Czech Technical University in Prague
Faculty of Electrical Engineering

- bachelor project -

Administration GUI for the SonkalWEB

Victoria Lopez Clemente

Supervisor: Ing. Pavel Loupal
Education Program: Computer Technology-structured studies
Option: Computer Science
June 2009

Acknowledgments

I would like to thank everybody who has supported me doing this project. I want to say thanks to my supervisor Ing. Pavel Loupal who has always supported me and helped me in the most difficult moments of the project. Also, I want to thank my friends, especially to Checa and Raulongo who have supported me in my bad moments and gave me strength for continue until I solve them. I want to thank my family too for giving me the chance of studying this year in Praha. Thank you very much.

- Index -

1.Project Definition

The goal of the project is to put into practice the knowledge of SQL and JSP. The project is a Sonkal web extension, where it will have two different kinds of users:

1.1 Scope of work

The project starts as a final degree work, in which we want to put into practice the knowledge of SQL and JSP, managing a database. The project is based primarily on the expansion of the judo Sonkal website for anonymous web users. The aim is to provide efficient management of the site, orientating it to a more dynamic website where users can interact with information stored in the database, creating, editing or deleting data.

In order to do this task, a level of security has been created, where only authorized users can access to it. Because of that, we have created two kinds of users that can access to the system for managing the database. This are the administrator and the editor.

The Editor will be the user with the responsibility of helping the administrator to maintain the website. The Editor will update the data or fill it with information . Some of the tasks of the editor will be: assign a person to a photograph, to a video, write some articles...

The administrator will be responsible to allow / disallow the editor users. On the other hand the administrator is allowed to do the same tasks as the editor, but also it can create, edit or delete data. The administrator will also be responsible of maintaining the relationship between tasks with the system and approve the changes proposed by the editors.

2. Analysis:

2.1 Short Description about the actors and Use Case

Web application, which will develop for manage data about Sonkal web, will be used for two different kinds of users. This application offers some services to be able to do different operations like update photography, edit data, read a list of people...

The Administrator and the Editor interact with the web through a Java application. This application is responsible for managing the data.

The tasks that can be performed by the Administrator are:

- Create a new Rank
- Read a list of ranks
- Edit an existing rank
- Delete an existing rank

- Create a new tool
- Read a list of tools
- Edit an existing tool
- Delete an existing tool

- Create a new event_type
- Read a list of event_type
- Edit an existing event_type
- Delete an existing event_type

- Create a new category
- Read a list of categories
- Edit an existing category
- Delete an existing category

- Create a new technique
- Read a list of techniques
- Edit an existing technique
- Delete an existing technique

- Create a new person
- Read a list of people
- Edit an existing person
- Delete an existing person

- ...

The tasks that will be held by the editor:

- Assign a person to a photograph.
- Assign a person to a video.
- Write an article for an event.
- Set type of a photograph.
- Enter competitors' results to a competition.
- ...

In each use case, the user will receive a response. Each request will get a valid value in the form of text message.

This message will indicate whether the operation being carried out has been implemented whitout problems, or an error has occurred.

Create a new rank

Allows adding a rank to the database of the website. This will be clearly identifiable in the list of ranks by an identifier, the colour, the English colour name and the name of the rank. Related to what I said before, the user may not be able to include an existing rank. This will occur when the colour or colours in English has been used by yet another introduced range.

Read a list of rank

Allows listing all ranks created, identifying for each rank: id, colour, English name and colour range.

Edit an existing rank

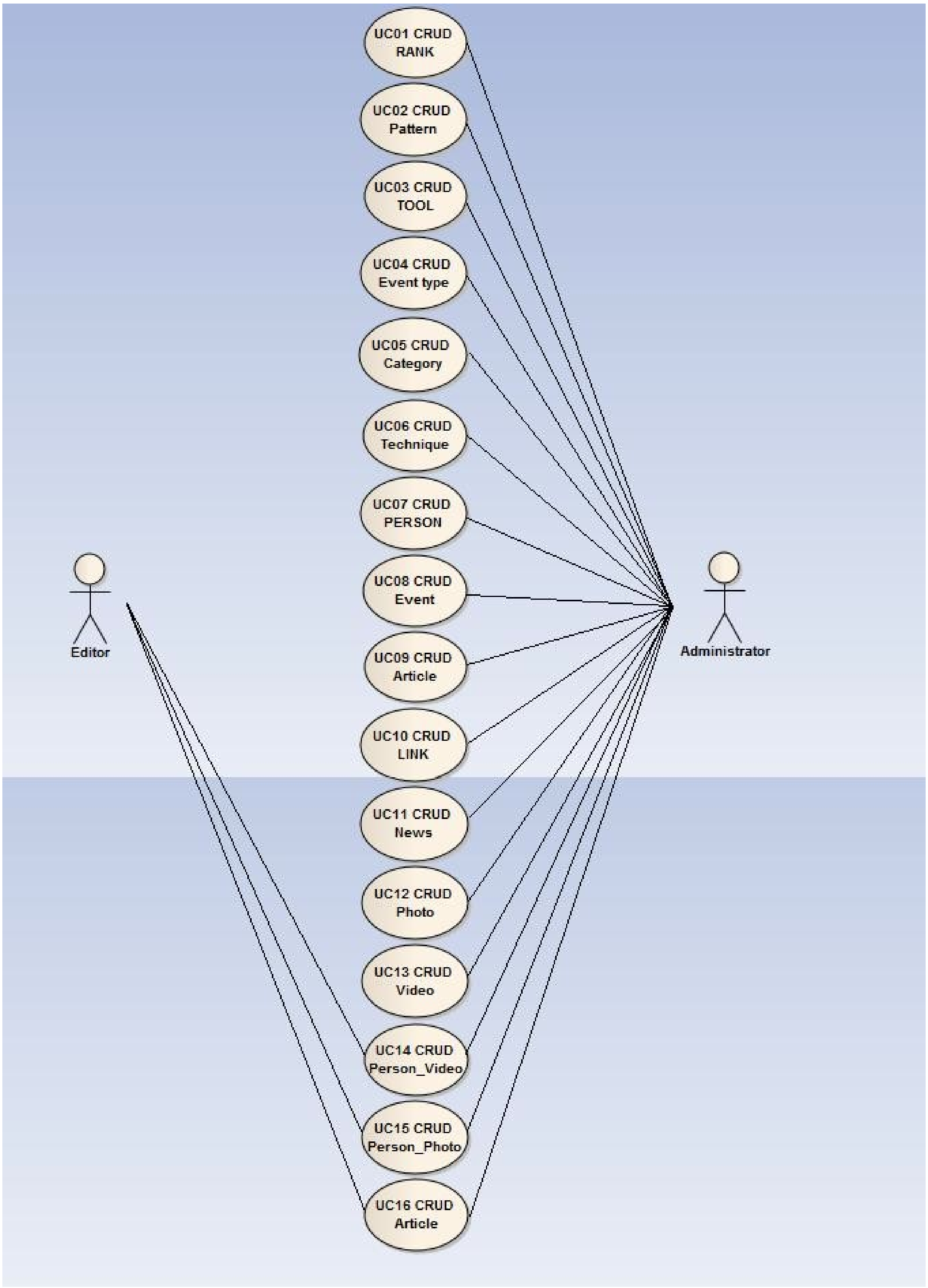
Allows editing an existent rank in the database. Also, the colour, the colour in English and the rank name can be edited. The id can't be edited though.

Delete an existing rank

Allows deleting one or more ranks of the database.

For each table of the explained before, the explanation is the same.

2.2 Use Cases Diagram



Some list-related Use Cases:

ID:	UC01
Name:	CRUD Rank
User Role	Administrator
Description:	User can create, retrieve, update or delete data in the rank table.

ID:	UC02
Name:	CRUD Pattern
User Role:	Administrator
Description:	User can create, retrieve, update or delete data in the pattern table.

ID:	UC03
Name:	CRUD Tool
User Role:	Administrator
Description:	User can create, retrieve, update or delete data in the tool table.

This use case is the same for this tables: event_type, category, technique, person, event, article, link news, photo and video.

ID:	UC14
Name:	CRUD Person Photo
User Role:	Editor and Administrator
Description:	User can create, retrieve, update or delete data in the person_photo table.

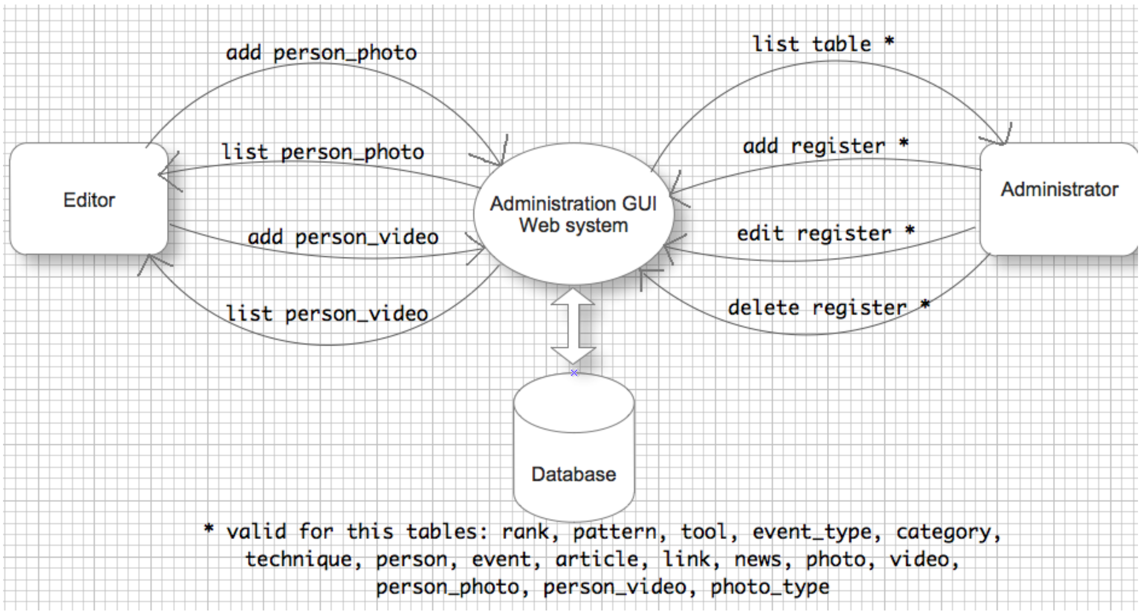
ID:	UC15
Name:	CRUD Person Video
User Role:	Editor and Administrator
Description:	User can create, retrieve, update or delete data in the person_video table.

