Databases 2 @ PoliMi

optional project - A.A. 2020/2021

- Elena Righini
- Federico Romeo
- Francesco Puddu

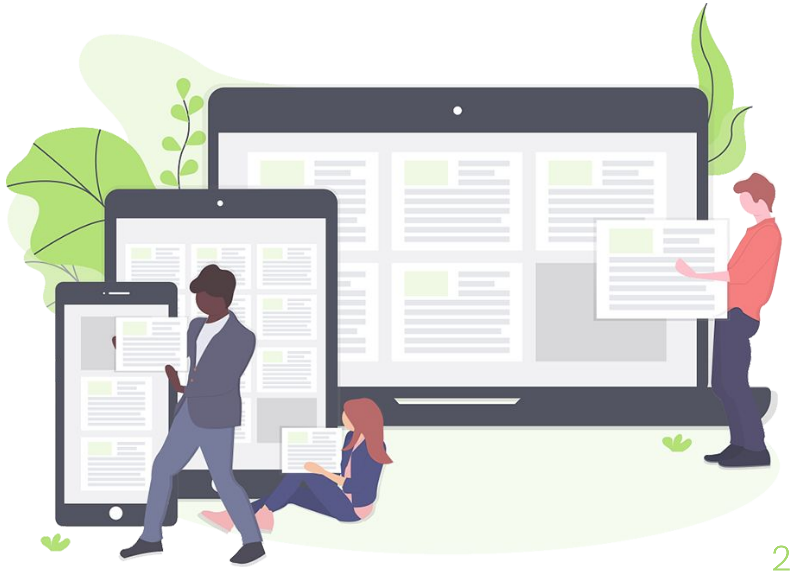
Gamified Marketing Application

Introduction

Our application for Gamified Marketing has been developed as a Java Web project, and is structured according to the specifications of **JavaEE**.

From the implementation point of view, the system is composed of **3 main modules**:

- A "web module" for the user presentation layer
- A "web module" for the admin presentation layer
- An "EJB module"



Overview

We used:

- **Thymeleaf** as template engine for ui

Among the additional technologies there are:

- **JavaScript** to add some dynamism
- **CSS** to enhance user experience

For the development, we relied on

- **Intellij IDEA** as IDE
- **MySQL** to manage our database

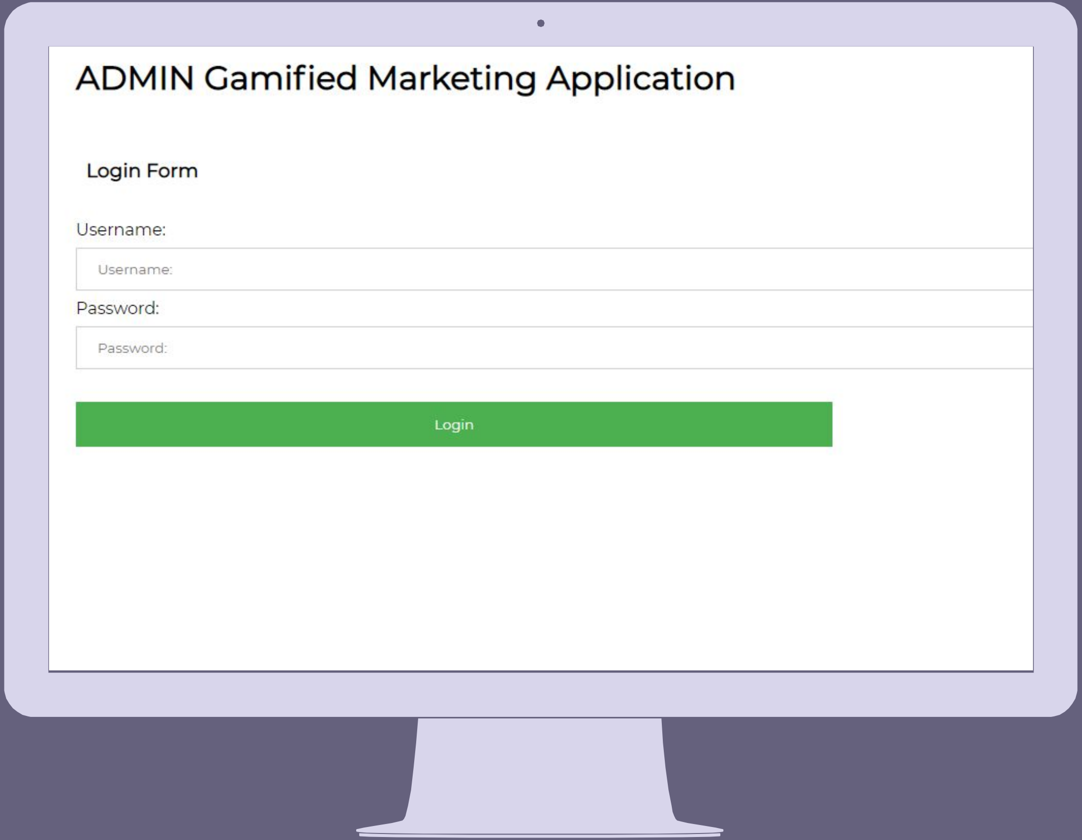
Frontend Overview

Here is how our project is going to be visualized

Admin login

Initial page to perform a login for an user with admin privileges

index.html


ADMIN Gamified Marketing Application

Login Form

Username:

Username:

Password:

