

Rock-Paper-Scissors-Lizard-Spock Web-Application

CS-M68 Coursework

Francesca Madeddu
803623@swansea.ac.uk

November 18, 2014

1 Intro

The web application implements the Rock-Paper-Scissors-Lizard-Spock game and offers the user a chance to play multiple games, to look at the statistics about the previous played games and eventually to reset them. The user needs to be logged-in in order to play and he can delete his account (and all his data) whenever he wants through the user admin panel. To perform the registration the user needs to give a username (alphanumeric) and his date of birth. Finally when the user is on his birthday a happy birthday message is shown to him.

2 Navigation workflow

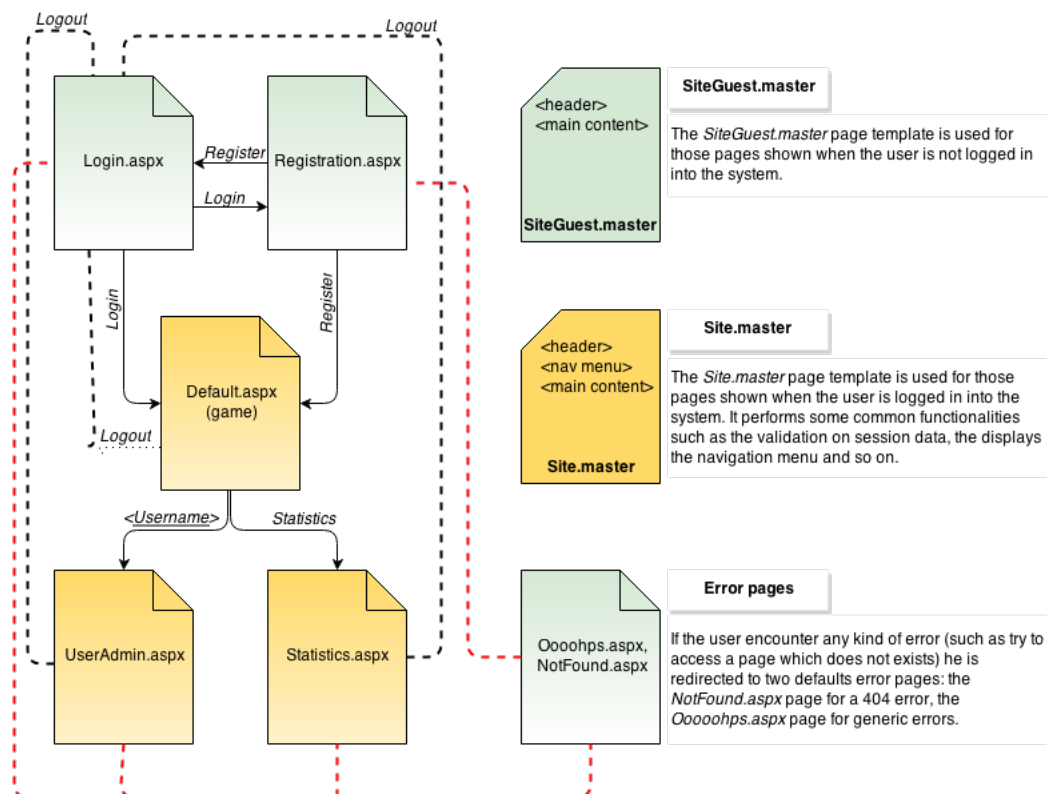


Figure 1: Pages of the web-application

2.1 Login and Registration

The navigation through the web application can start from the *Login.aspx* page or from the *Registration.aspx* page: the user in fact needs to be registered in order to play.

Figure 2 consists of two side-by-side screenshots of the web application interface. Both screenshots have the title 'ROCK-PAPER-SCISSORS-LIZARD-SPOCK GAME'.
(a) *Login.aspx*: Shows a 'Username' text input field. Below the field is a link 'or [register](#)'. To the right of the field is a 'Log In' button.
(b) *Registration.aspx*: Shows a 'Username' text input field. Below the field is a link 'or [login](#)'. To the right of the field is a 'Register' button. Above the 'Register' button are three dropdown menus for 'Date of Birth' with values '2014', 'November', and '17'.