This is the same for article and type photography

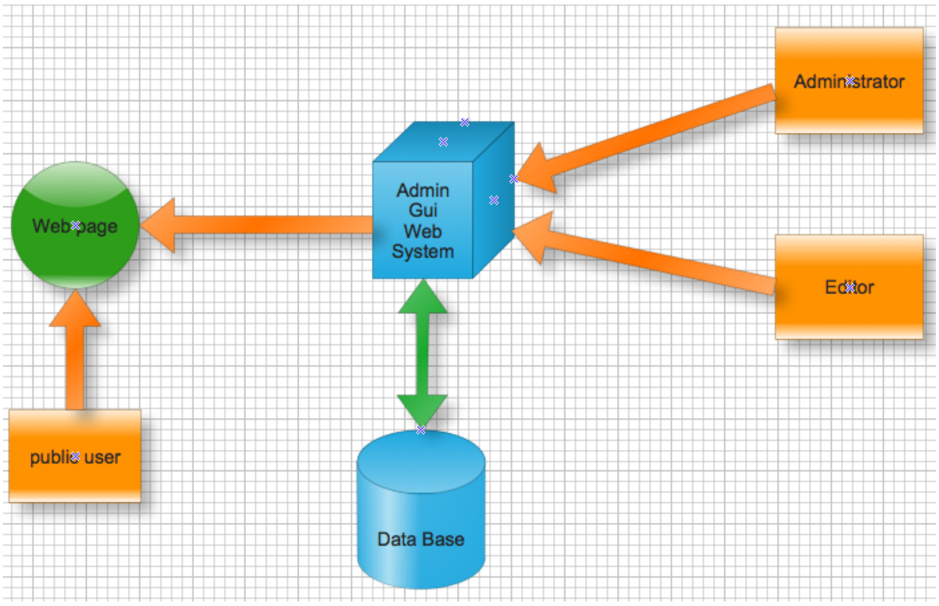
3. Design:

3.1 Context Diagram

This is the context diagram between the users (administrator and editor) and the Administration GUI. In this diagram we can see the differents functions its can use by te users.



This context diagram is more general, in it we can see easily how the public user can interact whit the web, and the administrator and editor interacts whit the database throght the administator GUI Web System.

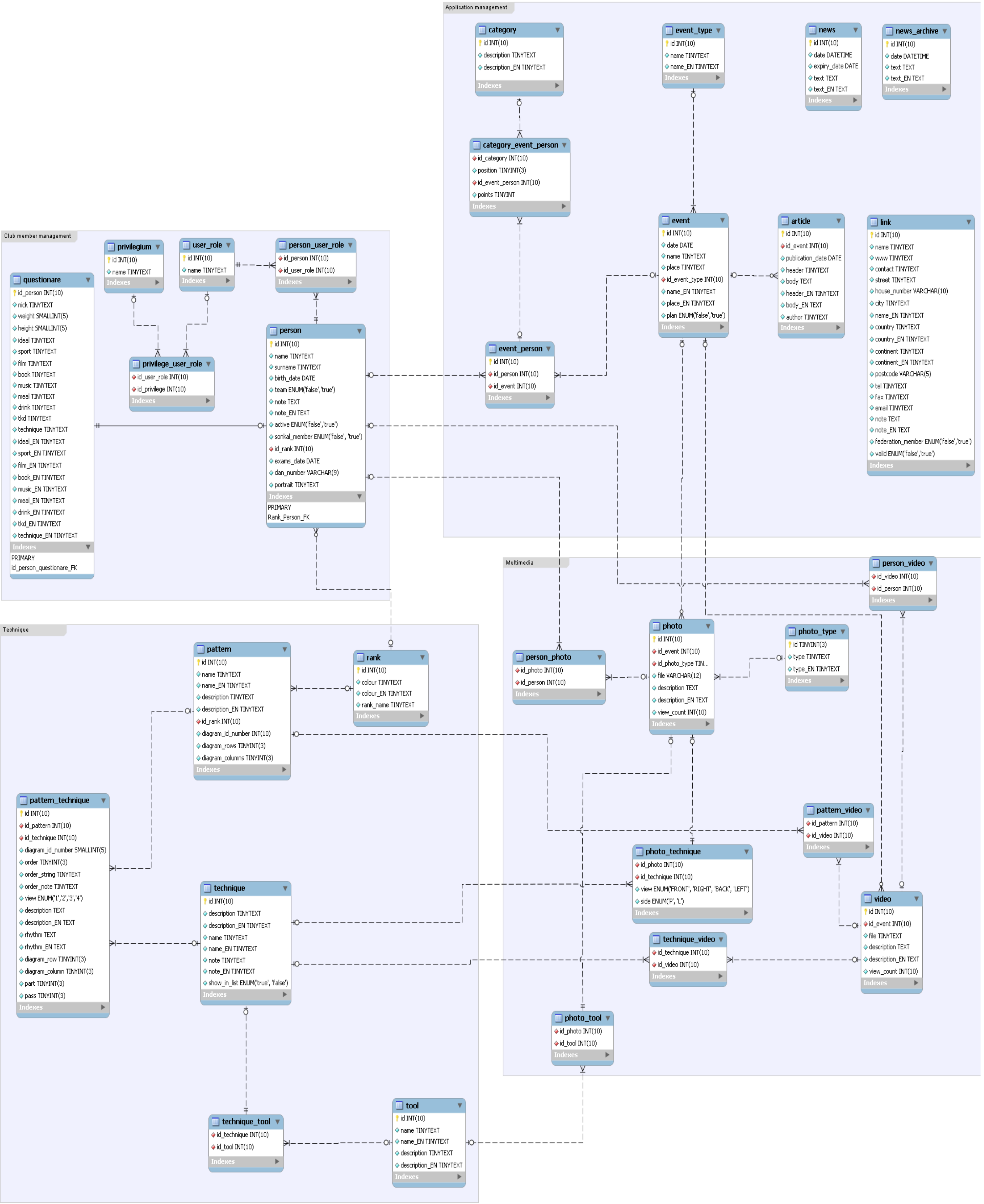


3.2 E-R Diagram

This diagram shows the data structure of the database and it contains 16 tables.

- Table of rank which stores the information about each existent rank in judo.
- Table of pattern which stores the information about the pattern of each rank.
- Table of tools which stores the information about the tools.
- Table of event_type. This stores the kind of all the types of judo events.
- Table of category which stores the description of each category.
- Table of techniques which explain each technique with the name and a description.
- Table of person. It stores all the information of the users. This table contains the user and the password of each user. This information is important to know which users are allowed to login.
- Table of event which stores the information about judo events.
- Table article which stores the articles created by the users.
- Table of links which stores the information of the most important judo webs and other’s interests.
- Table of news which stores the news created by the users.
- Table of photo. It contains the information about the photos.
- Table of video which stores the information about the videos.
- Table of person_video. In this table, the editors can assign a person to a video.
- Table of person_photo. In this table, the editors can assign a person to a photo.
- Table of photo_type. It contains the types of photos.

Administration GUI for SonkalWeb



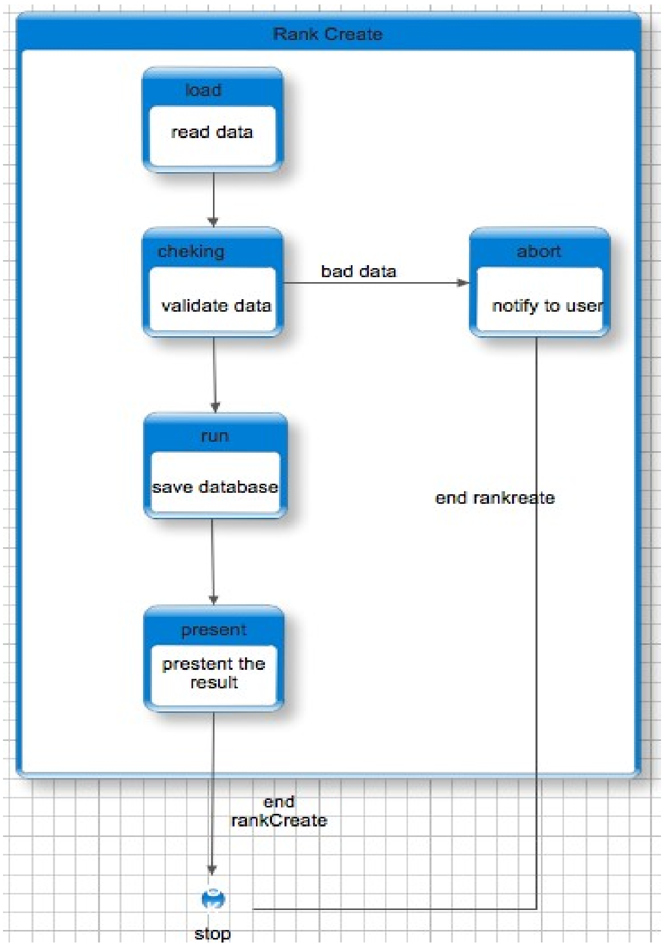
3.3 Web page design

When I designed the look of the web page application, I wanted to have two separate pages, one for the login and one for all the content. This second part is divided by 3 areas: One with the list of all the tables (in the left side), other with a menu for retrieving data (in the top), and finally, one more in the middle, which is related to the ones I explained before.

