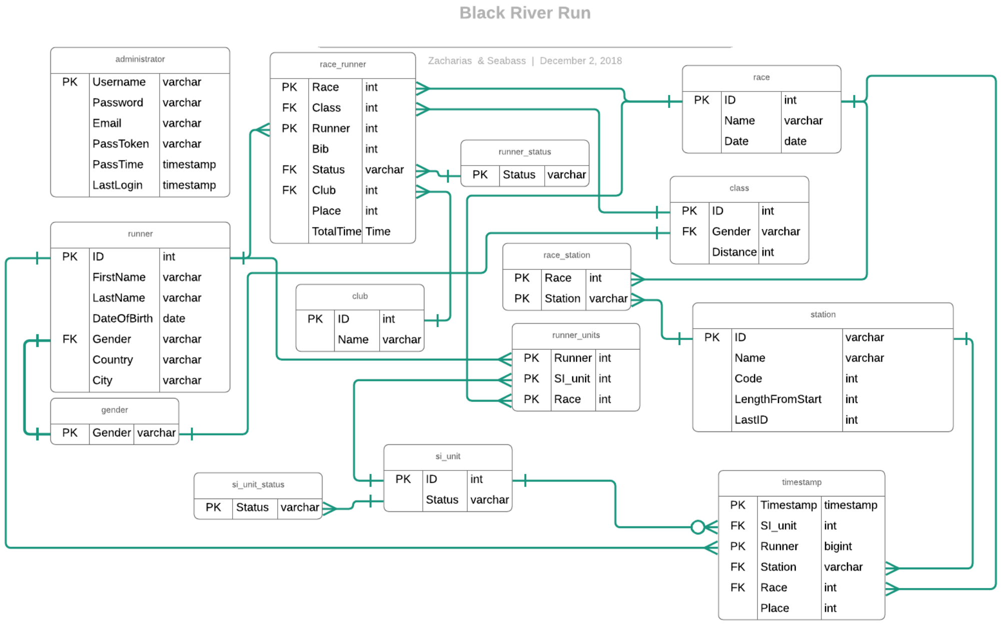


Figure 3: Database Diagram.

The database consists of 14 tables. These are :

- ADMINISTRATOR :
  - Username [varchar(50)], PRIMARY KEY.
  - Password [varchar(60)].
  - Email [varchar(100)].
  - PassToken [varchar(60)].
  - PassTime [timestamp].
  - LastLogin [timestamp].

  Comments : This table has no relationships with any other tables as it does not play any part of the actual flow between the rest of the system. It is simply a table to store information about administrators that can log into the administrator page of the website.

- RUNNER :
  - ID [int], PRIMARY KEY
  - FirstName [varchar(50)]
  - LastName [varchar(50)]
  - DateOfBirth [date]

  - Gender [varchar(10)], FOREIGN KEY.
  - Country [varchar(50)].
  - City [varchar(50)].

  Comments : This table holds personal information about all runners ; currently in a race or not. This table has four relationships. Gender is constrained by the Gender entry in the gender table. ID constraints the Runner entry in the race_runner table, the Runner entry in the runner_units table and the Runner entry in the timestamp table.

- gender :
  - Gender [varchar (10)], PRIMARY KEY.

  Comments : This is simply a table to constraint the Gender entries in the runner and class tables.

- race_runner :
  - Race [int], PRIMARY COMPOSITE KEY.
  - Class [int], FOREIGN KEY.
  - Runner [int], PRIMARY COMPOSITE KEY, FOREIGN KEY.
  - Bib [int].
  - Status [varchar(10)], FOREIGN KEY.
  - Club [int], FOREIGN KEY.
  - Place [int].
  - TotalTime [time].

  Comments : This table holds information about active runners. This table has five relationships with other tables in the database. Runner is constrained by the ID entry in the runner table. Race is constrained by the ID entry in the race table. Class is constrained by the ID entry in the class table. Status is constrained by the Status entry in the runner_status table. Club is constrained by the ID entry in the club table.

- club :
  - ID [int], PRIMARY KEY.
  - Name [varchar(100)].

  Comments : This table holds data about all the clubs that has been registered. ID constraints the Club entry in the race_runner table.

- si_unit_status :
  - Status [varchar(20)], PRIMARY KEY.

  Comments : This table simply constraints the Status entry in the si_unit table.

- runner_status :
  - Status [varchar(10)], PRIMARY KEY.

  Comments : This table simply constraints the Status entry in race_runner.

- race_station :
  - Race [int], PRIMARY COMPOSITE KEY, FOREIGN KEY.
  - Station [varchar(12)], PRIMARY COMPOSITE KEY, FOREIGN KEY.

  Comments : This table ties a race to a station. Race is constrained by the ID entry in the race table. Station is constrained by the ID entry in the station table.

- RUNNER_UNITS :
  - Runner [int], PRIMARY COMPOSITE KEY.
  - SI_unit [int], PRIMARY COMPOSITE KEY, FOREIGN KEY.
  - Race [int], PRIMARY COMPOSITE KEY, FOREIGN KEY.

  Comments : This table holds data about SI-Units that a runner is using during a race. Race is constrained by the ID entry in the race table. Runner is constrained by the ID entry in the runner table. SI_unit is constrained by the ID entry in the si_unit table.

- SI_UNIT :
  - ID [int], PRIMARY KEY.
  - Status [varchar(20)], FOREIGN KEY.

  Comments : Holds information about SI-units. ID constraints the SI_unit entry in the runner_units table. Status is constrained by the Status entry in the si_unit_status table.

- RACE :
  - ID [int], PRIMARY KEY.
  - Name [varchar(100)].
  - Date [date].

  Comments : Holds information about a race. ID constraints the Race entries in the race_runner, runner_units, race_station and timestamp tables.

- CLASS :
  - ID [int], PRIMARY KEY.
  - Gender [varchar(20)], FOREIGN KEY.
  - Distance [int].

  Comments : Holds information about how long a race is and which gender runs it. ID constraints the Class entry in the race_runner table. Gender is constrained by the Gender entry in the gender table.

- STATION :
  - ID [varchar(12)], PRIMARY KEY.
  - Name [varchar(20)].
  - Code [int].
  - LengthFromStart [int].
  - LastID [int].

  Comments : Holds information about a station (not tied to a specific race). ID constraints the Station entries in the race_station and the timestamp tables.

- TIMESTAMP :
  - Timestamp [datetime], PRIMARY COMPOSITE KEY.
  - SI_unit [int], FOREIGN KEY.
  - Runner [int], PRIMARY COMPOSITE KEY, FOREIGN KEY.
  - Station [varchar(12)], FOREIGN KEY.
  - Race [int], FOREIGN KEY.
  - Place [int].

  Comments : Holds information about a timestamp. SI_unit is constrained by the ID entry in the si_unit table. Runner is constrained by the ID entry in the runner table. Station is constrained by the ID entry in the station table. Race is constrained by the ID entry in the race table.

4.3.2 Front-End

The structure of the website consists of a main page INDEX.PHP that holds the main structure of the site. This page displays the correlated content depending on the value of the parameter PAGE in the URL.

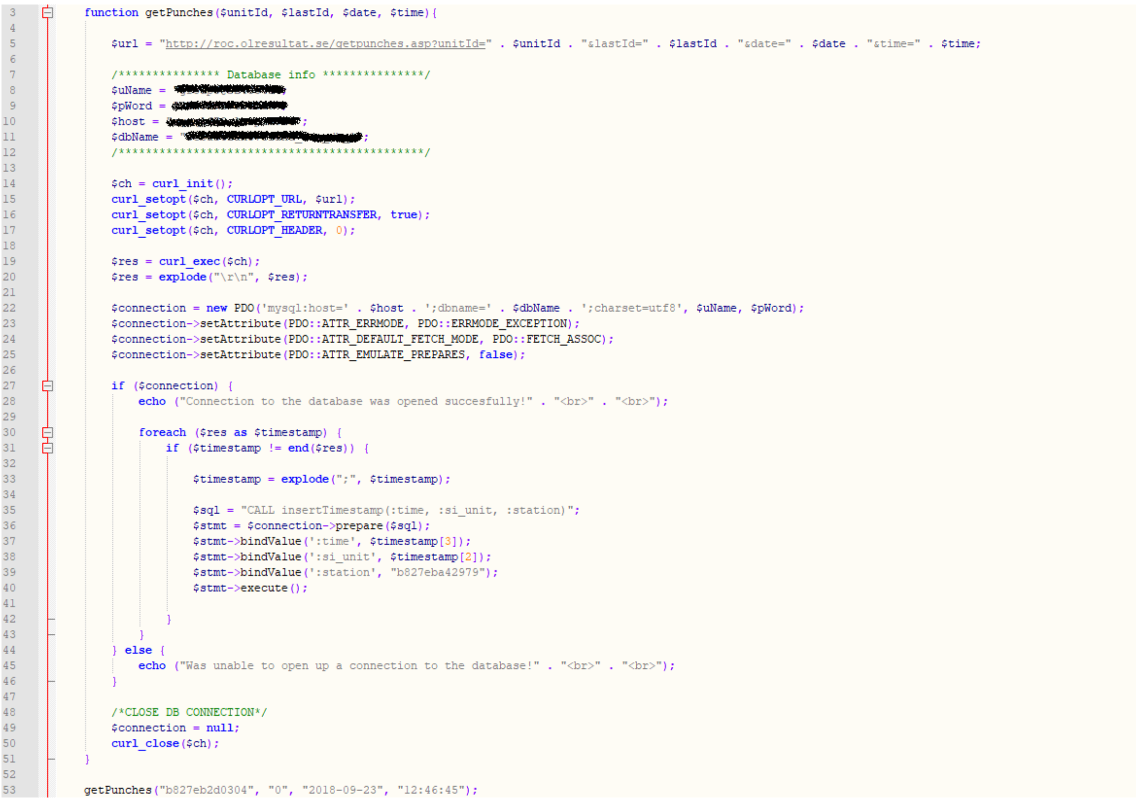
The file tree of the website is currently as following :