Figure 2: Different kind of log-in errors

If he tries to access the other pages without being logged-in first, he is redirected to the *Login.aspx* page and a warn is shown to him.

Figure 3 is a screenshot of the *Login.aspx* page. It shows the 'Username' text input field, the 'or [register](#)' link, and the 'Log In' button. Below the 'Log In' button, there is a red error message that says 'You need to login'.

Figure 3: *Login.aspx* after a unauthorized access page attempt

In order to register the user has to choose:

- a username (only letters and/or digits, between 5 and 25);
- his birthday date;

both are mandatory and the user name has to be unique (since it is primary key in the database).

In order to log-in the user has to provide:

- only the user name.

If the user does not follow these instructions an error is shown.

The *Login.aspx* and the *Registration.aspx* pages are accessible only if the user is not logged-in: if he is and he tries to access them he is redirected to the *Default.aspx* page.

Figure 4 consists of two side-by-side screenshots of the *Registration.aspx* page. Both screenshots show the 'Username' and 'Date of Birth' fields, the 'or [login](#)' link, and the 'Register' button.
(a) Mandatory field error: The 'Username' field is empty. Below the 'Register' button, there is a red error message that says 'Username can't be blank'.
(b) Wrong format error: The 'Username' field contains the text 'sds'. Below the 'Register' button, there is a red error message that says 'Choose only letter/digits, at least five'.

Figure 4: Different validation controls

2.2 The Game

The game is located in the *Default.aspx* page. The game is self explaining since the rules are shown through the image: the user can press one of the five buttons corresponding to the Rock, Paper, Scissors, Lizard or Spock option and the result (win, lost, drawn) is immediately shown above them.

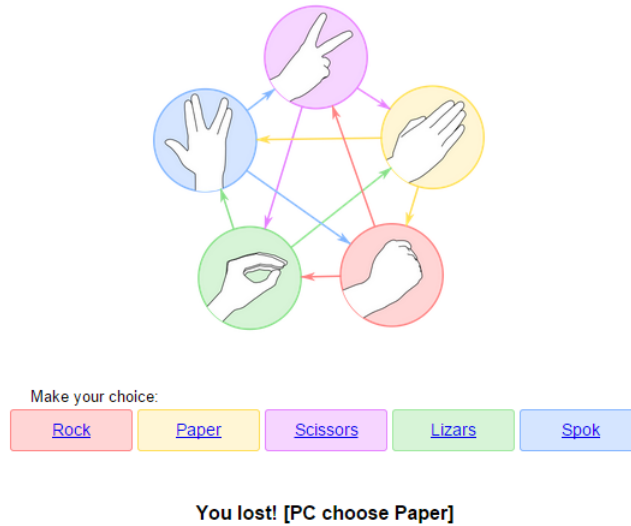


Figure 5: *Default.aspx*

From the *Default.aspx* page the user can reach:

- the *Statistics.aspx* page,
- the *UserAdmin.aspx* page,

or perform the logout from the web-application. In that case he is redirected to the *Login.aspx* page.

2.3 Statistics

Global statistics

User Name	Won games	Lost games	Drawn games	Total games
maoow	1	0	0	1
pippo	0	1	1	2
swanseauser	2	0	1	3
thewinner	0	0	0	0
toomanycoursework	0	0	0	0

Clear my results

(a) The statistics table

Global statistics

User Name	Won games	Lost games	Drawn games	Total games
maoow	1	0	0	1
Pippo	0	0	0	0
swanseauser	2	0	1	3
thewinner	0	0	0	0
toomanycoursework	0	0	0	0

Clear my results

All your statistics have been deleted

(b) Statistics deleted confirmation message

Figure 6: *Statistics.aspx*

On the *Statistics.aspx* page the user can see the statistics related to games played by all the users which can be ordered by column. He can also delete statistics only related to him. The changes to the table are immediately shown.