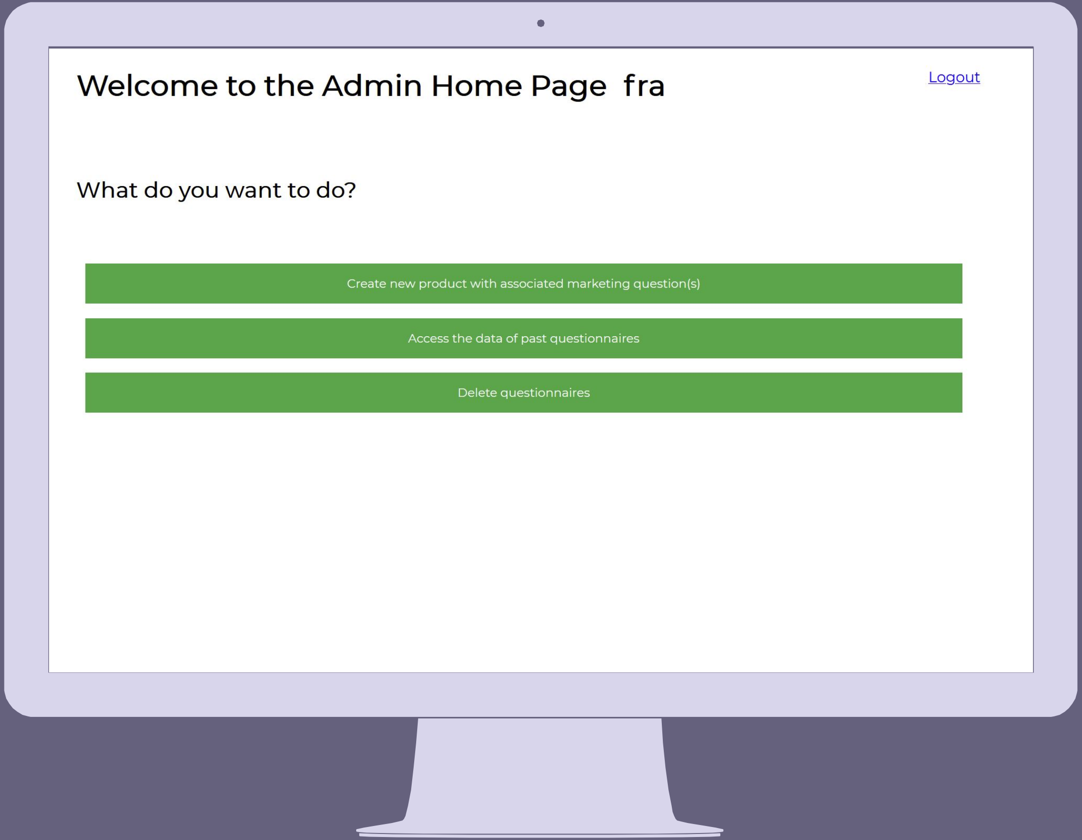
Password:

Login

Admin home

Home page for an user with admin privileges. From here he can choose to create a new product with associated marketing questions, delete questionnaires, or inspect old questionnaires

home_admin.html


Welcome to the Admin Home Page fra

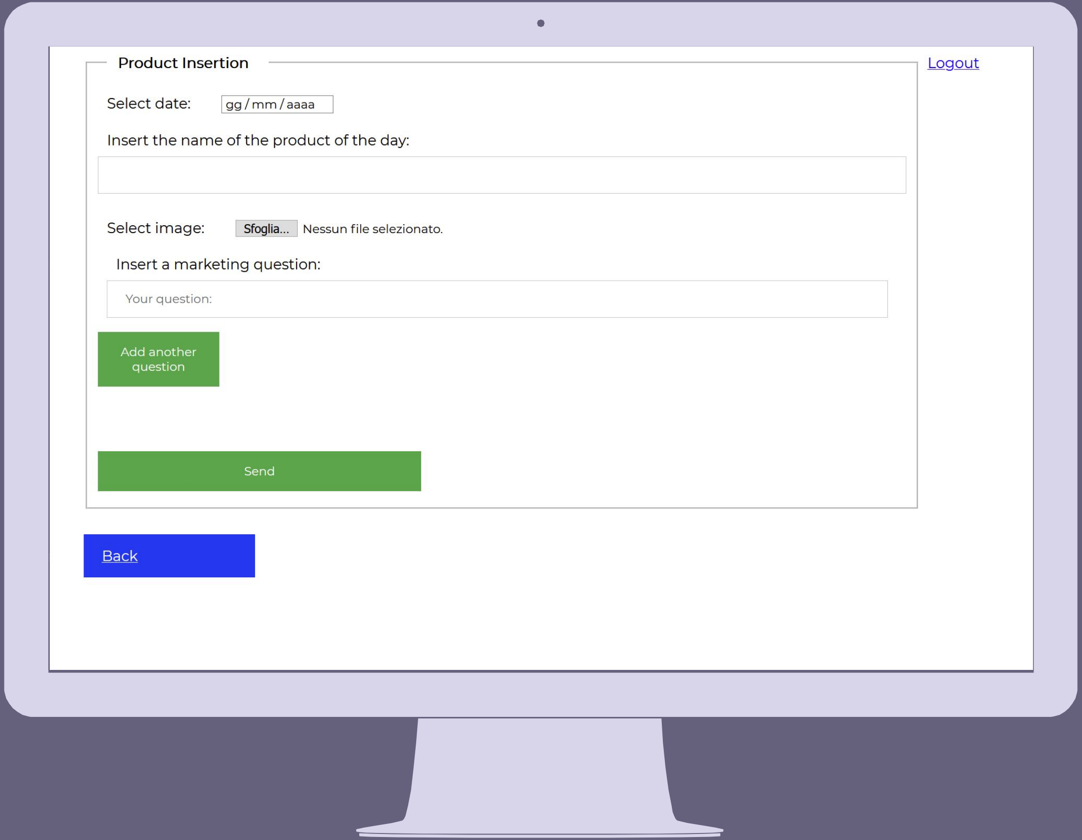
What do you want to do?