- BOOTSTRAP : Contains all Bootstrap files needed in the project.
  - *Bootstrap files.*
- CSS : Contains CSS files in order to change visual properties of the website.
  - STYLES.CSS : Changes and adds properties to elements in the pages.
- FUNCTIONS : Contains the PHP files.
  - MAIN.PHP : Contains all the functions needed to manage the database from the site and fetch data from the database to the site.
- IMAGES : Contains the images included in the website.
  - *Some images.*
- JS : Contains all the Javascript files needed in the project.
  - *Needed javascript files.*
  - MAIN.JS : Contains Javascript functions developed in order to trigger events like pop-up...
- PAGES : Contains all the website pages content that can be required by the INDEX.PHP page.
  - ADMIN.PHP
  - ERROR.PHP
  - HOME.PHP
  - LOGIN.PHP
  - LOGOUT.PHP
  - MANAGE-RACE.PHP
  - MANAGE-RUNNER.PHP
  - MANAGE-TEAM.PHP
  - MANAGE-TIMESTAMP.PHP
  - RACE.PHP
  - RACE-LIST.PHP
  - RUNNER.PHP
  - RUNNER-LIST.PHP
  - TEAM.PHP
  - TEAM-LIST.PHP
- INCLUDE.PHP : Called the MAIN.PHP
- INDEX.PHP : Contains the main structure of the website and requires INCLUDE.PHP and content in the PAGES folder.

In future development, the group wants to divide the MAIN.PHP file into various files in order to make it lighter. The main page will call only PHP files that are required according to the page displayed.

4.3.3 Back-End

- **GetPunches script :**



```php
function getPunches($unitId, $lastId, $date, $time){

    $url = "http://roc.olresultat.se/getpunches.asp?unitId=" . $unitId . "&lastId=" . $lastId . "&date=" . $date . "&time=" . $time;

    /*************** Database info ***************/
    $uName = [redacted]
    $pWord = [redacted]
    $host = [redacted];
    $dbName = [redacted];
    /*********************************************/

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_HEADER, 0);

    $res = curl_exec($ch);
    $res = explode("\r\n", $res);

    $connection = new PDO('mysql:host=' . $host . ';dbname=' . $dbName . ';charset=utf8', $uName, $pWord);
    $connection->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $connection->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
    $connection->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);

    if ($connection) {
        echo ("Connection to the database was opened succesfully!" . "<br>" . "<br>");

        foreach ($res as $timestamp) {
            if ($timestamp != end($res)) {

                $timestamp = explode(";", $timestamp);

                $sql = "CALL insertTimestamp(:time, :si_unit, :station)";
                $stmt = $connection->prepare($sql);
                $stmt->bindValue(':time', $timestamp[3]);
                $stmt->bindValue(':si_unit', $timestamp[2]);
                $stmt->bindValue(':station', "b827eba42979");
                $stmt->execute();

            }
        }
    } else {
        echo ("Was unable to open up a connection to the database!" . "<br>" . "<br>");
    }

    /*CLOSE DB CONNECTION*/
    $connection = null;
    curl_close($ch);
}

getPunches("b827eb2d0304", "0", "2018-09-23", "12:46:45");
```

Figure 4: GETPUNCHES Script.

This script shall be run whenever the website has to update its data. The function getPunches fetches the punches from the third-party webserver as a string and then parses them into an array. After that, a connection to the database is opened and the punches are added to it through a stored procedure.
It is currently (2018-12-01) just in a "prototype" stage. The function getPunches is called with static values, which will not be the case in the live version. The function also has the database credentials hardcoded, which is not a good practice and this will be dealt with before the actual website goes live.

Some other back-end scripts will be described here in the future version of this document.

4.4 Frameworks

A framework is a great tool to use to get a readable overview structure even if the user is inexperienced [1]. This is because the tool only allows the user to use it in some ways, and the structure is almost always similar when built. With this in mind, framework is a good tool if the work is going to be handed over because of the readable structure. Another thing is if the implementer is inexperienced with what is being built, the framework can help the implementer stay within the frames of the programming norms of that language.

The group have had some experience with all the different areas of the project, mostly the back-end and database related. This made the choice to not use a framework to help the group easier to take as the group is experienced enough to make good implemental decisions and structure these parts in a good way.
In the front-end all the group members have done some work before with both frameworks and

without, but not to the degree of the other parts of the program. Here the group decided that in the testing face of the navigation and colors, no framework is going to be used.
Later, when the final prototype is being implemented, the group have decided to use what the group have worked with before, which is Bootstrap. This framework can help the group in different ways, but the most important to the group is the responsive aspects of the front-end.

5 System Usage

5.1 Use Case

As described in the previous document (Project Plan) the diagram below presents the different actors of the system as well as their possible actions.

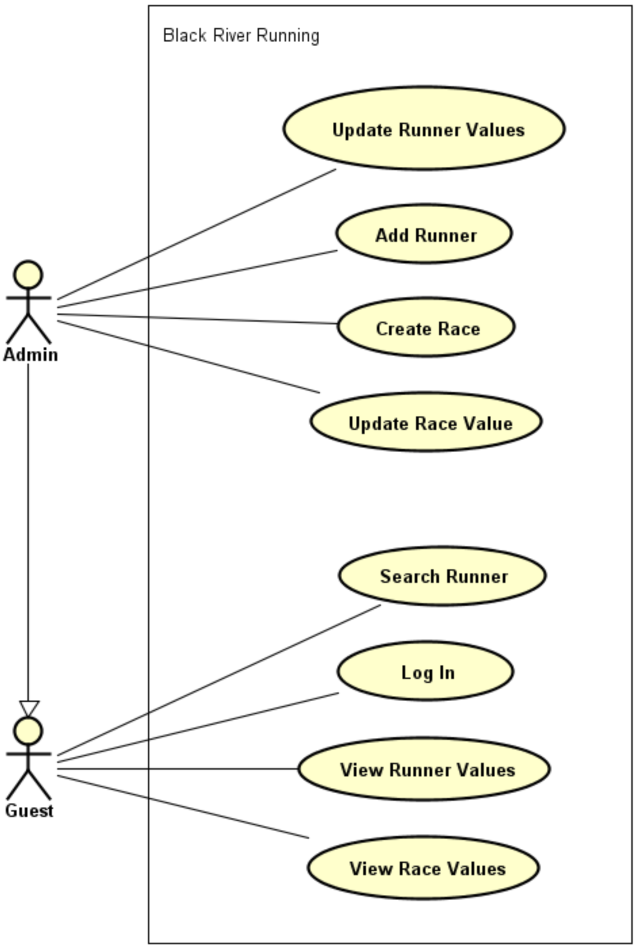


Figure 5: Use case Diagram.

In the rest of this section, each of the actors and the functionalities of the system will be described in more detail.

5.2 Users Functions And Interactions

5.2.1 Guest

The interactions between the guest user and the system are limited by clicking on buttons to navigate around the website. The functionalities provided are the possibility to display lists and search for information (runners, races and so on) by entering keywords.

The guest user is not able to change the data stored in the database. He/she can only view certain data in the database that are displayed on the website.

5.2.2 Administrator

The administrator status is given to a user by accessing to the Login page of the website and successfully logging in to the page.

The administrator inherits from the guest user, which means that all features and functionalities of the guest user can be performed by the administrator.

The new interaction possible for the administrator is to click +, . . . and x buttons.

5.3 Graphical User Interface Wireframe

In this part, the wireframes of the final system is defined through several images. These wireframes were used by the group in order to get an overview and a similar idea of the final design of the website.
The wireframes are models for developping the UI and are still under construction.

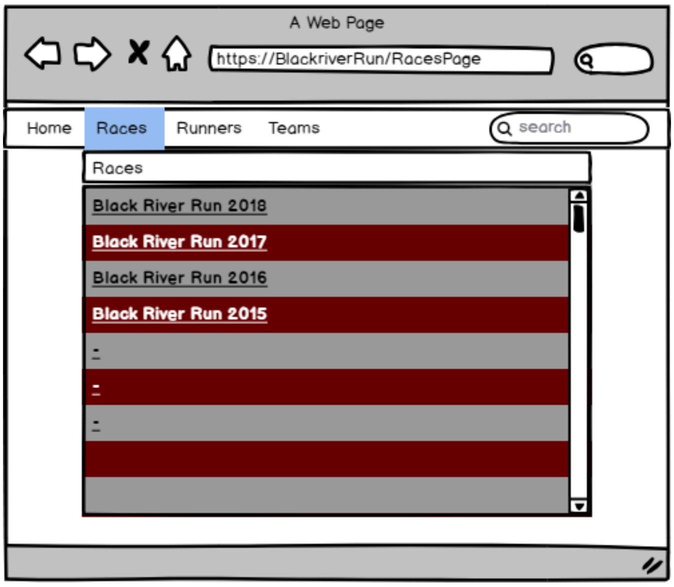
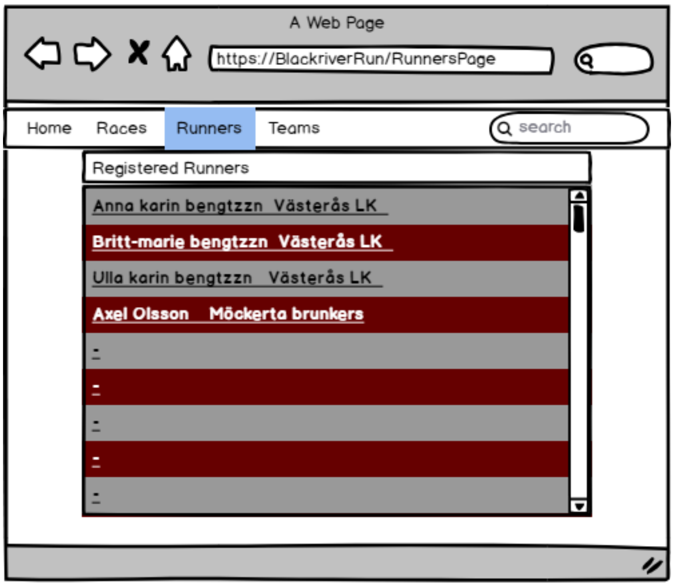


Figure 6: Race List page wireframe.