From the *Statistics.aspx* page the user can reach:

- the *Default.aspx* page,
- the *UserAdmin.aspx* page,

or perform the logout from the web-application. In that case he is redirected to the *Login.aspx* page.

2.4 The User Admin panel

The *UserAdmin.aspx* page is reachable through the link (the username) on the header of the page. The *UserAdmin.aspx* page gives the opportunity to the user to delete his account. If it happens, he is redirected to the *Login.aspx* page and a message is shown to him.

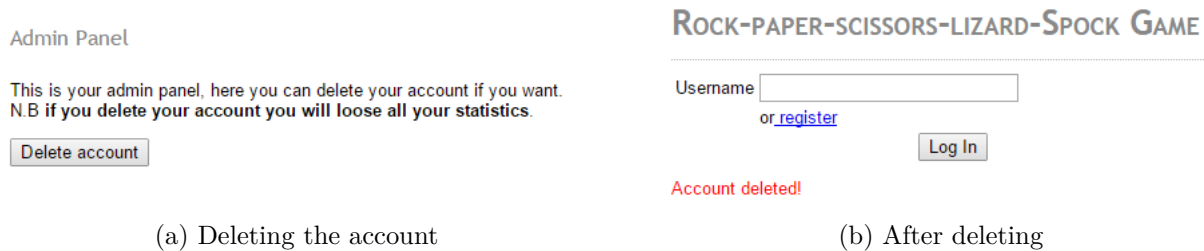


Figure 7: *UserAdmin.aspx*

2.5 Error pages

Finally if the user encounters any kind of error (such as he tries to access a page which does not exist) he is redirected to two defaults error pages:

- the *NotFound.aspx* page for a 404 error,
- the *Oooooohps.aspx* page for generic errors.

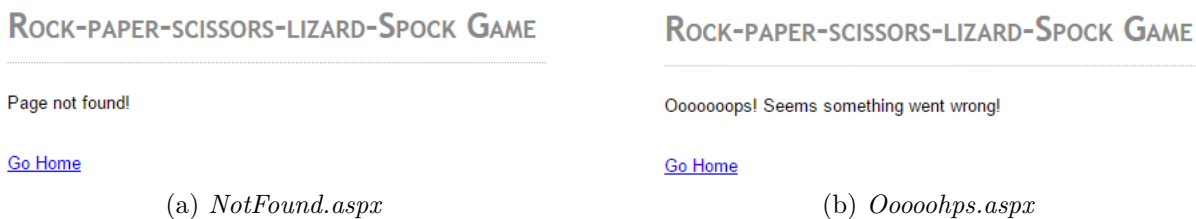


Figure 8: Error pages

2.6 Master pages

The web application is organized through two different master pages:

- the *Site.master* page template is used for those pages shown when the user is logged-in into the system. It performs some common functionalities such as the validation on session data, displaying the navigation menu and so on;
- the *Guest.master* page template is used for those pages shown when the user is not logged-in into the system.