Logout
Create new product with associated marketing question(s)
Access the data of past questionnaires
Delete questionnaires

Creation Page

Page in which an admin can insert a new product by specifying date, name, questions, and inserting an image

creation.html


Product Insertion

Logout

Select date: gg / mm / aaaa

Insert the name of the product of the day:

Select image: Sfoglia... Nessun file selezionato.

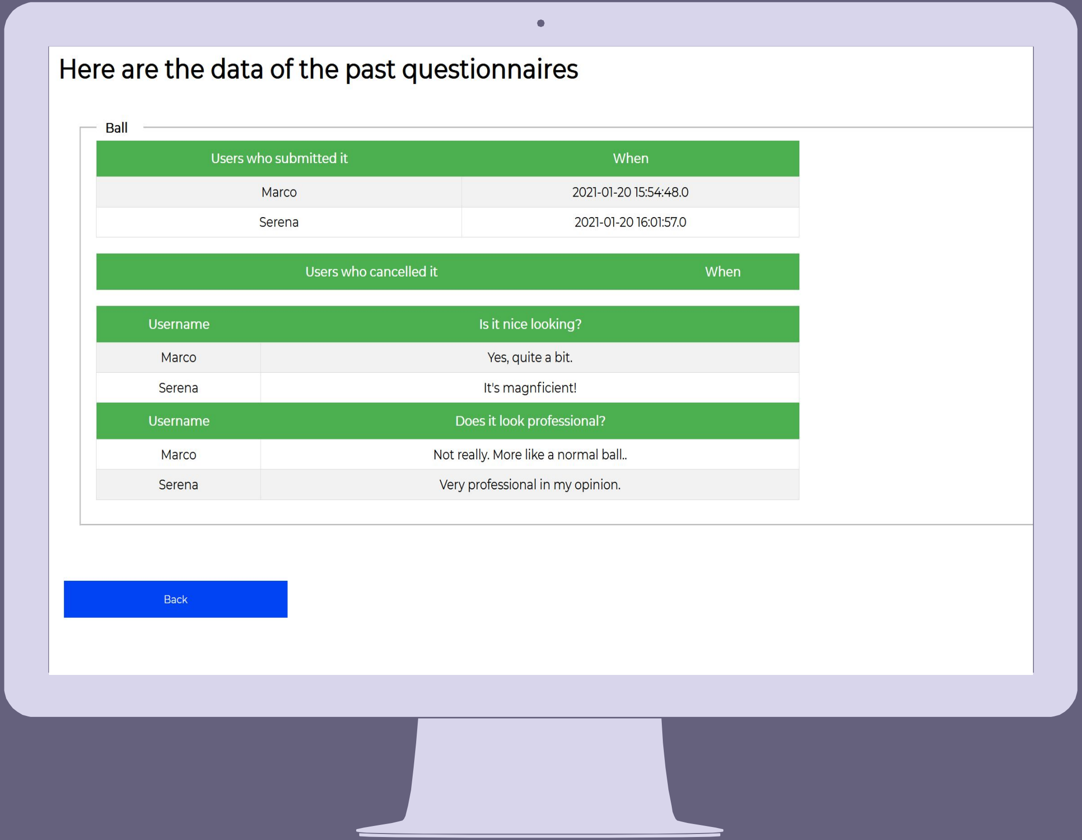
Insert a marketing question:

Your question:

Inspection Page

Page where an admin can inspect data of past questionnaires, such as user who submitted/cancelled it, and the corresponding answers

inspection.html



Here are the data of the past questionnaires

Ball

Users who submitted it	When
Marco	2021-01-20 15:54:48.0
Serena	2021-01-20 16:01:57.0