Registered Runners

Anna karin bengtzzn Västerås LK

Britt-marie bengtzzn Västerås LK

Ulla karin bengtzzn Västerås LK

Axel Olsson Möckerta brunkers

-

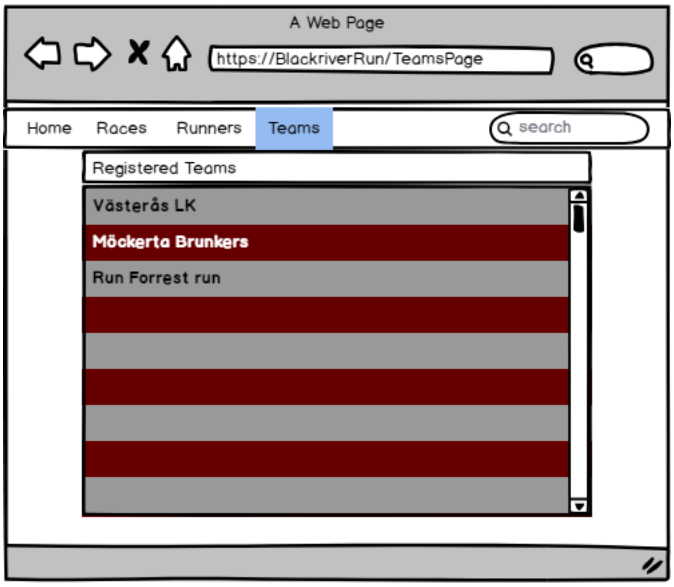
-

-

-

-

Figure 7: Runner List page wireframe.



Registered Teams

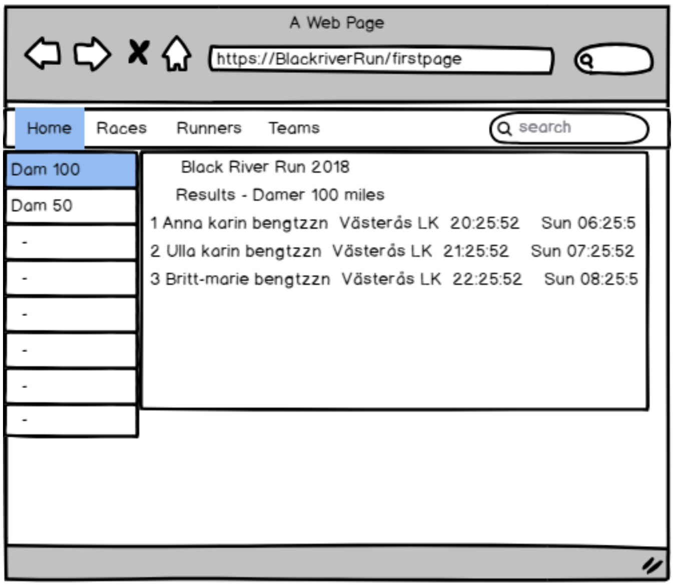
Västerås LK

Möckerta Brunkers

Run Forrest run

Figure 8: Team List page wireframe.

The three figures above shows three lists that guest users are able to see in the website by clicking on the three buttons in the main menu.

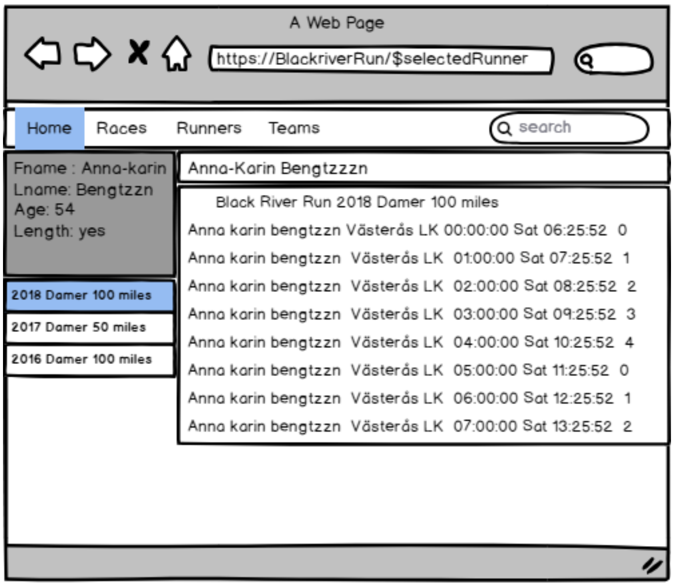


Black River Run 2018

Results - Damer 100 miles

1 Anna karin bengtzzn Västerås LK 20:25:52 Sun 06:25:5
2 Ulla karin bengtzzn Västerås LK 21:25:52 Sun 07:25:52
3 Britt-marie bengtzzn Västerås LK 22:25:52 Sun 08:25:5

Figure 9: RACE page wireframe.



Fname : Anna-karin
Lname: Bengtzzn
Age: 54
Length: yes

Anna-Karin Bengtzzzn

Black River Run 2018 Damer 100 miles

Anna karin bengtzzn Västerås LK 00:00:00 Sat 06:25:52 0
Anna karin bengtzzn Västerås LK 01:00:00 Sat 07:25:52 1
Anna karin bengtzzn Västerås LK 02:00:00 Sat 08:25:52 2
Anna karin bengtzzn Västerås LK 03:00:00 Sat 09:25:52 3
Anna karin bengtzzn Västerås LK 04:00:00 Sat 10:25:52 4
Anna karin bengtzzn Västerås LK 05:00:00 Sat 11:25:52 0
Anna karin bengtzzn Västerås LK 06:00:00 Sat 12:25:52 1
Anna karin bengtzzn Västerås LK 07:00:00 Sat 13:25:52 2

Figure 10: RUNNER page wireframe.

The two figures above show respectively a page that displays specific race information and a page that displays specific runner information.

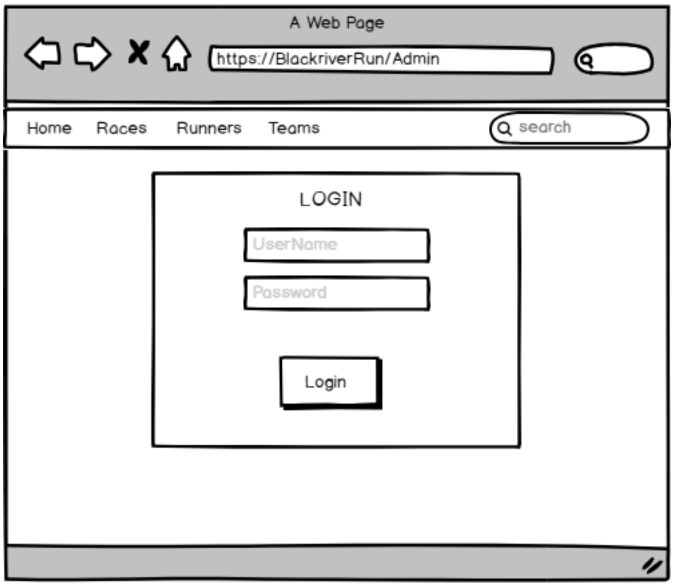


Figure 11: Login page wireframe.

The figure above shows the login page. By entering good combination of username and password, a user can become a administrator.