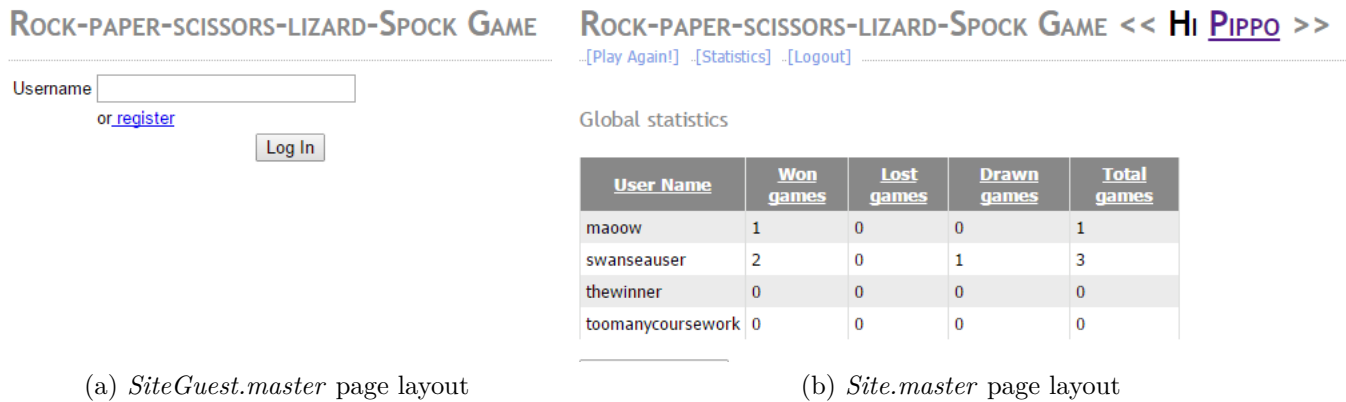


Figure 9: Master pages

2.6.1 The birthday function

When the current logged-in user is on his birthday a message is shown next to his username.

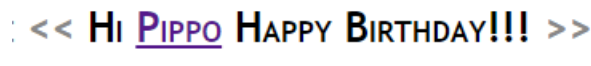


Figure 10: Happy birthday message

3 Classes

As shown in the UML diagram nine classes have been implemented in order to support the web application. With the exception of the `RPSLS_User` and of the `RPSLS_Date`, the classes can be grouped in three main categories:

- database management,
- session management,
- game management.

The system in the figure represents the web-application classes corresponding to each *.aspx* page.