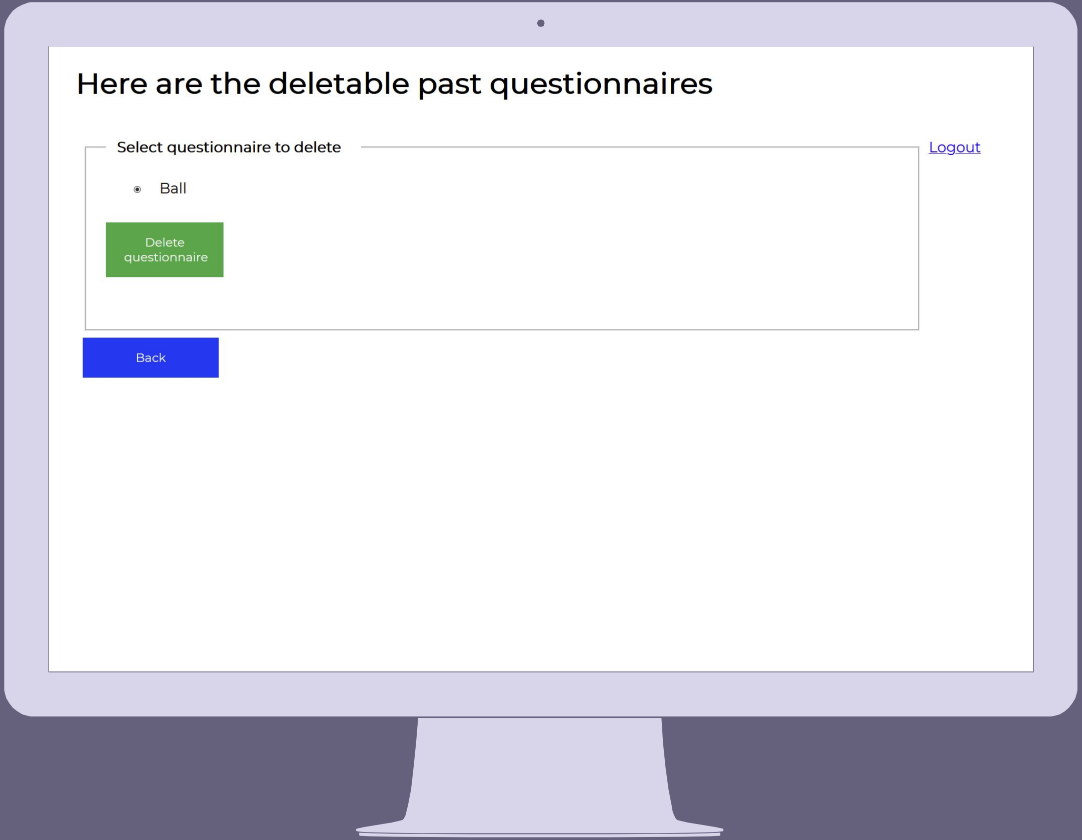
Users who cancelled it	When

Username	Is it nice looking?
Marco	Yes, quite a bit.
Serena	It's magnficient!

Username	Does it look professional?
Marco	Not really. More like a normal ball..
Serena	Very professional in my opinion.

Deletion Page

Page where an admin can delete past questionnaires, and with it the corresponding answers



Here are the deletable past questionnaires

Select questionnaire to delete

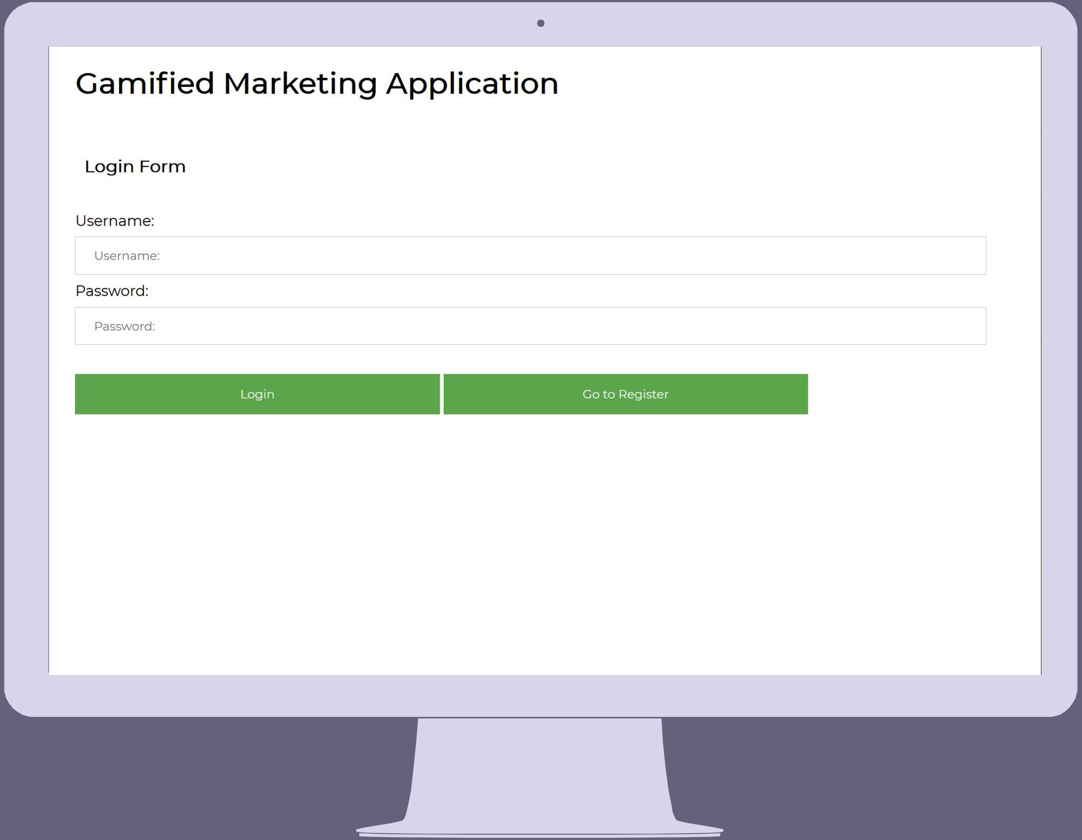
[✓ Ball]

deletion.html

User Sign in

Initial page to perform a login or to access the registration form

index.html


Gamified Marketing Application

Login Form

Username:

Username:

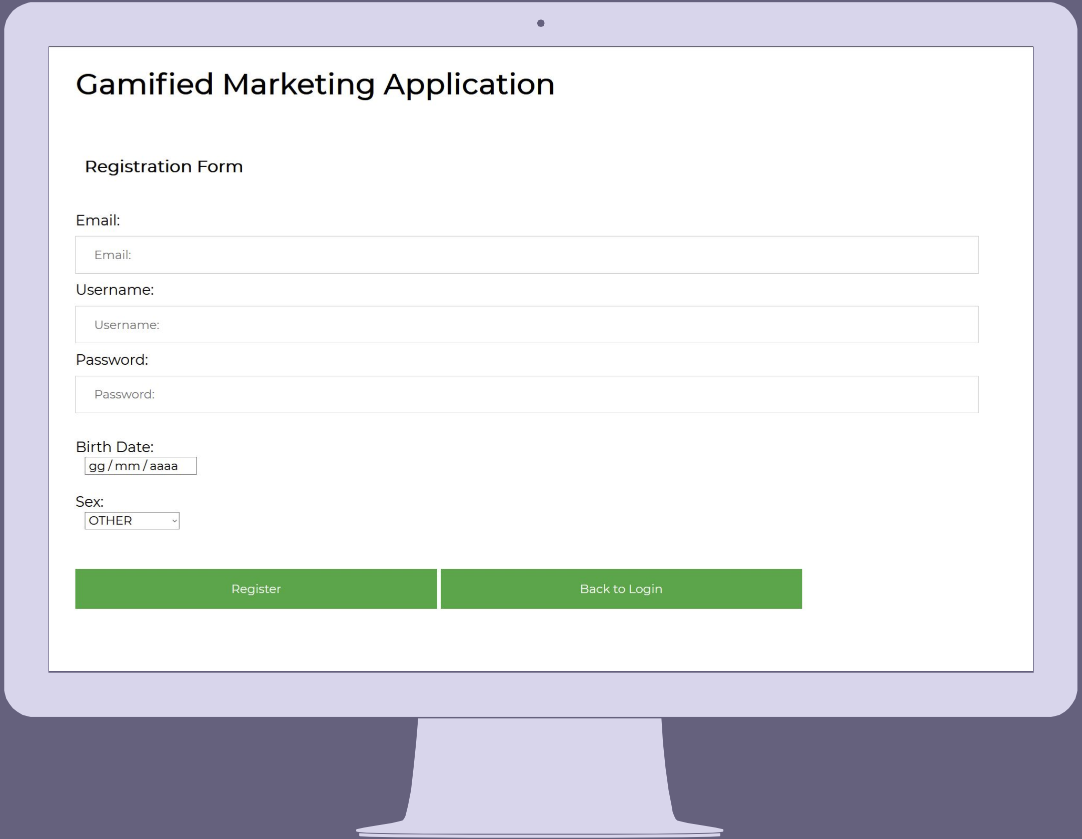
Password:

Password:

User Sign up

Registration form to create a new user account

index.html



Gamified Marketing Application

Registration Form

Email:

Email:

Username:

Username:

Password:

Password:

Birth Date:

gg / mm / aaaa

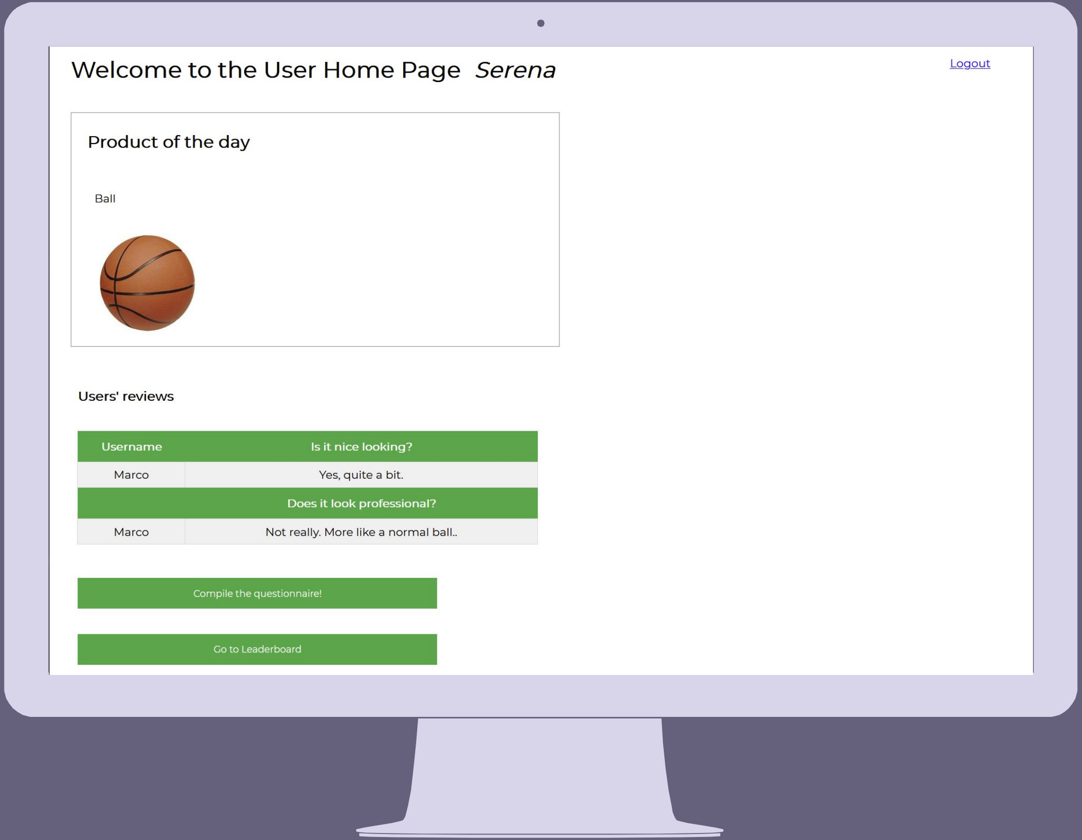
Sex:

OTHER

User home

Home of user where he can see the product of the day and the answers related to all the questionnaire's questions from the users

home_user.html


Welcome to the User Home Page *Serena*

Product of the day

Ball

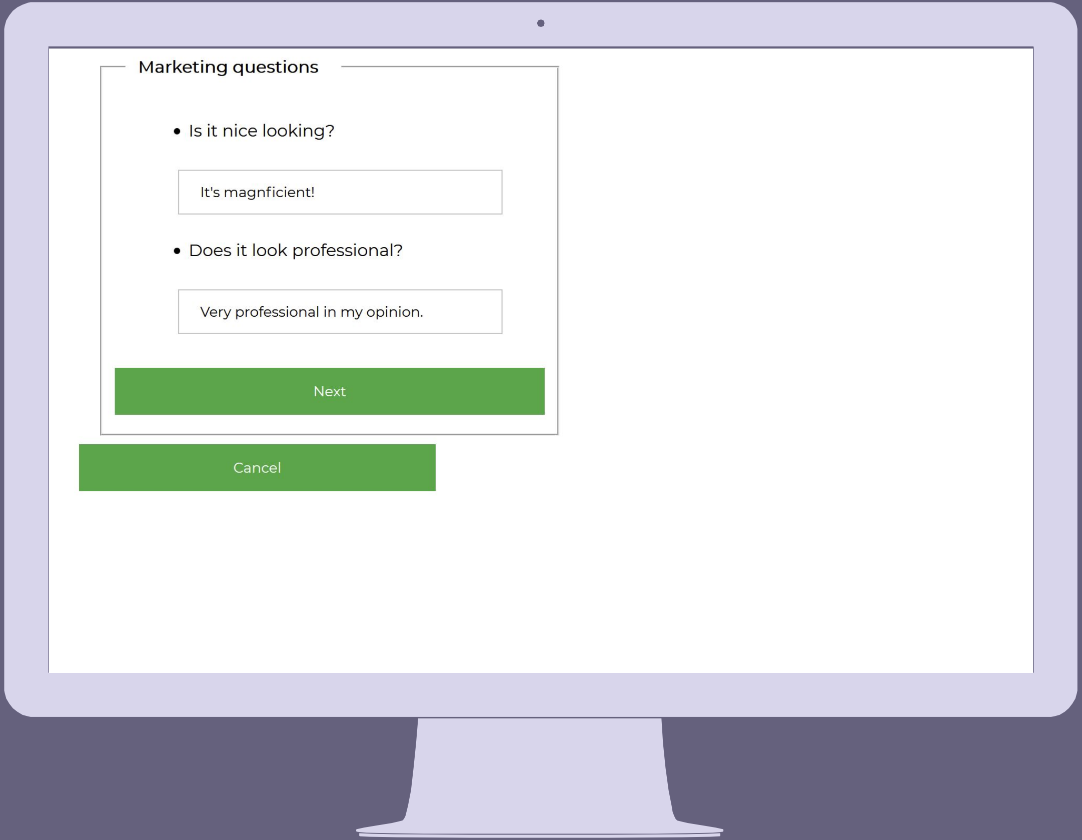
Users' reviews

Username	Is it nice looking?
Marco	Yes, quite a bit.
	Does it look professional?
Marco	Not really. More like a normal ball..

Marketing questions

Page where the user compile the questionnaire by replying to all the mandatory marketing questions

questionnaire.html



Marketing questions

- Is it nice looking?

It's magnficient!

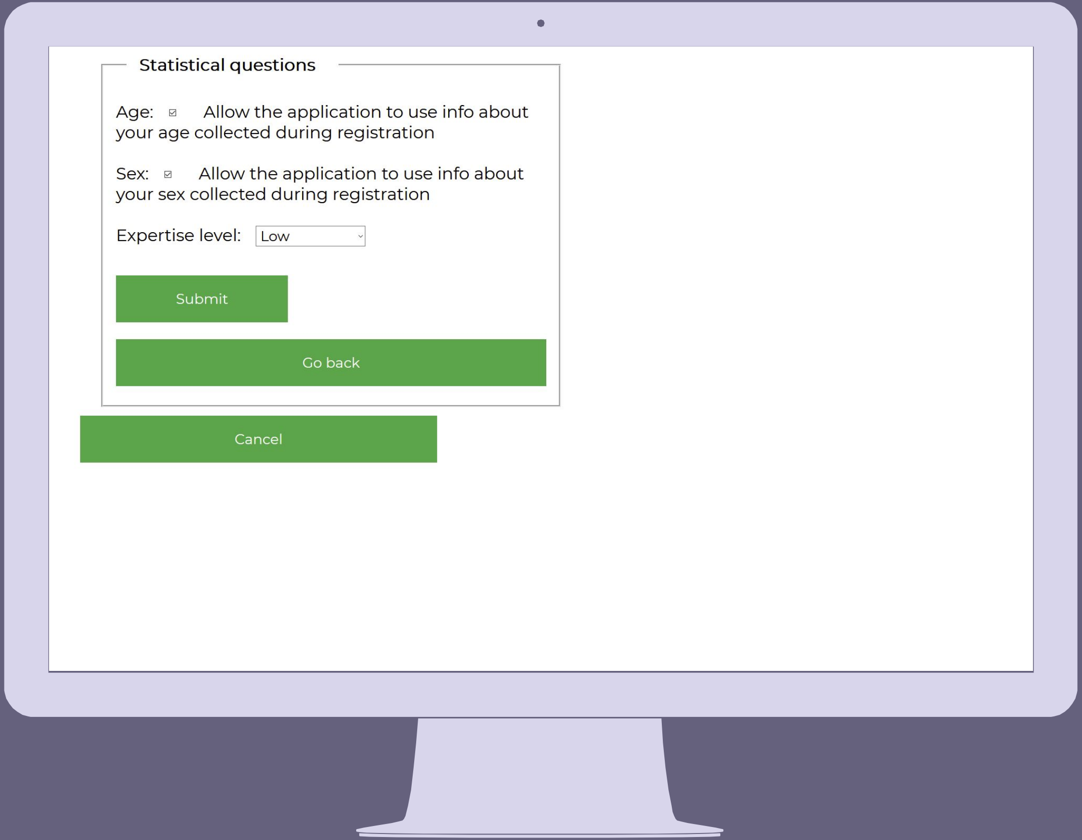
- Does it look professional?