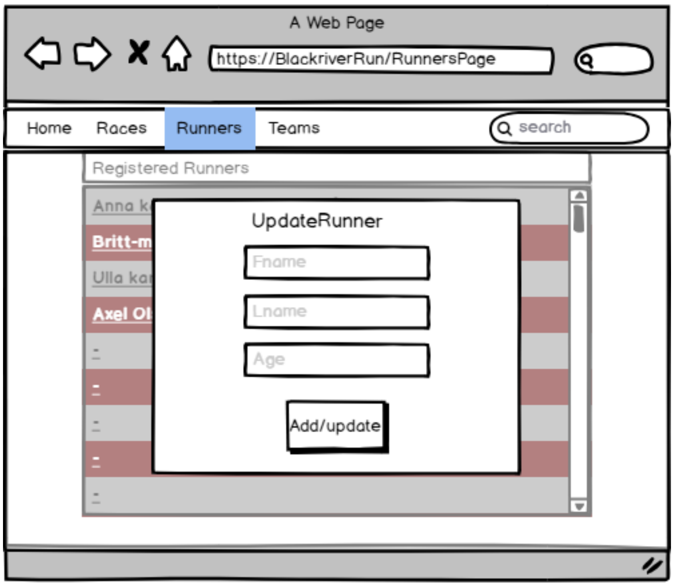
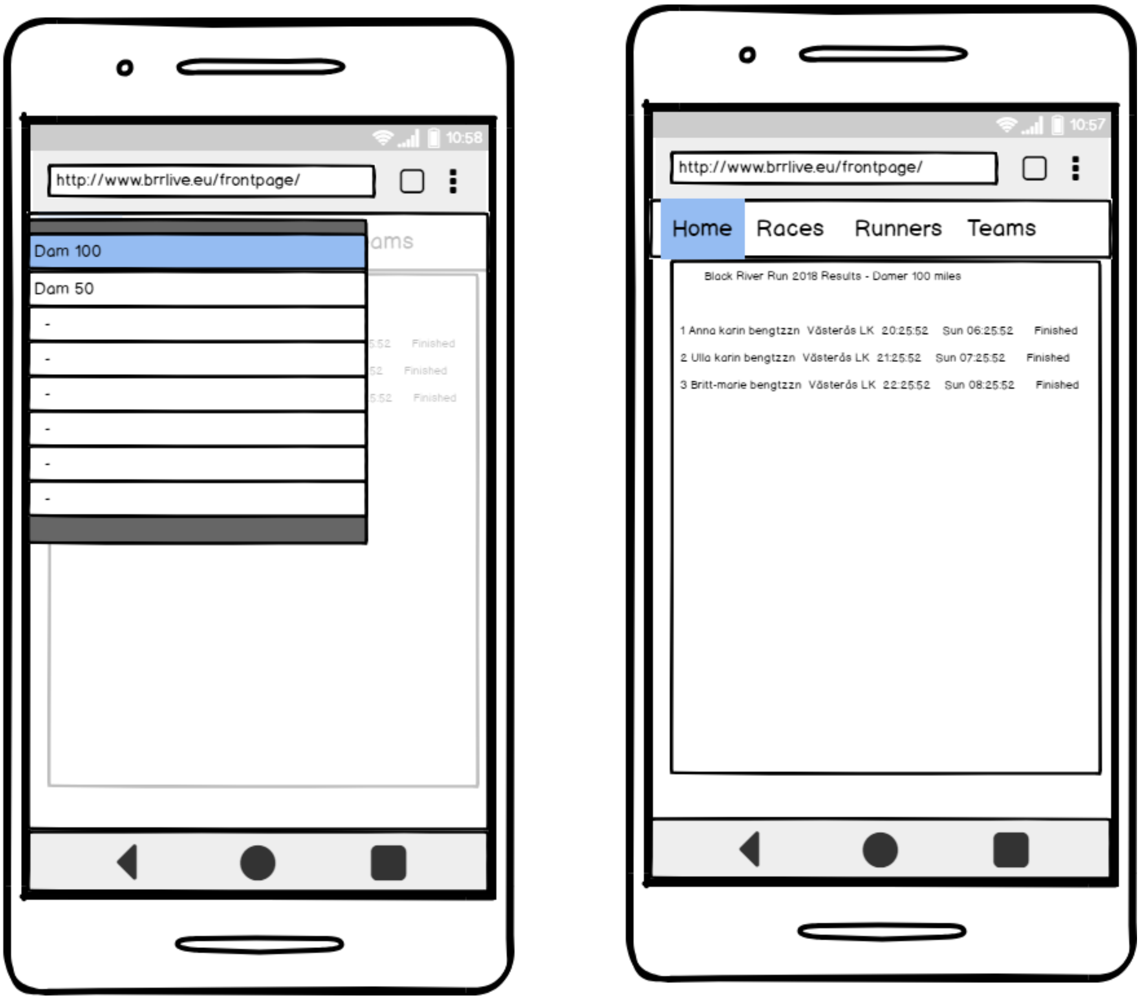


Figure 12: Manage Runner page wireframe.

The figure above shows the page that allows to modify runner's data.


10:58
http://www.brrlive.eu/frontpage/
ams
Dam 100
Dam 50
-
-
-
-
-
-
10:57
http://www.brrlive.eu/frontpage/
Home
Races
Runners
Teams

Black River Run 2018 Results - Damer 100 miles

1 Anna karin bengtzzn	Västerås LK	20:25:52	Sun 06:25:52	Finished
2 Ulla karin bengtzzn	Västerås LK	21:25:52	Sun 07:25:52	Finished
3 Britt-marie bengtzzn	Västerås LK	22:25:52	Sun 08:25:52	Finished

Figure 13: Mobile version wireframe.

The group has also think about the mobile version of the website. Indeed, a user who wants to see results of a race in progress will probably be outside and may not have access to a computer.

5.4 Graphical User Interface Overview

The core functionality and design implementation of the GUI website is done. It was developed in PHP, HTML/CSS, with the help of the Bootstrap framework. Bootstrap helped with keeping the development neat and clear.
In this part of the report, the different pages are shown in categories :

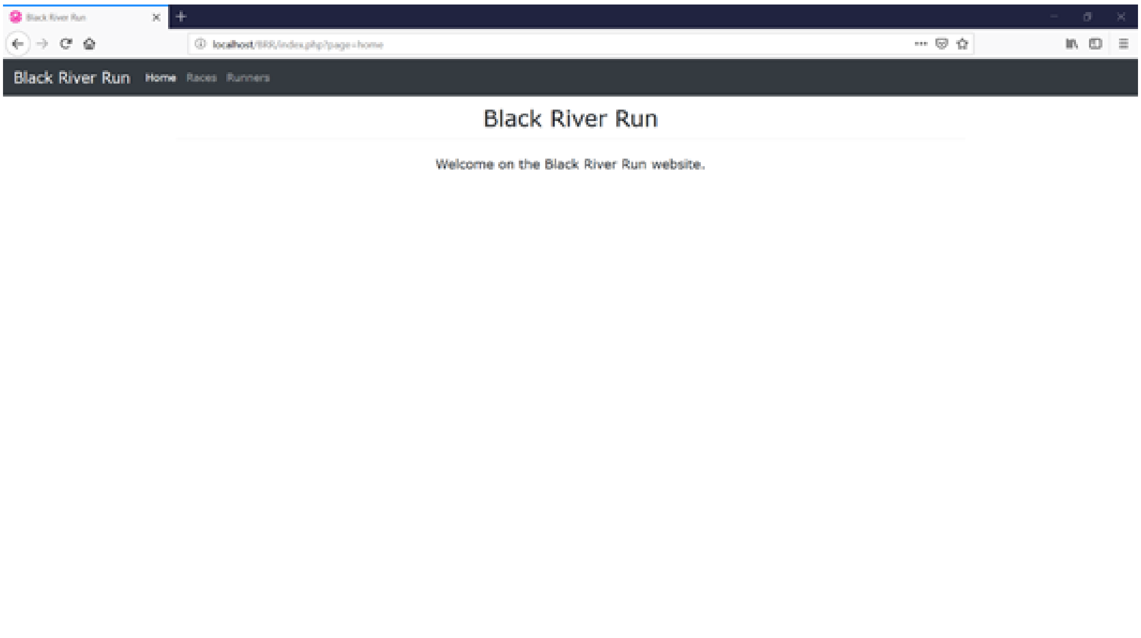
- Home.
- Login.
- Race.
- Runner.
- Team.
- Control Panel.

Each one of these categories contains screen captured images and information about the GUI.

The already existing and implemented behavior of the pages will be explained, and if it is not, the reader will be informed of it.

Furthermore, the data displayed on the pages are test data read from the database. Some of this data may be inconsistent at the moment, but this will be corrected when the GUI will be used to test the back-end implementation.

5.4.1 Home Page



Black River Run

Welcome on the Black River Run website.

Figure 14: Home page.

The screenshot above shows the Home page of the new UI for viewing the Black River Run results. On this page guest user can access the races list or the runners list by clicking on button in the top menu.

On this page the guest user can access the list of races, the list of teams and the list of runners by clicking the buttons on the top.

This page is not finished yet and will contain in the future the current race results displayed in live, information about the event, and links to the race website.