When choosing the colours, I specially selected colours which can be easily read alongside the characters. For this, I have a light blue background and gray characters. In order to make the menu buttons, I've chosen the characters in orange and a blue background.

3.4 Class Diagram

I decided to distribute application logic over several packages and classes. Classes are divided by several packages. Each package contains the information of all the classes used to manage each data of the table, and also to show it to the user.



4. Implementation of Database:

4.1 DB Schema

The application works with this database tables.
Each entity has a corresponding relation with the web Sonkal, which are essential for proper maintenance of the information on the web.

Rank Table

Name	**Type**
id	Integer
colour	Tinytext
colour_EN	Tinytext
rank_name	Tinytext

Pattern

Name	**Type**
id	Integer
name	Tinytext
name_EN	Tinytext
description	Tinytext
description_EN	Tinytext
id_rank	Integer
diagram_id_number	TinyInt
diagram_rows	TinyInt
diagram_columns	TinyInt

Tool

Name	**Type**
id	Integer
name	Tinytext
name_EN	Tinytext
description	Tinytext
description_EN	Tinytext

Event_type

Name	Type
id	Integer
name	Tinytext
name_EN	Tinytext

Category

Name	Type
Id	Integer
description	Tinytext
description_EN	Tinytext

Technique

Name	Type
id	Integer
description	Tinytext
description_EN	Tinytext
name	Tinytext
name_EN	Tinytext
note	Tinytext
note_EN	Tinytext
show_in_list	Enum (true, false)

Person

Name	Type
Id	Integer
name	Tinytext
username	Tinytext
password	Tinytext
surname	Tinytext
birht_date	Date
team	Enum(true, false)

note	Text
note_EN	Text
active	Enum(true,false)
sonkal_member	Enum(true,false)
id_rank	Integer
exams_date	Date
dan_number	Varchar
portrait	Tinytext

Event

Name	Type
id	Integer
date	Date
name	Tinytext
place	Tinytext
id_event_type	Integer
name_EN	Tinytext
place_EN	Tinytext
plan	Enum(true,false)

Article

Name	**Type**
id	Integer
id_event	Integer
publication_date	Date
header	Tinytext
body	Text
header_EN	Tinytext
body_EN	Text
author	Tinytext

Link

Name	**Type**
id	Integer

name	Tinytext
www	Tinytext
contact	Tinytext
street	Tinytext
house_number	Varchar
city	Tinytext
name_EN	Tinytext
country	Tinytext
country_EN	Tinytext
continent	Tinytext
continent_EN	Tinytext
postcode	Varchar
telf	Tinytext
fax	Tinytext
email	Tinytext
note	Text
note_EN	Text
federation_member	Enum(false,true)
valid	Enum(false,true)

News

Name	**Type**
id	Integer
date	DateTime
expiry_date	Date
text	Text
text_EN	Text

Photo

Name	**Type**
id	Integer
id_event	Integer
id_photo_type	Tinyint
file	Varchar

description	Text
description_EN	Text
viewcount	Integer

Video

Name	**Type**
id	Integer
id_event	Integer
file	Tinytext
description	Text
description_EN	Text
view_count	Integer

Person_photo

Name	**Type**
id_person	Integer
id_photo	Integer

Person_video

Name	**Type**
id_person	Integer
id_video	Integer

Photo_type

Name	**Type**
id	Integer
type	Tinytext
type_EN	Tinitext

4.2 Implementation of BD

A script of this table is follows:

Table rank:

```
CREATE  TABLE IF NOT EXISTS `rank` (
  `id` INT(10) NOT NULL ,
  `colour` TINYTEXT NOT NULL ,
  `colour_EN` TINYTEXT NOT NULL ,
  `rank_name` TINYTEXT NOT NULL ,
  PRIMARY KEY (`id`) )
ENGINE = InnoDB;
```

Table person:

```
CREATE  TABLE IF NOT EXISTS `person` (
  `id` INT(10) NOT NULL AUTO_INCREMENT ,
  `name` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
 `username` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
 `password` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `surname` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `birth_date` DATE NULL ,
  `team` ENUM('false','true') CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL,
  `note` TEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL DEFAULT NULL ,
  `note_EN` TEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL DEFAULT NULL ,
  `active` ENUM('false','true') NULL DEFAULT 'true' ,
  `sonkal_member` ENUM('false', 'true') NULL ,
  `id_rank` INT(10) NULL ,
  `exams_date` DATE NULL ,
  `dan_number` VARCHAR(9) NULL ,
  `portrait` TINYTEXT NULL ,
  PRIMARY KEY (`id`) ,
  INDEX `Rank_Person_FK` (`id_rank` ASC) ,
  CONSTRAINT `Rank_Person_FK`
    FOREIGN KEY (`id_rank` )
    REFERENCES `rank` (`id` )
    ON DELETE RESTRICT ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 304;
```

Table pattern:

```
CREATE  TABLE IF NOT EXISTS `pattern` (
  `id` INT(10) NOT NULL AUTO_INCREMENT ,
  `name` TINYTEXT NOT NULL ,
  `name_EN` TINYTEXT NULL ,
  `description` TINYTEXT NOT NULL ,
  `description_EN` TINYTEXT NULL ,
  `id_rank` INT(10) NOT NULL ,
  `diagram_id_number` INT(10) NOT NULL ,
  `diagram_rows` TINYINT(3) NOT NULL ,
  `diagram_columns` TINYINT(3) NOT NULL ,
  PRIMARY KEY (`id`) ,
  INDEX `Rank_Pattern_FK` (`id_rank` ASC) ,
  CONSTRAINT `Rank_Pattern_FK`
    FOREIGN KEY (`id_rank` )
    REFERENCES `rank` (`id` )
    ON DELETE RESTRICT ON UPDATE CASCADE)
ENGINE = InnoDB
```

Table tool:

```
CREATE  TABLE IF NOT EXISTS `tool` (
  `id` INT(10) NOT NULL AUTO_INCREMENT ,
  `name` TINYTEXT NOT NULL ,
  `name_EN` TINYTEXT NULL ,
  `description` TINYTEXT NOT NULL ,
  `description_EN` TINYTEXT NULL ,
  PRIMARY KEY (`id`) )
ENGINE = InnoDB
```

Table event_type:

```
CREATE  TABLE IF NOT EXISTS `event_type` (
  `id` INT(10) NOT NULL AUTO_INCREMENT ,
  `name` TINYTEXT NOT NULL ,
  `name_EN` TINYTEXT NOT NULL ,
  PRIMARY KEY (`id`) )
ENGINE = InnoDB;
```

Table category:

```
CREATE  TABLE IF NOT EXISTS `category` (
  `id` INT(10) NOT NULL AUTO_INCREMENT ,
  `description` TINYTEXT NOT NULL ,
  `description_EN` TINYTEXT NOT NULL ,
  PRIMARY KEY (`id`) )
ENGINE = InnoDB
```

Table event:

```
CREATE  TABLE IF NOT EXISTS `event` (
  `id` INT(10) NOT NULL AUTO_INCREMENT ,
  `date` DATE NOT NULL ,
  `name` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `place` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `id_event_type` INT(10) NOT NULL ,
  `name_EN` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL ,
  `place_EN` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL ,
  `plan` ENUM('false','true') NOT NULL DEFAULT 'false' ,
  PRIMARY KEY (`id`) ,
  INDEX `id_Event_type_FK` (`id_event_type` ASC) ,
  CONSTRAINT `id_Event_type_FK`
    FOREIGN KEY (`id_event_type` )
    REFERENCES `event_type` (`id` )
    ON DELETE RESTRICT
    ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 425
```

Table person

```
CREATE  TABLE IF NOT EXISTS `person` (
  `id` INT(10) NOT NULL AUTO_INCREMENT ,
  `name` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `surname` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `birth_date` DATE NULL ,
  `team` ENUM('false','true') CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' ,
  `note` TEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL DEFAULT NULL ,
  `note_EN` TEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL DEFAULT NULL ,
  `active` ENUM('false','true') NULL DEFAULT 'true' ,
  `sonkal_member` ENUM('false', 'true') NULL ,
  `id_rank` INT(10) NULL ,
  `exams_date` DATE NULL ,
  `dan_number` VARCHAR(9) NULL ,
  `portrait` TINYTEXT NULL ,
  PRIMARY KEY (`id`) ,
  INDEX `Rank_Person_FK` (`id_rank` ASC) ,
  CONSTRAINT `Rank_Person_FK`
    FOREIGN KEY (`id_rank` )
    REFERENCES `rank` (`id` )
    ON DELETE RESTRICT
    ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 304;
```

Table Tehcnique:

```
CREATE  TABLE IF NOT EXISTS `technique` (
  `id` INT(10) NOT NULL AUTO_INCREMENT ,
  `description` TINYTEXT NULL ,
  `description_EN` TINYTEXT NULL ,
  `name` TINYTEXT NOT NULL ,
  `name_EN` TINYTEXT NULL ,
  `note` TINYTEXT NULL ,
  `note_EN` TINYTEXT NULL ,
  `show_in_list` ENUM('true', 'false') NOT NULL ,
  PRIMARY KEY (`id`) )
ENGINE = InnoDB;
```