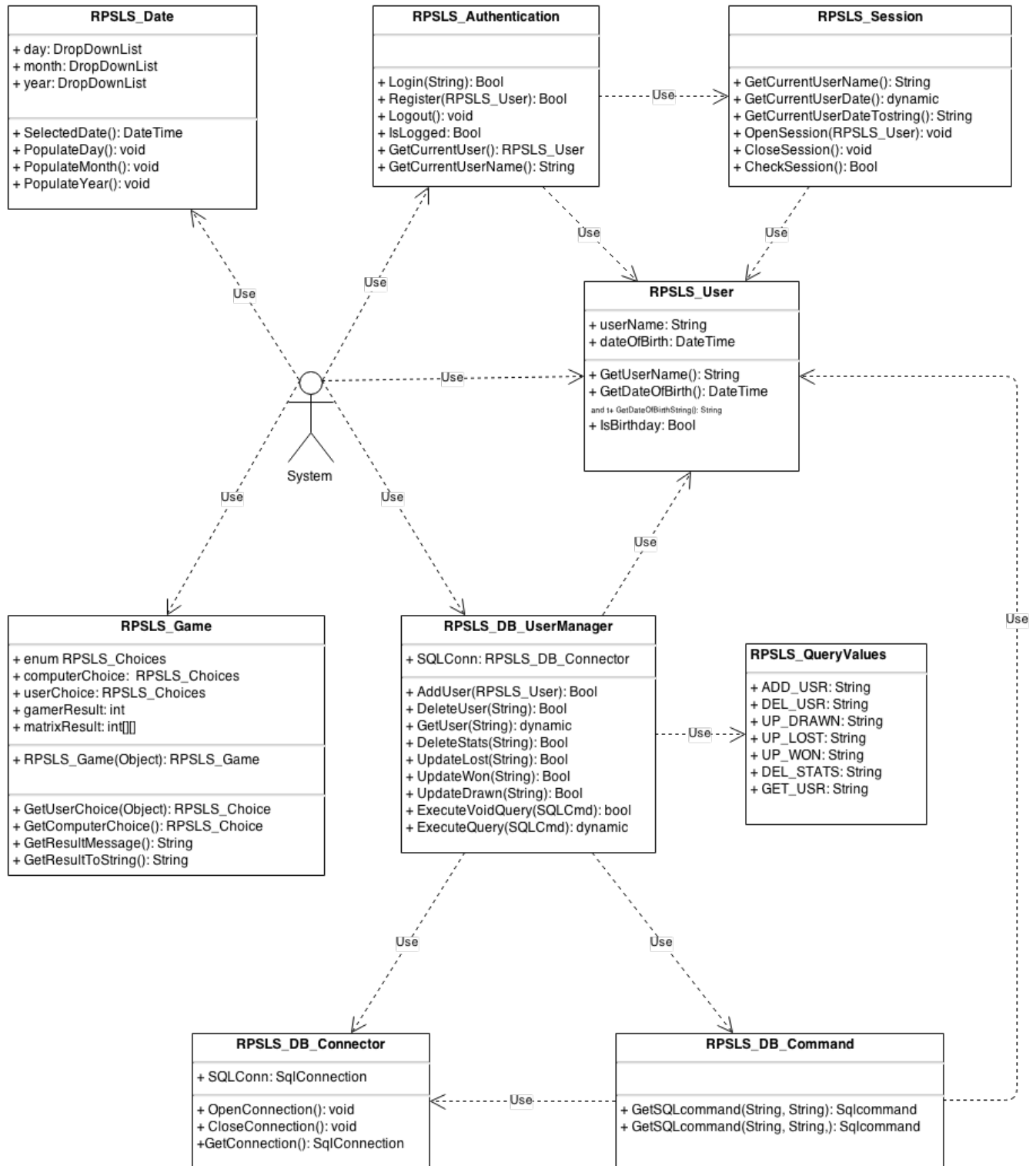


Figure 11: UML diagram

3.1 Database management

The three following classes manage the access to the database for inserting, updating, deleting or retrieving data. In particular:

- the `RPSLS_DB_Connector` class manages the connection with the database, opening, closing and returning the connection when requested;
- the `RPSLS_DB_Command` class creates and returns `SqlCommand` given an `SqlConnection` and the query in a `String` format;
- the `RPSLS_DB_UserManager` class allows the interaction between the database and the system: it is invoked by the web-application classes to perform actions on the database (update user statistics, adding a new user, etc.).

3.2 Session management

The two following classes manage the session functionalities such as the log-in and logout. In details:

- the `RPSLS_Session` class manages all aspects of the session variable environment (adding a new session variable, getting a current session variable, etc.);
- the `RPSLS_Authentication` class manages the interaction between the system and the `RPSLS_Session` (log-in, register, check if some user is logged-in in the system, get the current logged-in user, etc.).

3.3 Game management

To perform the game two classes are needed:

- the `RPSLS_Game` class performs the Rock-Paper-Scissors-Lizard-Spock game. Given the user and the computer choice as integer, it calculates the result using a matrix, and it stores the result inside a private field. Each position of the matrix corresponds to the player choice, according to the `RPSLS_Choices` integer value. The value refers to the result of the first player, in particular:

- -1 first player won;
- 0 drawn;
- 1 first player lost.

```
{ {0, -1, 1, 1, -1},  
  {1, 0, -1, -1, 1},  
  {-1, 1, 0, 1, -1},  
  {-1, 1, -1, 0, 1},  
  {1, -1, 1, -1, 0} };
```

So for example given the user and the computer choices respectively 2 and 3, the result is the number in the position `[2][3]` of the matrix, which is -1 and it means the user lost;

- the `RPSLS_Choices` class describes all the possible choice of the game as a `enum` class.

3.4 Other classes

Two other classes have been implemented:

- the `RPSLS_User` class models the user of the system. It contains only the name and the date of birth but it can be used to easily support the implementation of new functionalities (such as e-mail, password, etc.);
- the `RPSLS_Date[1]` allows the management of the `DropDownList` in the registration page. The class has been modified according to the actual needs.

3.5 Javascript

Four additional javascript functions have been implemented to support the correct operation of the web application:

- `PopulateDays()` and `AddOption()` manage the population of the `DropDownList` for the date of birth;
- `UserNameRegistrationChange()` and `UserNameLoginChange()` clear the error message when needed.

4 System requirements

The web application has been developed using Visual Studio 2012.

References

[1]<http://www.aspsnippets.com/Articles/Select-Day-Month-and-Year-Date-from-DropDownList-in-ASPNet.aspx>