5.4.2 Login Page

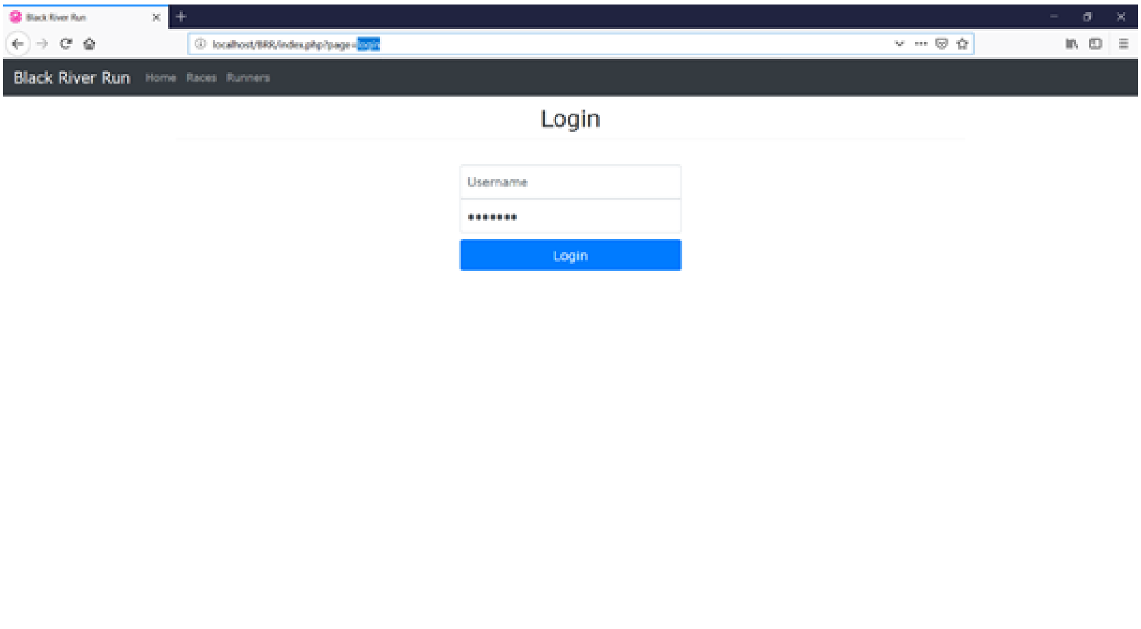
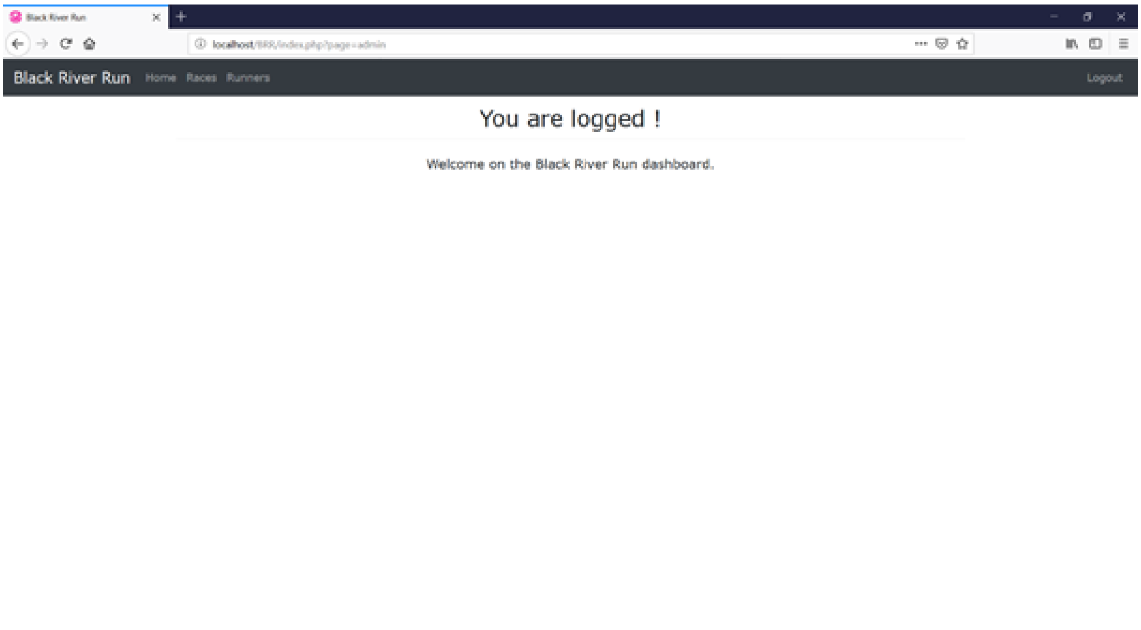


Figure 15: LOGIN page.



You are logged !

Welcome on the Black River Run dashboard.

Figure 16: LOGGED page.

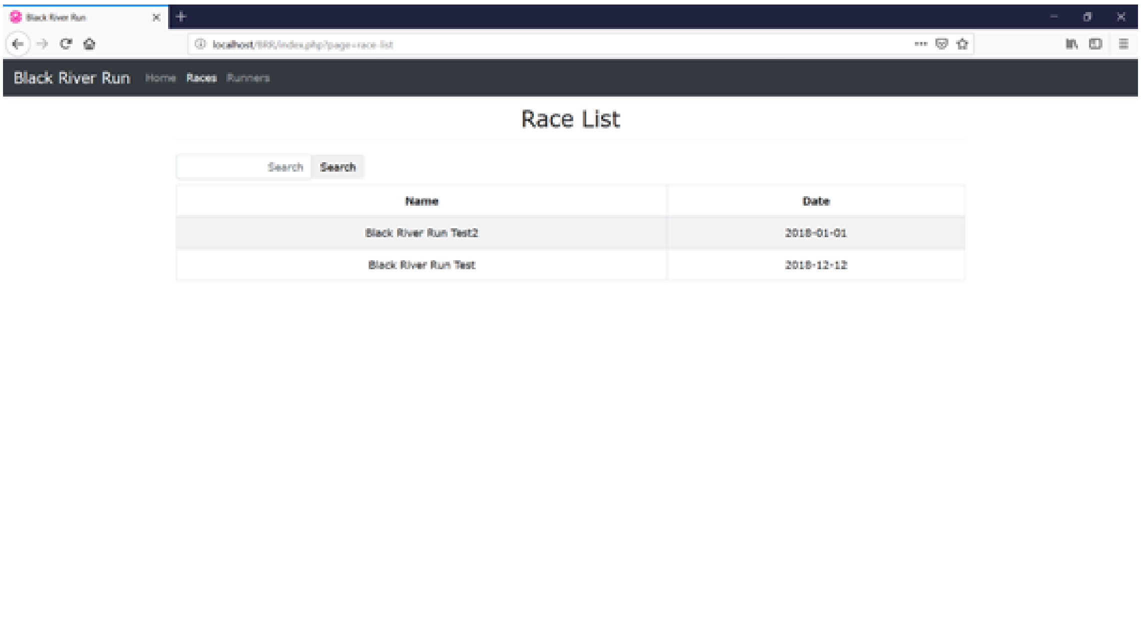
The screenshots above show the LOGIN page of the website. The client wanted it hidden and the only way of getting to that page is by typing the correct URL. Then the user must enter a correct combination of username and password to be identified by the system as an administrator.
When the system has acknowledged the user as an administrator, a success message is shown along with a LOGOUT button in the top right corner to leave the administrator status.

Once logged in, the website will contain more information and the administrator will be allowed to perform more actions. In the following parts, guest view and administrator view will be shown one after the other in order to see changes between both.

The main change when a user is logged is the display of three new buttons +, . . . , X. These buttons allow respectively to add, edit and delete information related to the element displayed

5.4.3 Race Pages

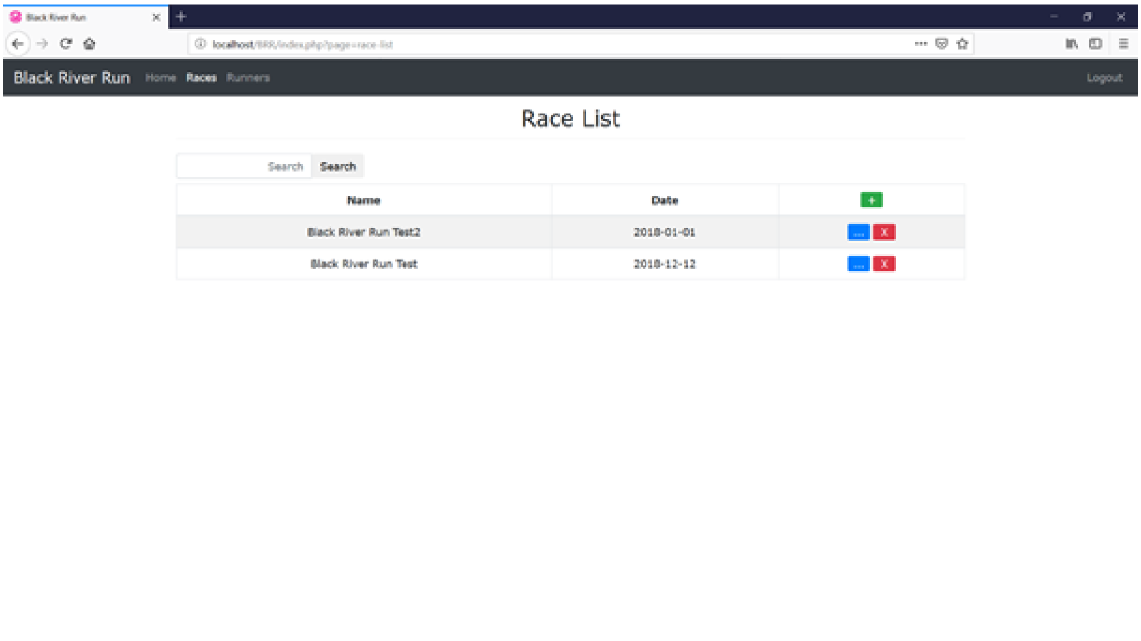
In this part, the pages related to the races will be described.



Race List

Name	Date
Black River Run Test2	2018-01-01
Black River Run Test	2018-12-12

Figure 17: Race List page from guest view.