Table article:

```
CREATE  TABLE IF NOT EXISTS `article` (
  `id` INT(10) NOT NULL AUTO_INCREMENT ,
  `id_event` INT(10) NULL ,
  `publication_date` DATE NOT NULL ,
  `header` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `body` TEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `header_EN` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `body_EN` TEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `author` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL ,
  PRIMARY KEY (`id`) ,
  INDEX `id_Event_FK` (`id_event` ASC) ,
  CONSTRAINT `id_Event_FK`
    FOREIGN KEY (`id_event` )
    REFERENCES `event` (`id` )
    ON DELETE SET NULL
    ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 266;
```

Table link:

```
CREATE  TABLE IF NOT EXISTS `link` (
  `id` INT(10) NOT NULL AUTO_INCREMENT ,
  `name` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `www` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `contact` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL DEFAULT NULL ,
  `street` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL DEFAULT NULL ,
  `house_number` VARCHAR(10) CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL  ,
  `city` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL DEFAULT NULL ,
  `name_EN` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL DEFAULT NULL ,
  `country` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `continent` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `postcode` VARCHAR(5) CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL DEFAULT NULL ,
  `tel` TINYTEXT NULL DEFAULT NULL ,
  `fax` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL DEFAULT NULL ,
  `email` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL DEFAULT NULL ,
  `note` TEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL DEFAULT NULL ,
  `note_EN` TEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL DEFAULT NULL ,
  `federation_member` ENUM('false','true') NOT NULL DEFAULT 'false' ,
  `valid` ENUM('false','true') CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL  ,
  PRIMARY KEY (`id`) )
ENGINE = MyISAM
AUTO_INCREMENT = 206
```

Table news:

```
CREATE  TABLE IF NOT EXISTS `news` (
  `id` INT(10) NOT NULL AUTO_INCREMENT ,
  `date` DATETIME NOT NULL ,
  `expiry_date` DATE NOT NULL ,
  `text` TEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `text_EN` TEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL ,
  PRIMARY KEY (`id`) )
ENGINE = MyISAM
```

Table photo:

```
CREATE  TABLE IF NOT EXISTS `photo` (
  `id` INT(10) NOT NULL AUTO_INCREMENT ,
  `id_event` INT(10) NULL ,
  `id_photo_type` TINYINT(3) NOT NULL ,
  `file` VARCHAR(12) CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `description` TEXT NULL ,
  `description_EN` TEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL ,
  `view_count` INT(10) NOT NULL DEFAULT '0' ,
  PRIMARY KEY (`id`) ,
  INDEX `id_Event_Photo_FK` (`id_event` ASC) ,
  INDEX `id_Photo_theme_FK` (`id_photo_type` ASC) ,
  CONSTRAINT `id_Event_Photo_FK`
    FOREIGN KEY (`id_event` )
    REFERENCES `event` (`id` )
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
  CONSTRAINT `id_Photo_theme_FK`
    FOREIGN KEY (`id_photo_type` )
    REFERENCES `photo_type` (`id` )
    ON DELETE RESTRICT ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 29722;
```

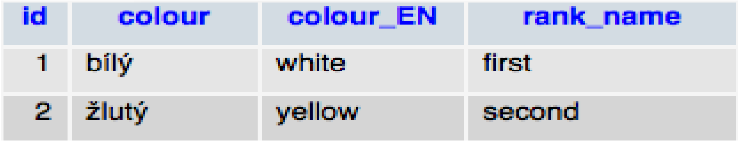
Table video:

```
CREATE  TABLE IF NOT EXISTS `video` (
  `id` INT(10) NOT NULL AUTO_INCREMENT ,
  `id_event` INT(10) NULL ,
  `file` TINYTEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NOT NULL ,
  `description` TEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL ,
  `description_EN` TEXT CHARACTER SET 'cp1250' COLLATE 'cp1250_czech_cs' NULL ,
  `view_count` INT(10) NOT NULL DEFAULT '0' ,
  PRIMARY KEY (`id`) ,
  INDEX `id_Event_Video_FK` (`id_event` ASC) ,
  CONSTRAINT `id_Event_Video_FK`
    FOREIGN KEY (`id_event` )
    REFERENCES `event` (`id` )
    ON DELETE RESTRICT
    ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 331;
```

4.2.1 Example of Information in Database

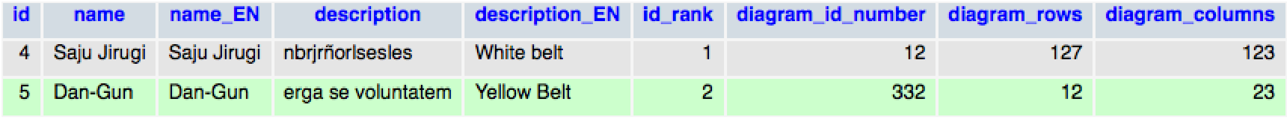
The database we have is big, so in order to understand them better, I'm about to show a simple example of some information stored in the database:

Rank



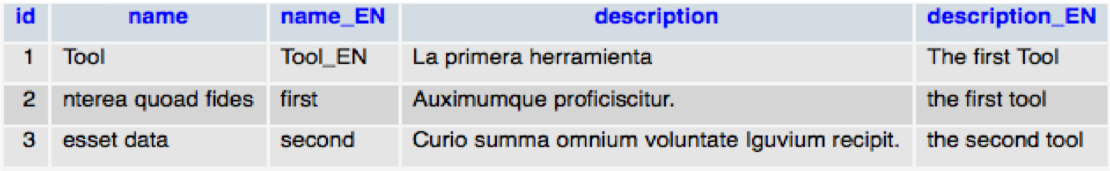
id	colour	colour_EN	rank_name
1	bílý	white	first
2	žlutý	yellow	second

Pattern



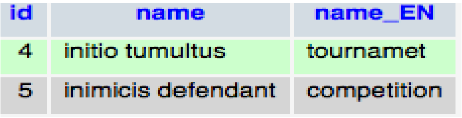
id	name	name_EN	description	description_EN	id_rank	diagram_id_number	diagram_rows	diagram_columns
4	Saju Jirugi	Saju Jirugi	nbrjrñorlsesles	White belt	1	12	127	123
5	Dan-Gun	Dan-Gun	erga se voluntatem	Yellow Belt	2	332	12	23

Tool



id	name	name_EN	description	description_EN
1	Tool	Tool_EN	La primera herramienta	The first Tool
2	nterea quoad fides	first	Auximumque proficiscitur.	the first tool
3	esset data	second	Curio summa omnium voluntate Iguvium recipit.	the second tool

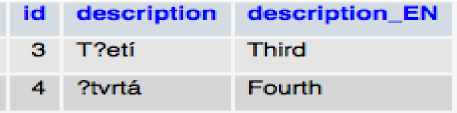
Event Type



id	name	name_EN
4	initio tumultus	tournamet
5	inimicis defendant	competition

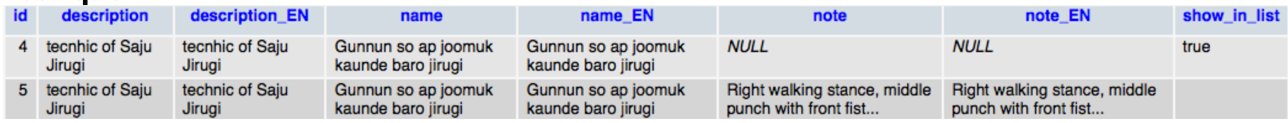
Administration GUI for SonkalWeb

Category



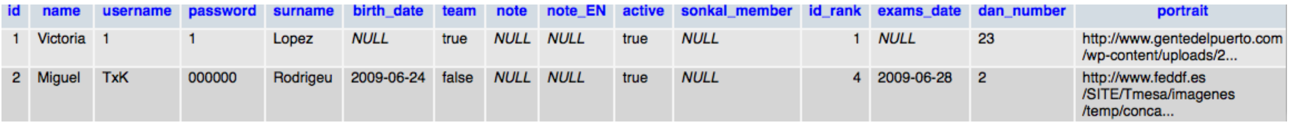
id	description	description_EN
3	T?etí	Third
4	?tvrtá	Fourth

Tecnique



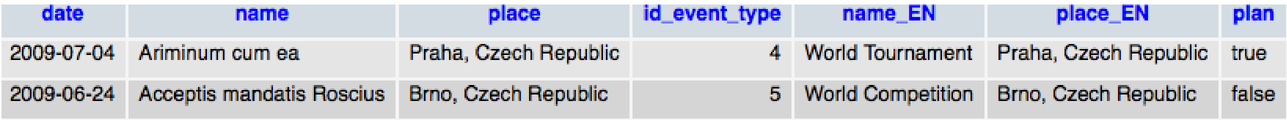
id	description	description_EN	name	name_EN	note	note_EN	show_in_list
4	tecnhic of Saju Jirugi	tecnhic of Saju Jirugi	Gunnun so ap joomuk kaunde baro jirugi	Gunnun so ap joomuk kaunde baro jirugi	*NULL*	*NULL*	true
5	tecnhic of Saju Jirugi	technic of Saju Jirugi	Gunnun so ap joomuk kaunde baro jirugi	Gunnun so ap joomuk kaunde baro jirugi	Right walking stance, middle punch with front fist...	Right walking stance, middle punch with front fist...	