Very professional in my opinion.

Statistical questions

Page where the user compile can choose whether to allow providing her/his statistical data

questionnaire.html



Statistical questions

Age: ☑ Allow the application to use info about your age collected during registration

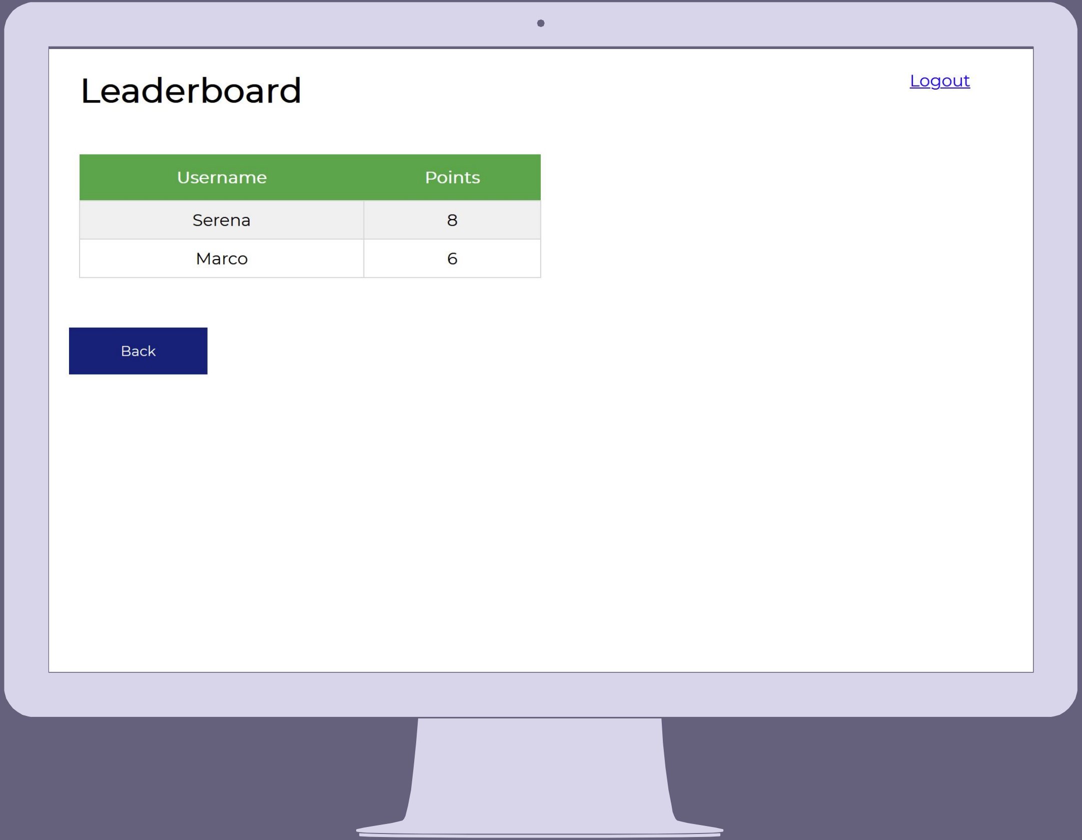
Sex: ☑ Allow the application to use info about your sex collected during registration