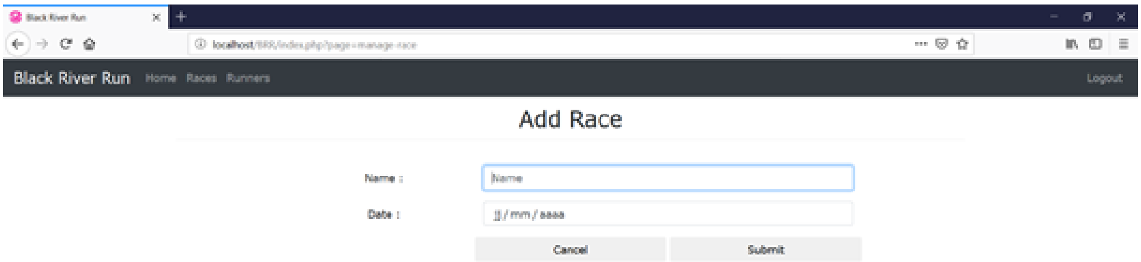
Race List

Name	Date	+
Black River Run Test2	2018-01-01	... X
Black River Run Test	2018-12-12	... X

Figure 18: Race List page from administrator view.

The Race List page is accessible by any other pages of the website, just by a simple click on the Races button.

On this page every races are displayed with their Name and Date. It is possible to search for a specific race by name to filter this list if it is too long.
A click on a race line will lead the user to the chosen race.
As an administrator, a click on + and . . . buttons will lead to another page that will be detailed below.



localhost/BRR/index.php?page=manage-race

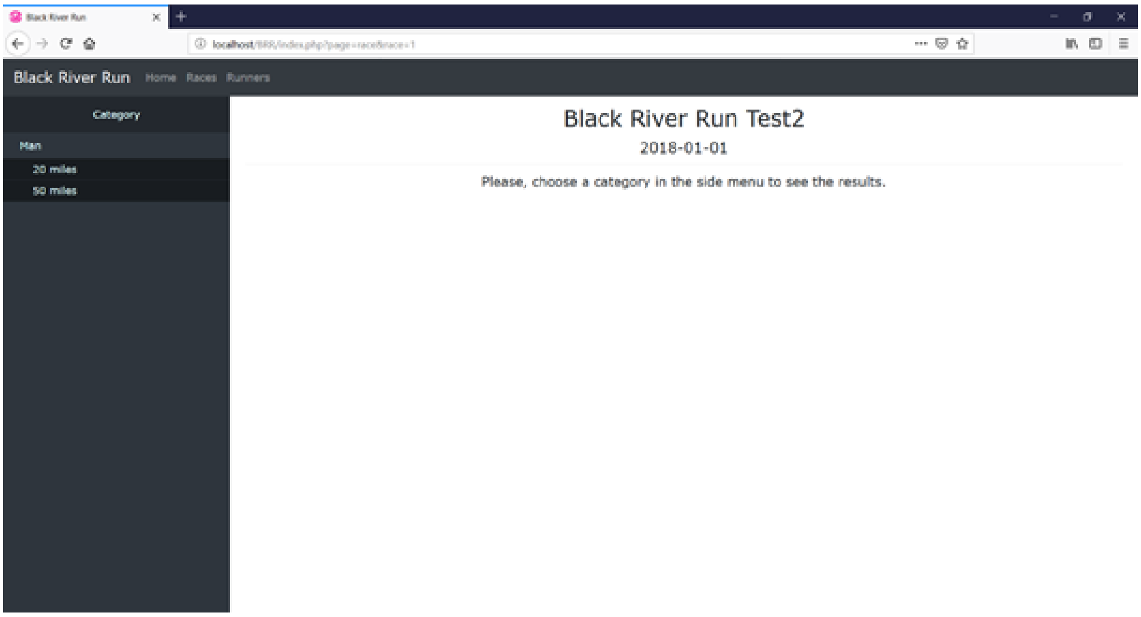
Add Race

Name : Name

Date : jj / mm / aaaa

Figure 19: Manage Race page.

The Manage Race page consists in a simple form that allows to add a race to the system. If the administrator choose to edit the form is prefilled with the value of the selected race.



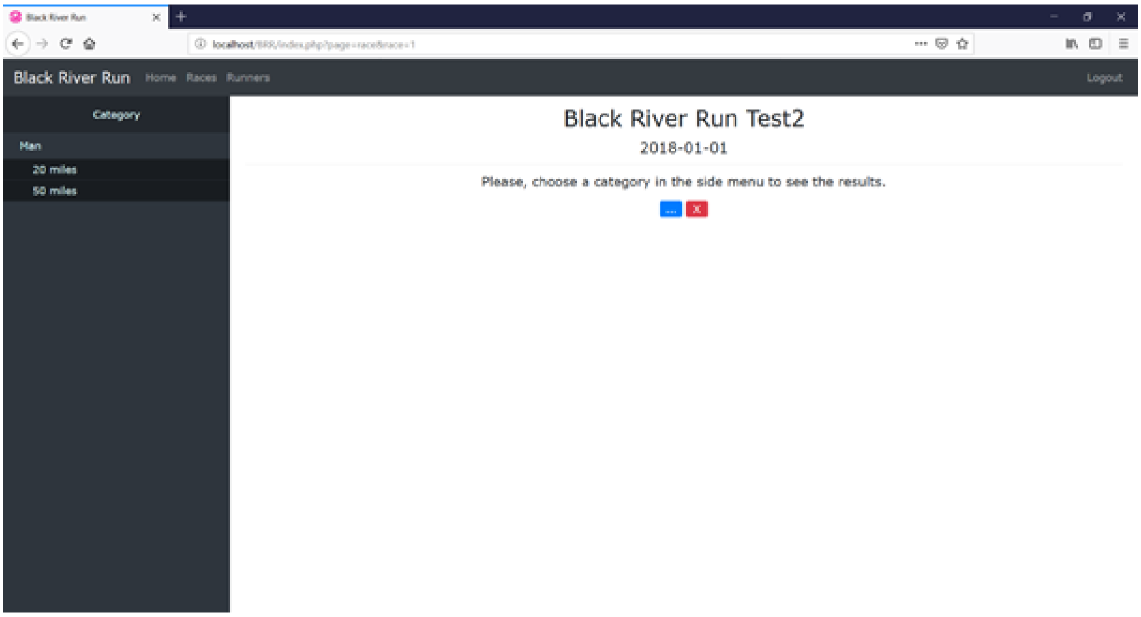
localhost/BRR/index.php?page=race&race=1

Black River Run Test2

2018-01-01

Please, choose a category in the side menu to see the results.

Figure 20: Race page from guest view.



localhost/BRR/index.php?page=race&race=1

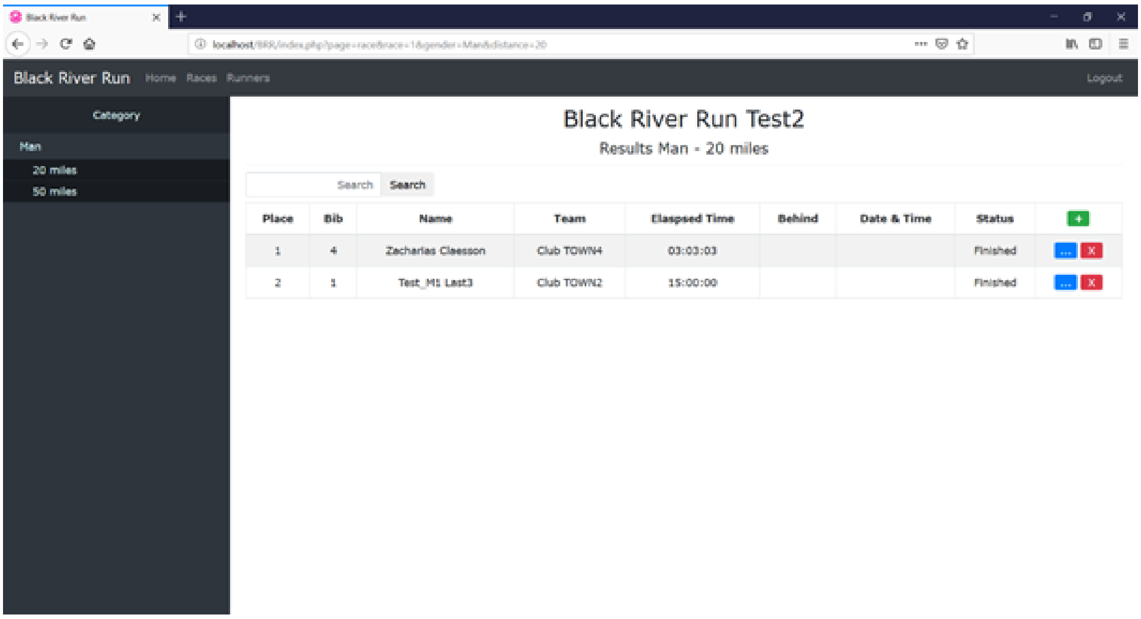
Black River Run Test2

2018-01-01

Please, choose a category in the side menu to see the results.

Figure 21: Race page from administrator view.

When a user navigates to a Race page, the navigation to different categories of runners will be displayed to the left of the page.
A click on one of the categories will display the results of this specific race category.