Person



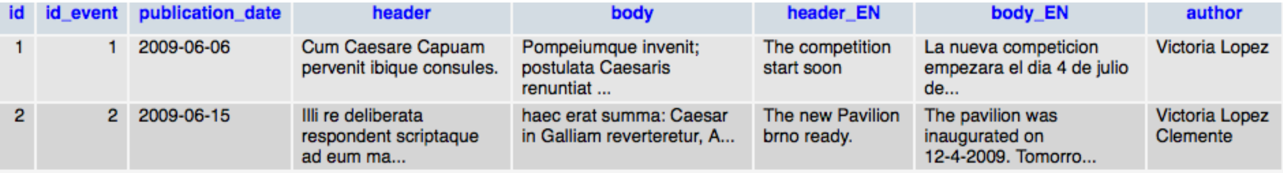
id	name	username	password	surname	birth_date	team	note	note_EN	active	sonkal_member	id_rank	exams_date	dan_number	portrait
1	Victoria	1	1	Lopez	*NULL*	true	*NULL*	*NULL*	true	*NULL*	1	*NULL*	23	http://www.gentedelpuerto.com /wp-content/uploads/2...
2	Miguel	TxK	000000	Rodrigeu	2009-06-24	false	*NULL*	*NULL*	true	*NULL*	4	2009-06-28	2	http://www.feddf.es /SITE/Tmesa/imagenes /temp/conca...

Event



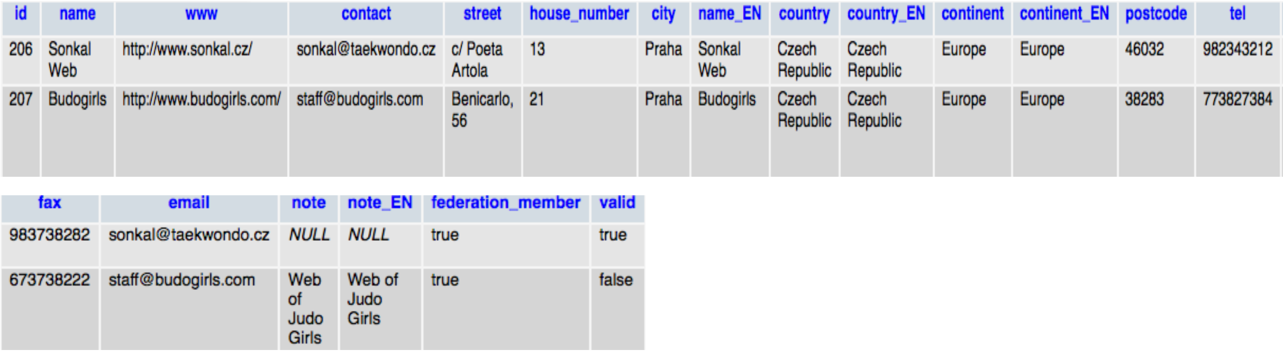
date	name	place	id_event_type	name_EN	place_EN	plan
2009-07-04	Ariminum cum ea	Praha, Czech Republic	4	World Tournament	Praha, Czech Republic	true
2009-06-24	Acceptis mandatis Roscius	Brno, Czech Republic	5	World Competition	Brno, Czech Republic	false

Article



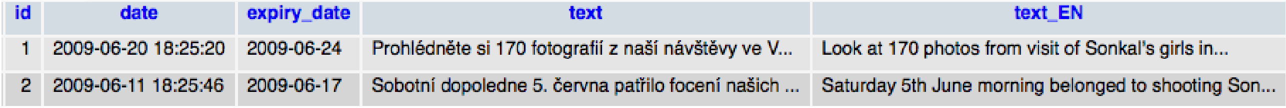
id	id_event	publication_date	header	body	header_EN	body_EN	author
1	1	2009-06-06	Cum Caesare Capuam pervenit ibique consules.	Pompeiumque invenit; postulata Caesaris renuntiat ...	The competition start soon	La nueva competicion empezara el dia 4 de julio de...	Victoria Lopez
2	2	2009-06-15	Illi re deliberata respondent scriptaque ad eum ma...	haec erat summa: Caesar in Galliam reverteretur, A...	The new Pavilion brno ready.	The pavilion was inaugurated on 12-4-2009. Tomorro...	Victoria Lopez Clemente

Link



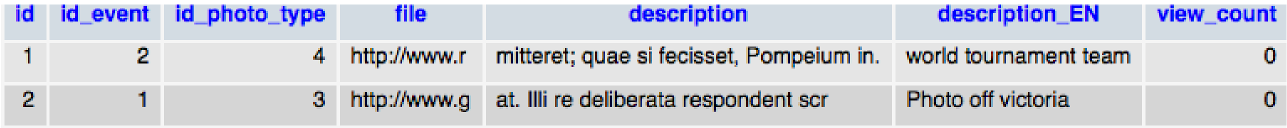
id	name	www	contact	street	house_number	city	name_EN	country	country_EN	continent	continent_EN	postcode	tel	fax	email	note	note_EN	federation_member	valid
206	Sonkal Web	http://www.sonkal.cz/	sonkal@taekwondo.cz	c/ Poeta Artola	13	Praha	Sonkal Web	Czech Republic	Czech Republic	Europe	Europe	46032	982343212	983738282	sonkal@taekwondo.cz	*NULL*	*NULL*	true	true
207	Budogirls	http://www.budogirls.com/	staff@budogirls.com	Benicarlo, 56	21	Praha	Budogirls	Czech Republic	Czech Republic	Europe	Europe	38283	773827384	673738222	staff@budogirls.com	Web of Judo Girls	Web of Judo Girls	true	false

News



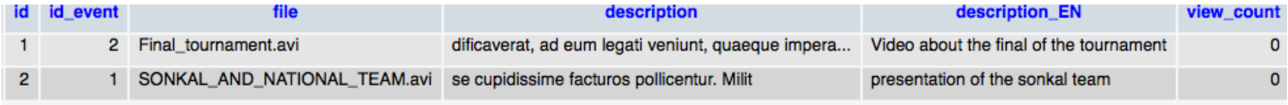
id	date	expiry_date	text	text_EN
1	2009-06-20 18:25:20	2009-06-24	Prohlédněte si 170 fotografií z naší návštěvy ve V...	Look at 170 photos from visit of Sonkal's girls in...
2	2009-06-11 18:25:46	2009-06-17	Sobotní dopoledne 5. června patřilo focení našich ...	Saturday 5th June morning belonged to shooting Son...

Photo



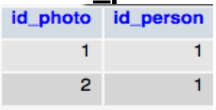
id	id_event	id_photo_type	file	description	description_EN	view_count
1	2	4	http://www.r	mitteret; quae si fecisset, Pompeium in.	world tournament team	0
2	1	3	http://www.g	at. Illi re deliberata respondent scr	Photo off victoria	0

Video



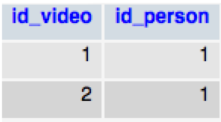
id	id_event	file	description	description_EN	view_count
1	2	Final_tournament.avi	dificaverat, ad eum legati veniunt, quaeque impera...	Video about the final of the tournament	0
2	1	SONKAL_AND_NATIONAL_TEAM.avi	se cupidissime facturos pollicentur. Milit	presentation of the sonkal team	0

Person_photo



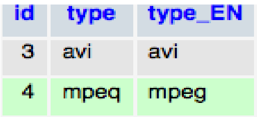
id_photo	id_person
1	1
2	1

Person_video



id_video	id_person
1	1
2	1

Photo_type



id	type	type_EN
3	avi	avi
4	mpeq	mpeg

5. Software Implementation

5.1 Coding

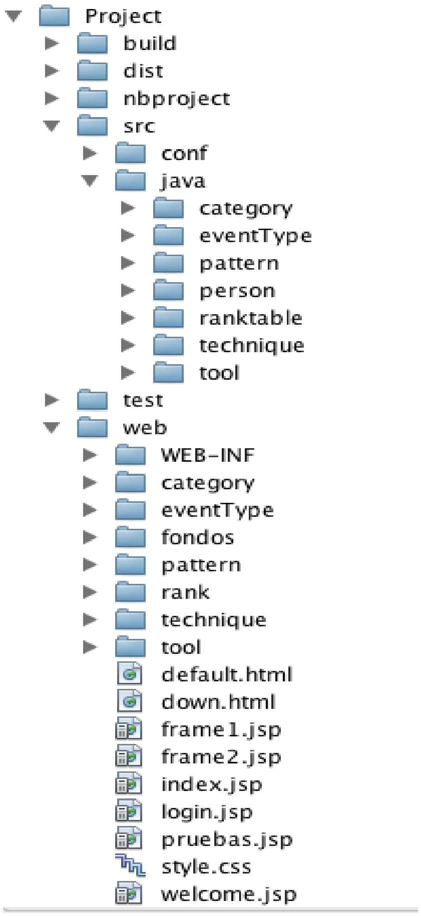
In this chapter I will describe some interesting implementation details of web application.

This application is implemented and coded in JSP, PHPMyAdmin and localhost server in order to visit the web page and test it, so we send a request trough http protocol requesting the information in the web server (MAMP Server) which is used for manage the website locally.

For each package I created the correspondent web pages, which each of them has its own correspondent function into the package in order to manage the data.

I created a standard folder structure in my application. The structure is like this:

- **/src /java /*”table_name”*---->** contains all java packages, different for each table
- **/web** ----> contains cascading style and the web pages commons for all the tables
- **/web /*"table_name"*** ----> contains the pages different for each table



```
▼ Project
  ▶ build
  ▶ dist
  ▶ nbproject
  ▼ src
    ▶ conf
    ▼ java
      ▶ category
      ▶ eventType
      ▶ pattern
      ▶ person
      ▶ ranktable
      ▶ technique
      ▶ tool
  ▶ test
  ▼ web
    ▶ WEB-INF
    ▶ category
    ▶ eventType
    ▶ fondos
    ▶ pattern
    ▶ rank
    ▶ technique
    ▶ tool
      default.html
      down.html
      frame1.jsp
      frame2.jsp
      index.jsp
      login.jsp
      pruebas.jsp
      style.css
      welcome.jsp
```

Let’s imagine that an user wants to insert a register. The application will work like this: The server responds by sending a form that the user has to fill. Once completed, the user will have to click on the button and the application will send the form through the browser to the server where it will be stored.