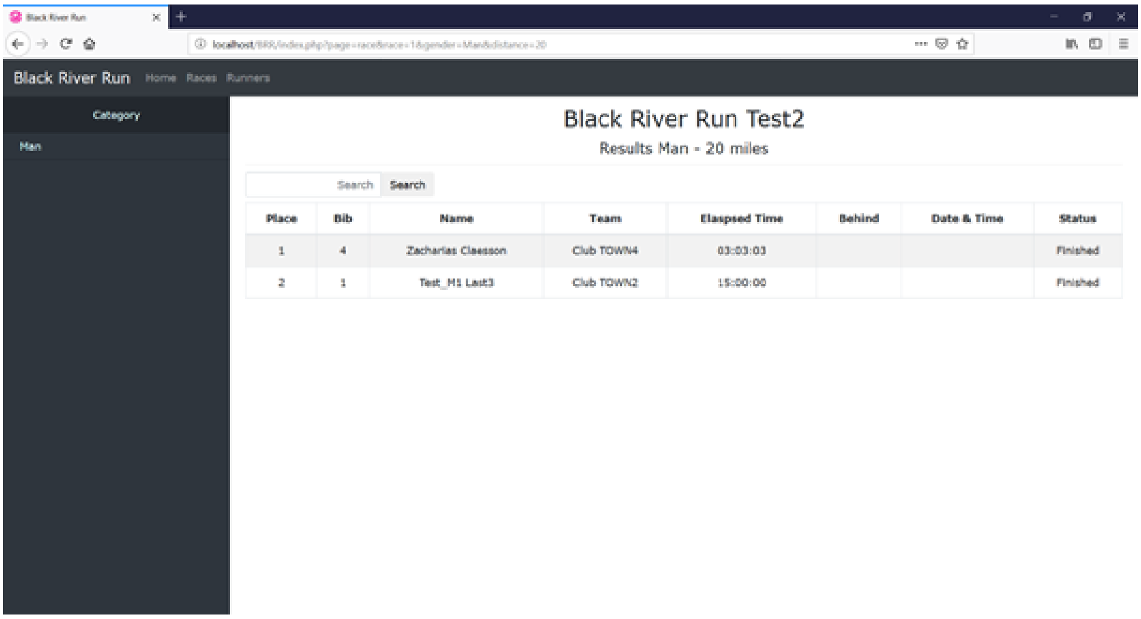
localhost/BRR/index.php?page=race&race=1&gender=Man&distance=20

Black River Run Test2

Results Man - 20 miles

Place	Bib	Name	Team	Elaspsed Time	Behind	Date & Time	Status	+
1	4	Zacharias Claesson	Club TOWN4	03:03:03			Finished	... X
2	1	Test_M1 Last3	Club TOWN2	15:00:00			Finished	... X

Figure 22: Race page with a category selected from administrator view.



Black River Run Test2

Results Man - 20 miles

Place	Bib	Name	Team	Elapsed Time	Behind	Date & Time	Status
1	4	Zacharias Claesson	Club TOWN4	03:03:03			Finished
2	1	Test_M1 Last3	Club TOWN2	15:00:00			Finished

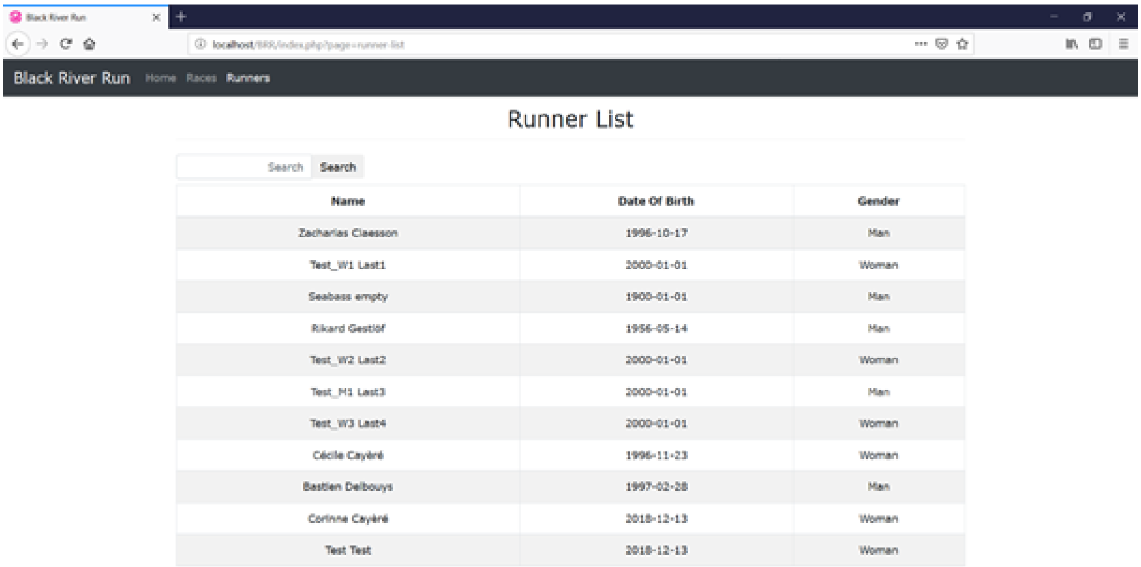
Figure 23: Race page with a category selected from guest view.

The display of the results for a chosen category in a chosen race are shown on the screenshots above. Every runners are listed with their Place, Bib, Name, Team, Elapsed Time, time Behind the first runner, Date & Time of finishing, Status (Finished, Do Not Finish, Do Not Start, Running).

The search field can be used to find a runner by Name, Bib or Team.
A click on a runner line will redirect the user on the specific Runner page. When logged as an administrator, the ... button allows to edit information about the runner in the race. For example, The administrator has the choice to take into account or not the timestamps. This functionnalities allows to avoid taking into account a series of false timestamps and permits to give the true final results of a runner.
A click on the + button will lead the administrator to the Add Runner-Race page (*Not Done Yet*).

5.4.4 Runner Pages

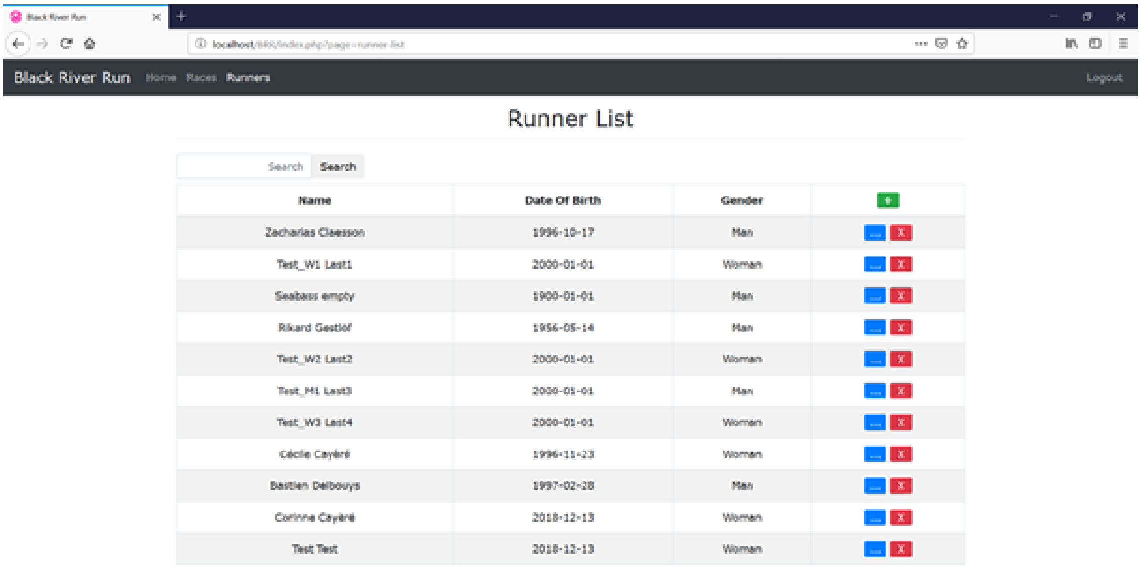
In this part, the pages related to the runners will be explained.



Runner List

Name	Date Of Birth	Gender
Zacharias Claesson	1996-10-17	Man
Test_W1 Last1	2000-01-01	Woman
Seabass empty	1900-01-01	Man
Rikard Gestlöf	1956-05-14	Man
Test_W2 Last2	2000-01-01	Woman
Test_M1 Last3	2000-01-01	Man
Test_W3 Last4	2000-01-01	Woman
Cécile Cayéré	1996-11-23	Woman
Bastien Delbouys	1997-02-28	Man
Corinne Cayéré	2018-12-13	Woman
Test Test	2018-12-13	Woman

Figure 24: Race List page from guest view.