I will explain with further detail each menu option to list, create, edit or delete a registry entry (this example uses the table rank)
(in this example uses the table rank)

List Rank

"List rank" lists all the registry entries stored in the database. Now, the user have to click "List" button and then the server, as soon as receiving this request, it will send through the browser the list of all the stored rank.
The function is this:

```
public static String getRanks() { //save all the registers from the table in a LinkedList
    LinkedList<Rank> RanksList = new LinkedList<Rank>();
    try {
      Class.forName(className).newInstance();
      Connection conexion = DriverManager.getConnection(host + dataBase, user, password);
      Statement st = conexion.createStatement();
      ResultSet rs = st.executeQuery("select * from rank"); //select all registers from DB
      while (rs.next()) // while exist more register I save in ranklist
      {
        Rank rank = new Rank();
        rank.setId(rs.getInt("id"));
        rank.setColour(rs.getString("colour")); //the name between the "" is the name of the column
        rank.setColour_en(rs.getString("colour_en"));
        rank.setRank_name(rs.getString("rank_name"));
        RanksList.add(rank); //add rank to the list
      }
      rs.close(); //close query sql
      st.close();// close statment
      conexion.close(); // close conexion with DB


    } catch (Exception e) {
      e.printStackTrace();
    }
    return rankResult(RanksList);
  }
```

New rank

New rank consists in a function where a new rank is created. Also, a new form is created too, in which the user will have to enter the data that will be checked with java later on. Finally, once the user has fill the form and the data has been checked, the new data is stored into the database.

```
public static void rankCreate(String colour, String colour_EN, String rank_name) {
    //si apanyamos el rank_create.jsp esta funcion no nos hace falta para nada.
    try {
      if (colour != null || colour_EN != null || rank_name != null || !"".equals(colour) || !"".equals(colour_EN)
|| !"".equals(rank_name)) {

        Class.forName(className);//.newInstance();
        Connection conexion = DriverManager.getConnection(host + dataBase, user, password);
        Statement st = conexion.createStatement();
        st.executeUpdate("INSERT INTO rank (id , colour ,colour_EN , rank_name)" +
                  " VALUES ( NULL , '" + colour + "', '" + colour_EN + "', '" + rank_name + "')");
            st.close();// close statment
            conexion.close();
         }// close conexion with DB
      } catch (Exception e) {
         e.printStackTrace();
      }
   }
```

Edit Rank

Edit rank is a function where the user edits an existing rank. The user introduces the parameters in a form to search for it and it is sent to the server through the browser. Then, the server will respond with all the records similar to the search entered before. The user chooses the registry entry to edit, and he will enter the new values through a form. Once the values are checked, these are sent to the server and are stored in the database.

The function responsible for this work would be this:

```
  public static void rankUpdate(int id, String newColour, String newColour_EN, String newRank_name) {
      try {
         Class.forName(className).newInstance();
         Connection conn = DriverManager.getConnection(host + dataBase, user, password);
         Statement st = conn.createStatement();
         ResultSet rst = st.executeQuery(
               "select * " +
               "from rank " +
               "where id = '" + id + "'");

         rst.next();
         Rank rank = new Rank();

         rank.setColour(rst.getString("colour"));
         rank.setColour_en(rst.getString("colour_en"));
         rank.setRank_name(rst.getString("rank_name"));

         if ("".equals(newColour) || newColour == null) { //if newColour is empty, it save the old value
            newColour = rank.colour;
         }
         if (newColour_EN.equals("") || newColour_EN == null) {//if newColour_EN is empty, it save the old value

            newColour_EN = rank.colour;
         }
         if (newRank_name.equals("") || newRank_name == null) { //if newRank_Name is empty, it save the old value

            newRank_name = rank.rank_name;
         }

         rst.close();
         st.close();
         conn.close();

         Class.forName(className).newInstance();
         // jdbc:mysql://[servidor]:[puerto]/[base de datos], usuario, contraseña
         Connection con = DriverManager.getConnection(host + dataBase, user, password);
      // create Statement
      Statement stmt = con.createStatement();
      // create the SQL query

      stmt.executeUpdate( //          modifying the parameters to new values.
          "UPDATE rank " +
          "SET colour = '" + newColour + "'," +
          "colour_EN = '" + newColour_EN + "'," +
          "rank_name = '" + newRank_name + "' " +
          "WHERE id = '" + id + "'");


      stmt.close();
      // close conexion
      con.close();

    } catch (Exception e) {
      e.printStackTrace();
    }
  }
```

Delete Rank

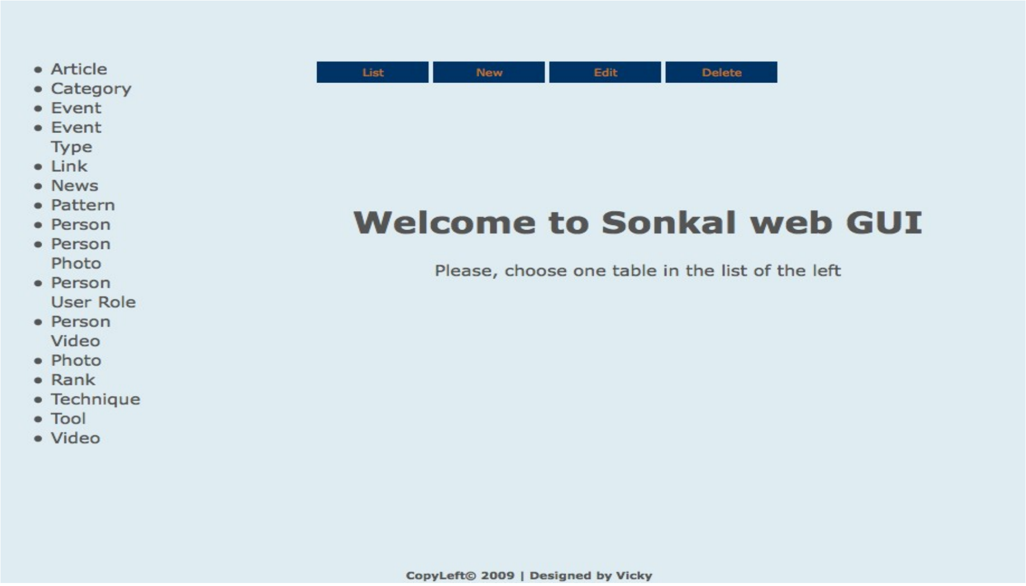
“Delete Rank” is responsible for deleting a register or multiple registers in the database. It works in a similar way to edit. Through a form, the user enters the parameters to find the record or records to delete. Through the button “Accept”, the server receives these queries and returns the corresponding list of rank. The user through some checkboxes, selects the records to delete. The server receives this list and then deletes the selected registers and updates the database.
The function responsible for deleting the records would be this:

```
public static String rankDelete(String[] id, boolean confirm) {
    String result = "";
    try {

      Class.forName(className).newInstance();
      Connection conn = DriverManager.getConnection(host + dataBase, user, password);

      if (!confirm) { //list the registers

        result = "<h2>Are this the ranks that you want to delete?</h2>" +
            "<table >" +
            "<tr>" +
            "<td> id </td>" +
            "<td> color </td>" +
            "<td> color_en </td>" +
            "<td> rank_name</td>" +
            "</tr>";

        int temp;

        for (int i = 0; i < id.length; i++) { //read the array
          Statement st = conn.createStatement();
          temp = Integer.parseInt(id[i]);
          ResultSet rs = st.executeQuery(
              "SELECT * FROM rank WHERE id = '" + temp + "'"); //search the regsiter wich id is the same that
id[i]
          rs.next();

          Rank rank = new Rank(); //save the register in the rankList
          rank.setId(rs.getInt("id"));
          rank.setColour(rs.getString("colour"));
          rank.setColour_en(rs.getString("colour_en"));
          rank.setRank_name(rs.getString("rank_name"));

          result = result + "<tr>"; //save the rank in the string result
          result = result + "<td>" + rank.getId() + "</td>";
          result = result + "<td>" + rank.getColour() + "</td>";
          result = result + "<td>" + rank.getColour_en() + "</td>";
          result = result + "<td>" + rank.getRank_name() + "</td>";
          result = result + "</tr>";

          rs.close(); //close query sql
          st.close();// close statment
        }

        result = result +  //form for accept or edit the list of registers to delete
                "</table> " +
                "<form method=\"post\" action=\"deleteRank.jsp\" name=\"ConfirmDeleteRank\">" +
                "<input type=\"submit\" name=\"accept\" value=\"Accept\">" +
                "<input type=\"submit\" name=\"edit\" value=\"Edit\">" +
                "</form>";

        conn.close(); // close conexiion with DB

      }

      Class.forName(className).newInstance();
      Connection connect = DriverManager.getConnection(host + dataBase, user, password);

      Statement stt = connect.createStatement();

      if (confirm) { // if the list of registers is correct, confirm is true
        for (int i = 0; i < id.length; i++) { //read the array
          stt.executeUpdate( //delete the register which id is the same that id[i]
              "DELETE FROM rank " +
              "WHERE id = '" + id[i] + "'");
        }
        result = "Save Succesfull";
      }
      stt.close();
      connect.close();
    } catch (Exception e) {
      e.printStackTrace();
    }
    return result;
  }
```

6. Realisation

In this part I’m going to explain how the GUI works from the point of view of an authorised user.

This index shows the different tables, and the menus, where the buttons are to list the tables, create new data, edit, or delete data.



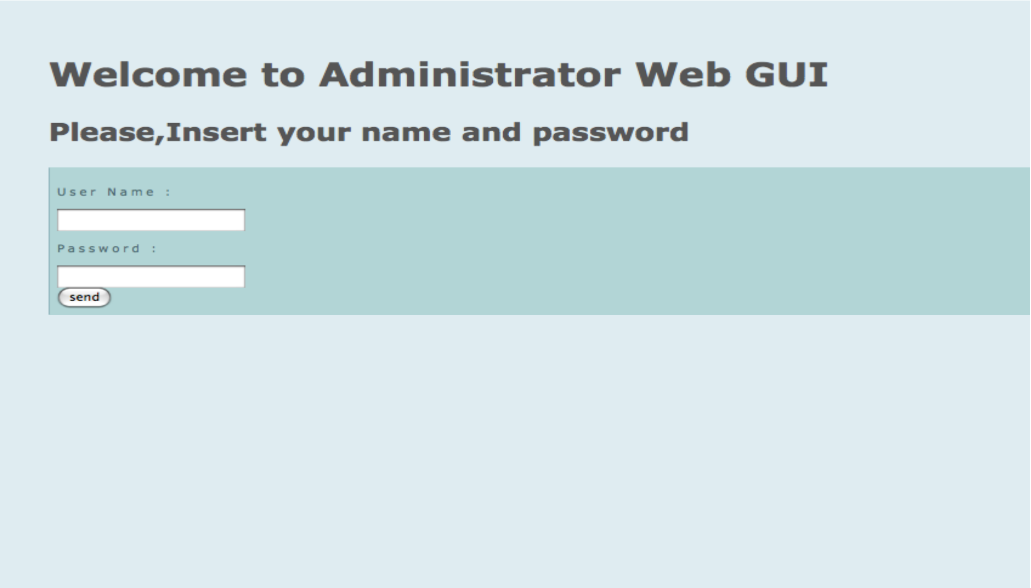
Welcome to Sonkal web GUI

Please, choose one table in the list of the left

CopyLeft© 2009 | Designed by Vicky

Login

This page is the security page, where only the administrator and editor can enter. Here, we must write the username and the password to authorize us to enter this section.

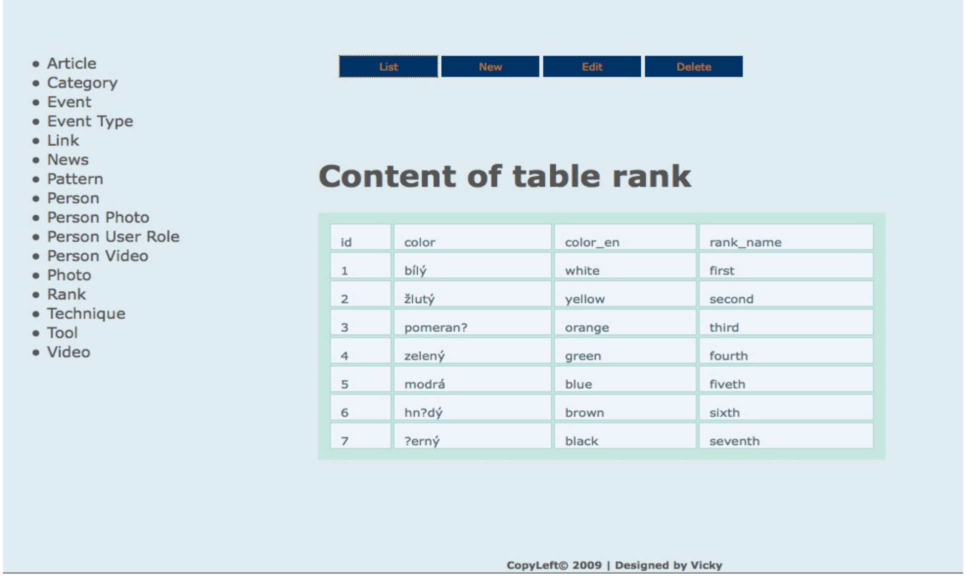


Welcome to Administrator Web GUI

Please,Insert your name and password

List Ranks

With this page the user can know what the data in the database is.



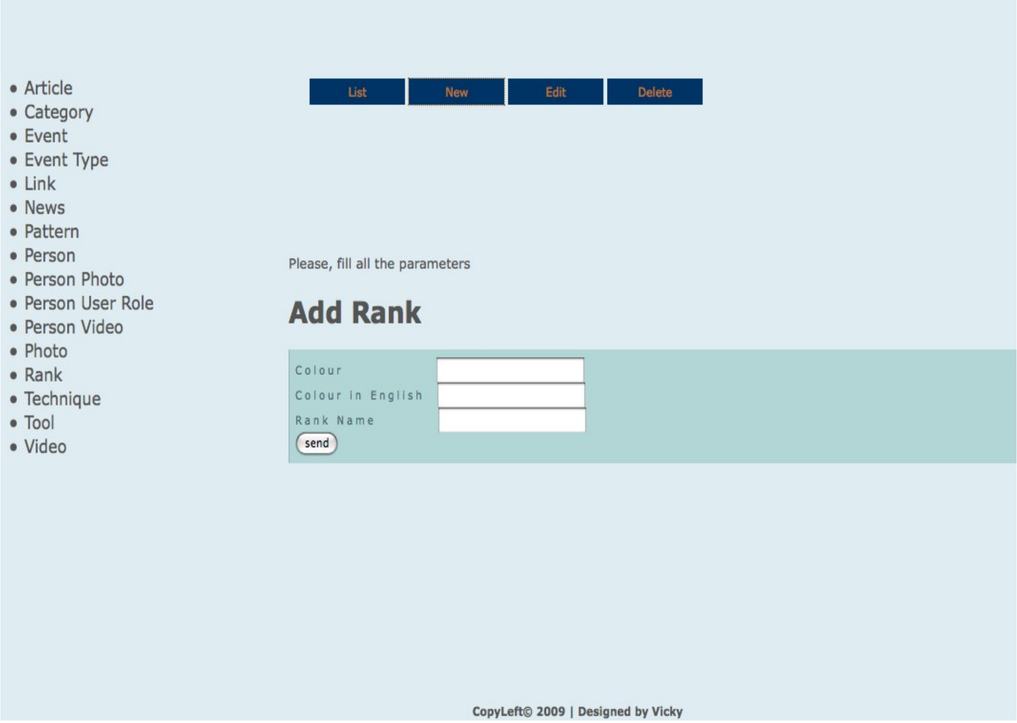
Content of table rank

id	color	color_en	rank_name
1	bílý	white	first
2	žlutý	yellow	second
3	pomeran?	orange	third
4	zelený	green	fourth
5	modrá	blue	fiveth
6	hn?dý	brown	sixth
7	?erný	black	seventh

CopyLeft© 2009 | Designed by Vicky

New Rank page

This page shows a form where the user can enter the data of the new rank. When all the parameters are written, the user may click in the “create button” and the new rank is saved in the database.



Please, fill all the parameters

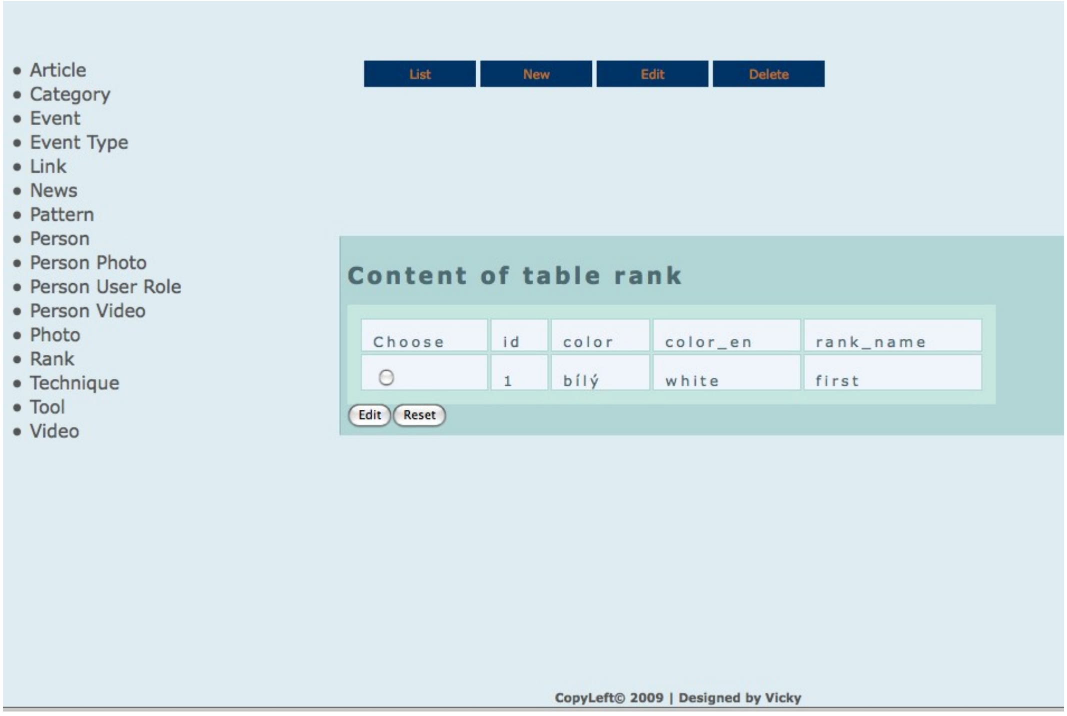
Add Rank

Colour
Colour in English
Rank Name

CopyLeft© 2009 | Designed by Vicky

Edit Rank page

When a user wants to change some parameters in the database, it can be changed with a form to search the registry, choosing between different options...