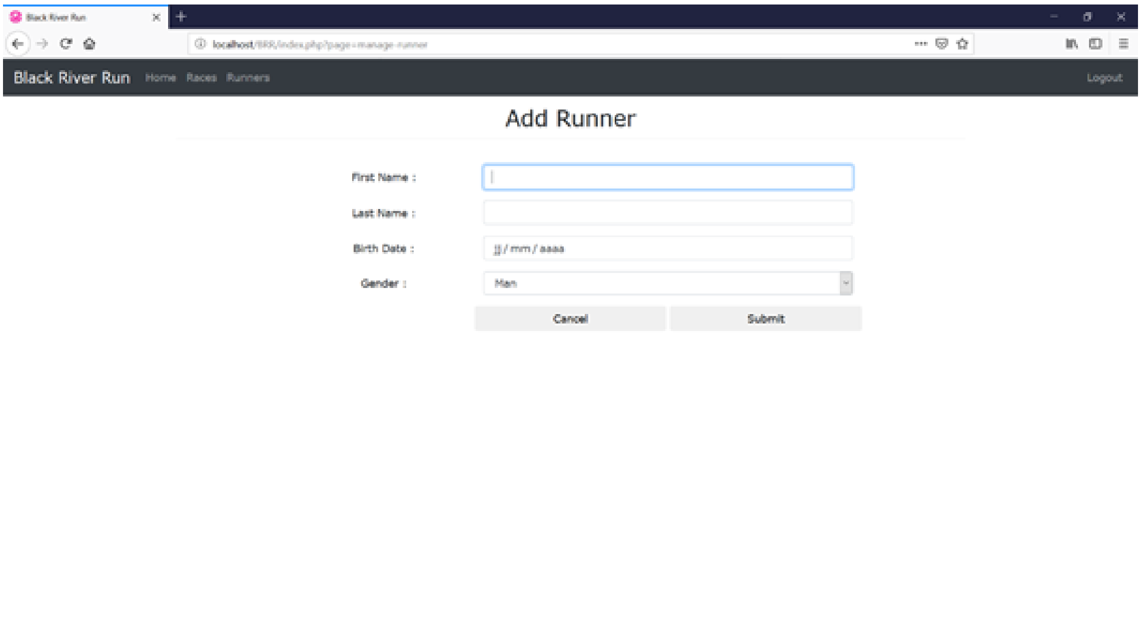
Runner List

Name	Date Of Birth	Gender	+
Zacharias Claesson	1996-10-17	Man	... X
Test_W1 Last1	2000-01-01	Woman	... X
Seabass empty	1900-01-01	Man	... X
Rikard Gestlöf	1956-05-14	Man	... X
Test_W2 Last2	2000-01-01	Woman	... X
Test_M1 Last3	2000-01-01	Man	... X
Test_W3 Last4	2000-01-01	Woman	... X
Cécile Cayéré	1996-11-23	Woman	... X
Bastien Delbouys	1997-02-28	Man	... X
Corinne Cayéré	2018-12-13	Woman	... X
Test Test	2018-12-13	Woman	... X

Figure 25: Race List page from administrator view.

The Runner List page displays every runners registered in the system with their Name, Date of Birth and Gender.
The search field allows a user to search for a runner by its First Name and/or Last Name.

A click on a runner line will lead to the Runner page associated to it.
A click on the + or ... buttons will redirect the administrator to the Manage Runner page described below.



Add Runner

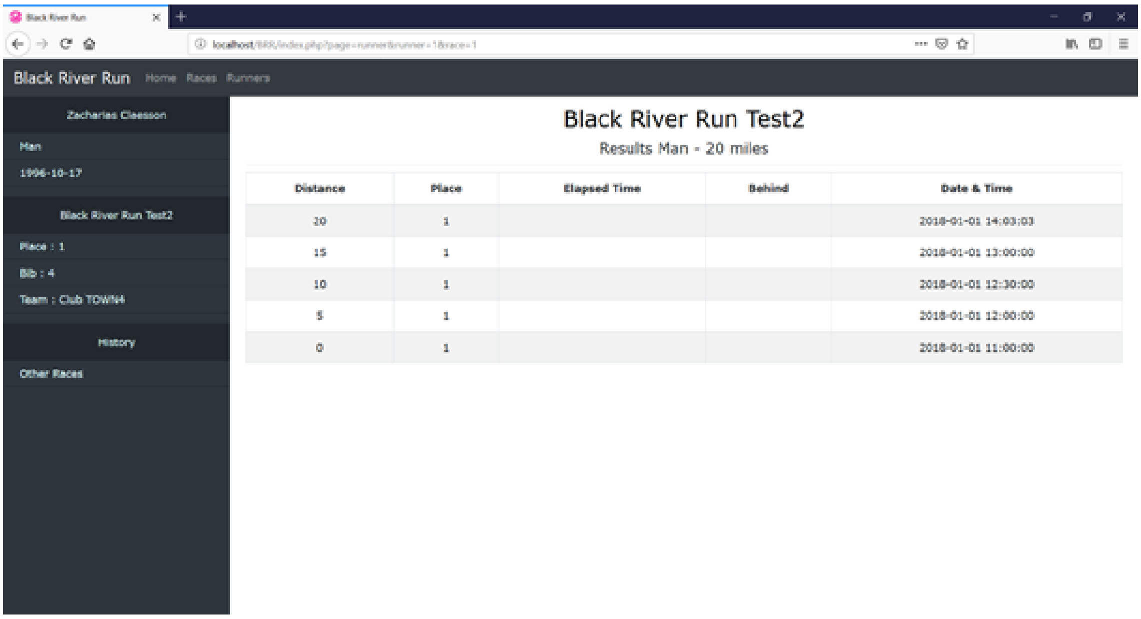
First Name :

Last Name :

Birth Date : jj / mm / aaaa

Gender : Man

Figure 26: Add Runner page.



Zacharias Claesson

Man

1996-10-17

Black River Run Test2

Place : 1

Bib : 4

Team : Club TOWN4

History

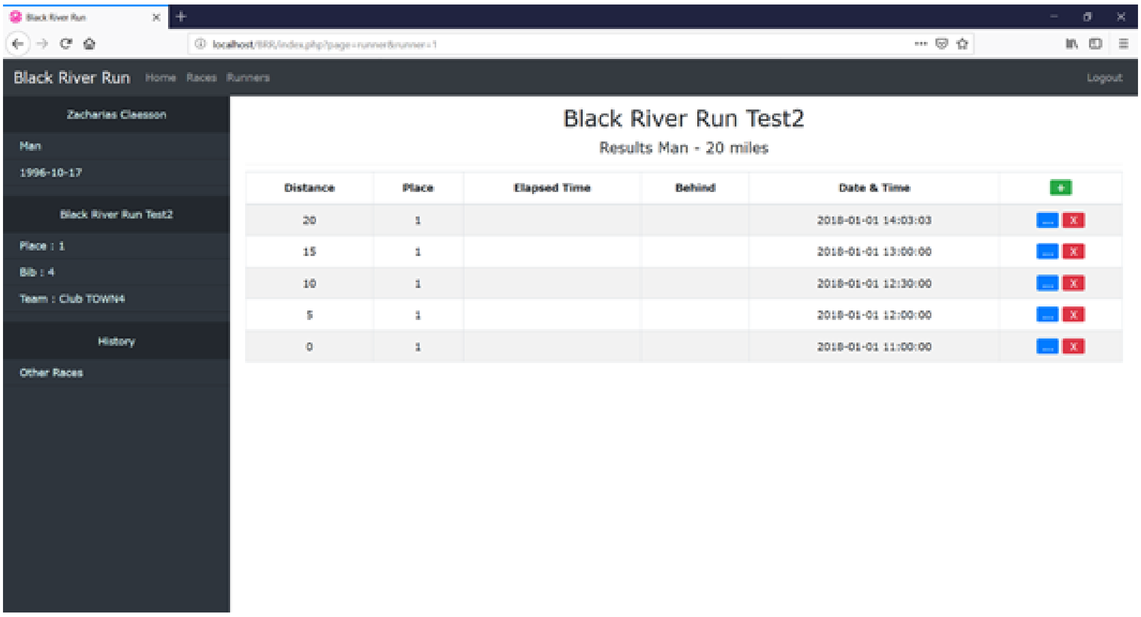
Black River Run Test2

Results Man - 20 miles

Distance	Place	Elapsed Time	Behind	Date & Time
20	1			2018-01-01 14:03:03
15	1			2018-01-01 13:00:00
10	1			2018-01-01 12:30:00
5	1			2018-01-01 12:00:00
0	1			2018-01-01 11:00:00

Figure 27: Runner page from guest view.

The Manage Runner page consists in a simple form that allows to add a runner to the system. If the administrator choose to edit the form is prefilled with the value of the selected runner.



Zacharias Claesson

Man

1996-10-17

Black River Run Test2

Place : 1

Bib : 4

Team : Club TOWN4

History

Other Races

Black River Run Test2

Results Man - 20 miles

Distance	Place	Elapsed Time	Behind	Date & Time	+
20	1			2018-01-01 14:03:03	... X
15	1			2018-01-01 13:00:00	... X
10	1			2018-01-01 12:30:00	... X
5	1			2018-01-01 12:00:00	... X
0	1			2018-01-01 11:00:00	... X

Figure 28: Runner page from administrator view.

The screenshots above show a Runner page. This page contains more detailed information :

- On the main part of the page :
  - The race Name and Category are displayed. The Name is clickable to go on the Race page.
  - All timestamps for this runner in this race, with the Distance, Place, Elapsed Time, time Behind the first runner, Date & Time.
- On the left side of the page :
  - The runner Name, Date of Birth and Gender are displayed.
  - The runner Place, Bib, Team for the selected race are displayed.
  - The history of all the races in which the runner participated. By clicking on Other Races, this list of races appears.

A click on + or . . . redirects the administrator to the Manage Timestamp page (*Not Done Yet*).

5.4.5 Team Pages

The Team List page (*Not Done Yet*) is accessible by clicking on the Teams button on the top. It allows every users to see the different teams registered in the system. Selecting one of them shows runners of this team.

5.4.6 Control Panel

The Control Panel (*Not Done Yet*) behavior and structure are sketched and defined.
It is accessible by clicking the Control Panel button. This button is only shown to administrator, on the right next to the Logout button.
On this page, the administrator can choose between a list of elements to see and manage :

- Race List : Redirects to the already existing page.
- Runner List : Redirects to the already existing page.
- Team List : Redirects to a page visible by all users where teams are listed with their members.

- SI-Unit List : Allows to add and remove them, give to a runner and take from a runner.
- Station List : Allows to manage station around the race track, Milestone, Id.
- Category List: Allows to add, remove and edit categories by choosing for a gender and distance.

In summary, this page will allow the administrator to manage every tables from the database directly from the website.

References

[1] Dirk Riehle. Framework Design: A Role Modeling Approach, Swiss Federal Institute of Technology. 2000.