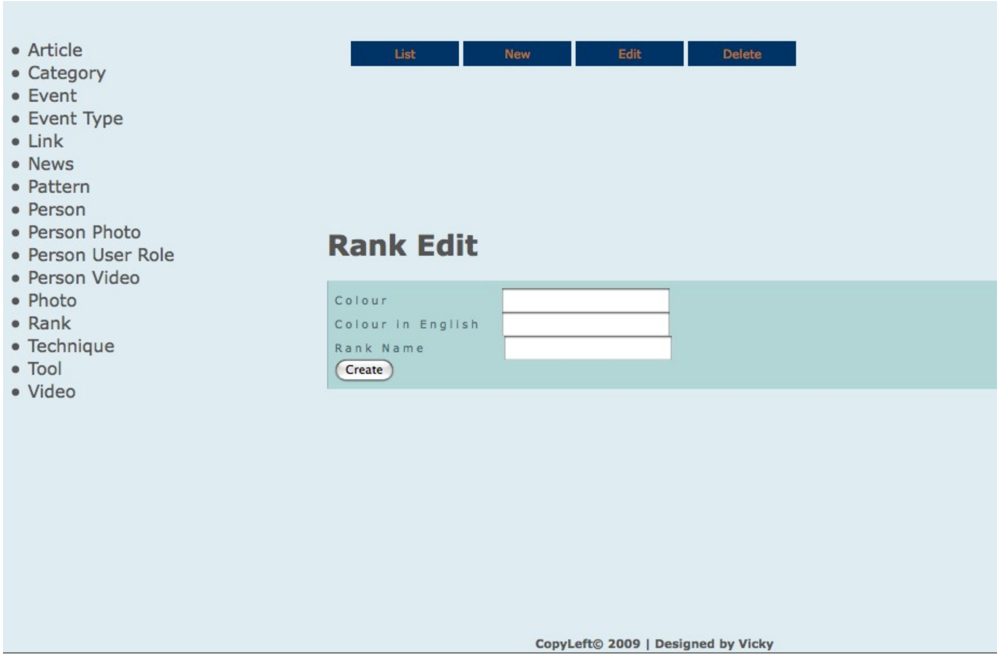


Content of table rank

Choose	id	color	color_en	rank_name
○	1	bílý	white	first

and finally entering the news values to the registry.
User may click in the button “edit” to perform the operation and save the new data in the database.

Delete Rank page



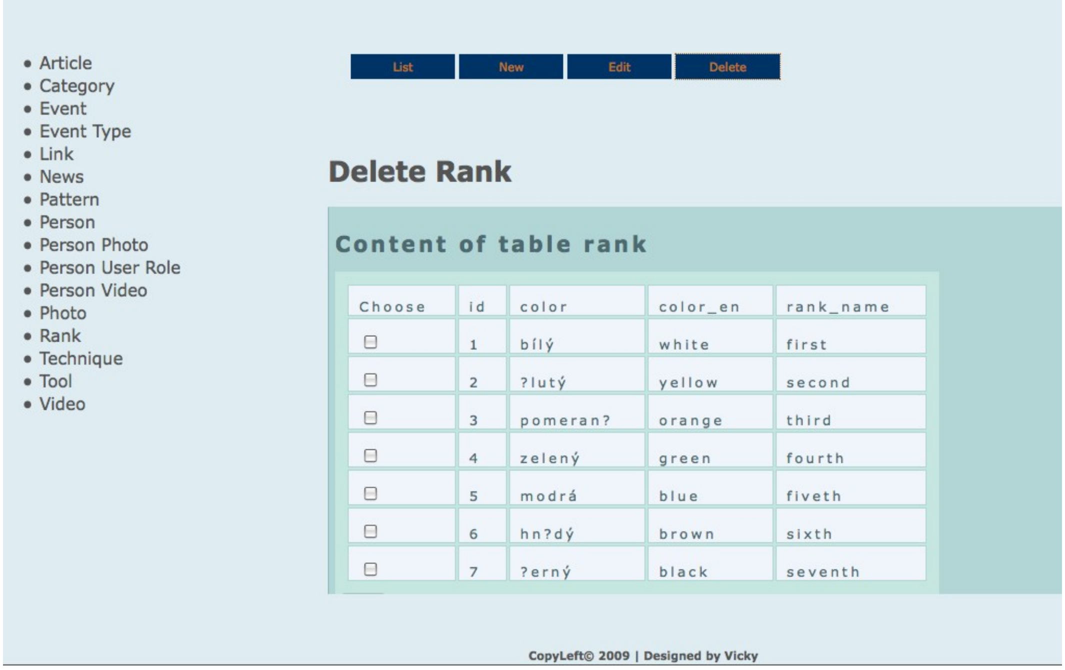
Rank Edit

Colour

Colour in English

Rank Name

This page shows the list of all the ranks with a checkbox, one per each registry. The user can choose all the registries that want to delete. This page redirects to another..



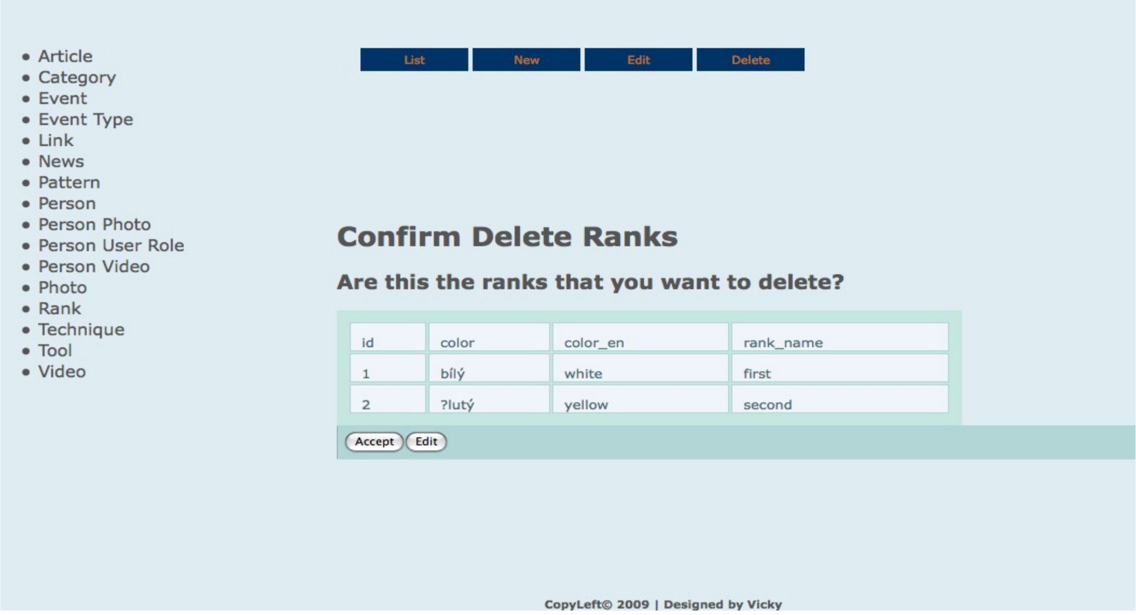
Delete Rank

Content of table rank

Choose	id	color	color_en	rank_name
□	1	bílý	white	first
□	2	?lutý	yellow	second
□	3	pomeran?	orange	third
□	4	zelený	green	fourth
□	5	modrá	blue	fiveth
□	6	hn?dý	brown	sixth
□	7	?erný	black	seventh

CopyLeft© 2009 | Designed by Vicky

… to accept the election. In order to finish the operation, the user may click in the Accept button or the user may click in the Edit button in order to choose different registers.



Confirm Delete Ranks

Are this the ranks that you want to delete?

id	color	color_en	rank_name
1	bílý	white	first
2	?lutý	yellow	second

CopyLeft© 2009 | Designed by Vicky

7 Conclusion

When I started planning the project, its schedules, etc... It seemed to have a lot of time to do my work development and implementation, but when I really started, I realize that I was wrong as I didn’t had much time.

To have a complete control over the whole structure of the website I was about to develop, a part from seeing several similar pages, the challenging part was learning to use software I never used and understanding HTM and JSP languages. I’ve never done anything like this before with JSP and my knowledge and design in HTML were very limited.

This allowed me to understand how a web page works. The creation of the structure of the website began testing it a few times, and finally getting into the definitive.

For each CRUD menu item of the web page, I must find lots of information, testing it every time I changed it, slowly, until the project worked in each of the steps I’ve made.

I hope my project can teach people who read it, or at least give ideas that are clearly enough to understand.

I’m planning to expand this project and improve it as far as my possibilities and capabilities allow me to do it, learning and evolving into this branch of computer science.

8. Bibliography

- Tutorials of SQL and JSP
  - http://www.htmlpoint.com/sql/sql_02.htm
  - http://sherekan.com.ar/2008/06/27/introduccion-a-jsp/
  - http://www.jsptut.com/

- Introduction to servers:
  - http://geneura.ugr.es/~jmerelo/JSP/

- GlassFish page
  - https://glassfish.dev.java.net/

- Some pages wich help me with the code:
  - http://www.nabble.com/Tomcat-derby-JSP-%22java.sql.SQLException:-No-suitable-driver%22-td20776666.html
  - http://suhasjavablog.wordpress.com/2009/06/20/a-sample-jsp-program-to-fetch-the-information-from-request-parameter/
  - http://codingcoding.com/JSP/Tips_and%20Tutorials-185-0.php

- Some pages wich help me with the diagrams:
  - http://www.mitecnologico.com/Main/DiagramasEntidadRelacionER
  - http://msdn.microsoft.com/en-us/library/bb521236.aspx
  - http://www.visualcase.com/tutorials/class-diagram.htm
  - http://www.dcc.uchile.cl/~psalinas/uml/modelo.html
  - http://www.mailxmail.com/curso-sistemas-informativos-diseno-creacion-2/diagrama-contexto-determinacion-tareas-sistema

- Software:
  - http://tomcat.apache.org/
  - http://www.eclipse.org/
  - http://www.netbeans.org/
  - http://www.mamp.info/en/index.html