Expertise level: Low

Leaderboard Page

Page where the user can take a look at the leaderboard ordered by points, gained by replying to the questionnaire of the day

leaderboard.html



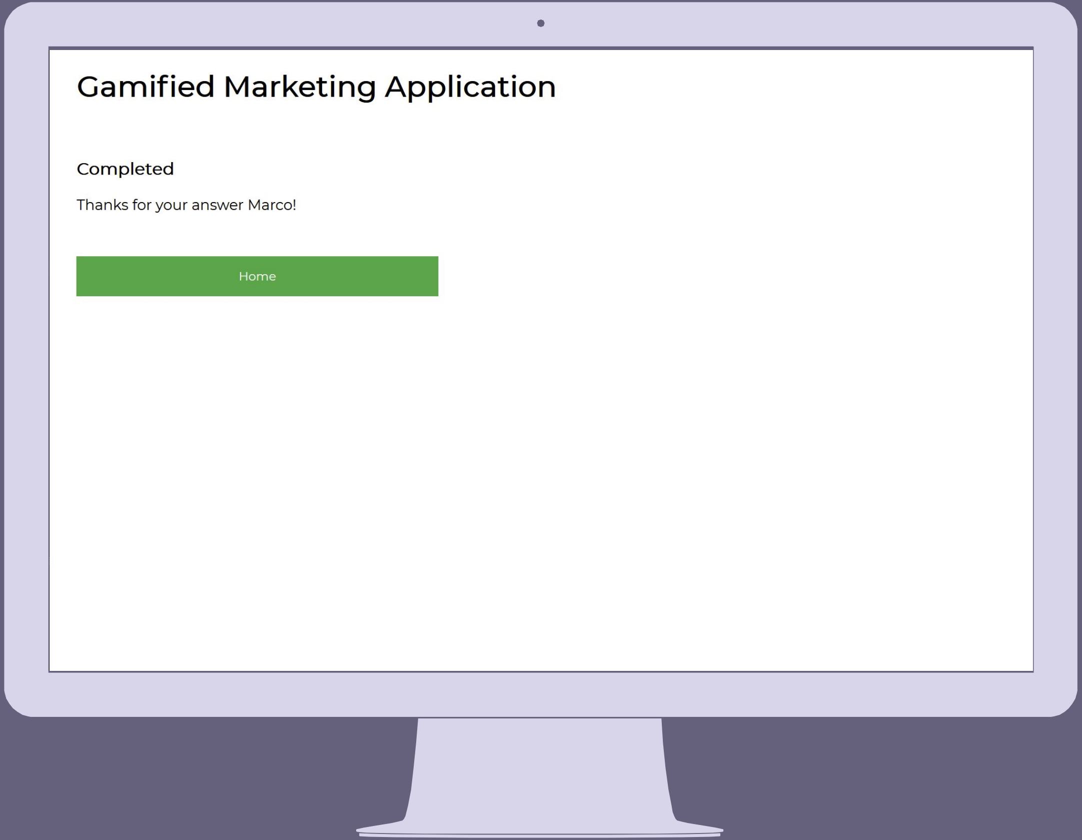
Leaderboard

Username	Points
Serena	8
Marco	6

Greetings Page

Greeting page for thanking the user for the reply

greetings.html

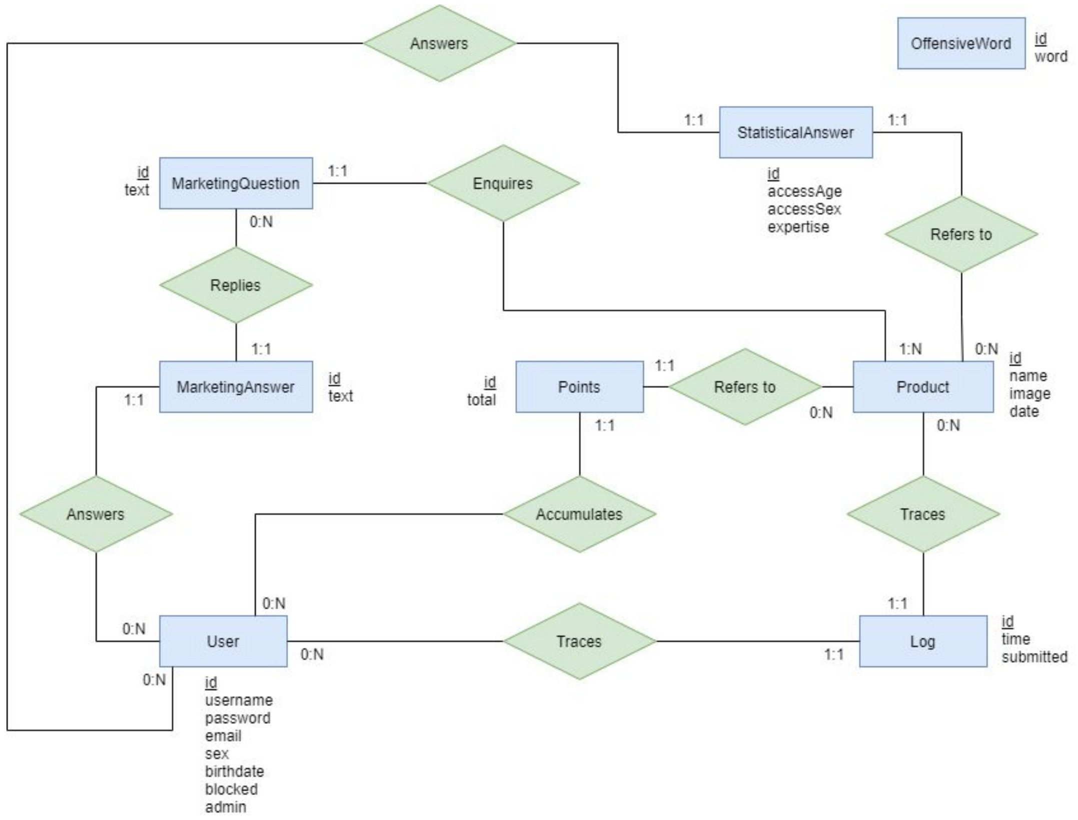


Gamified Marketing Application

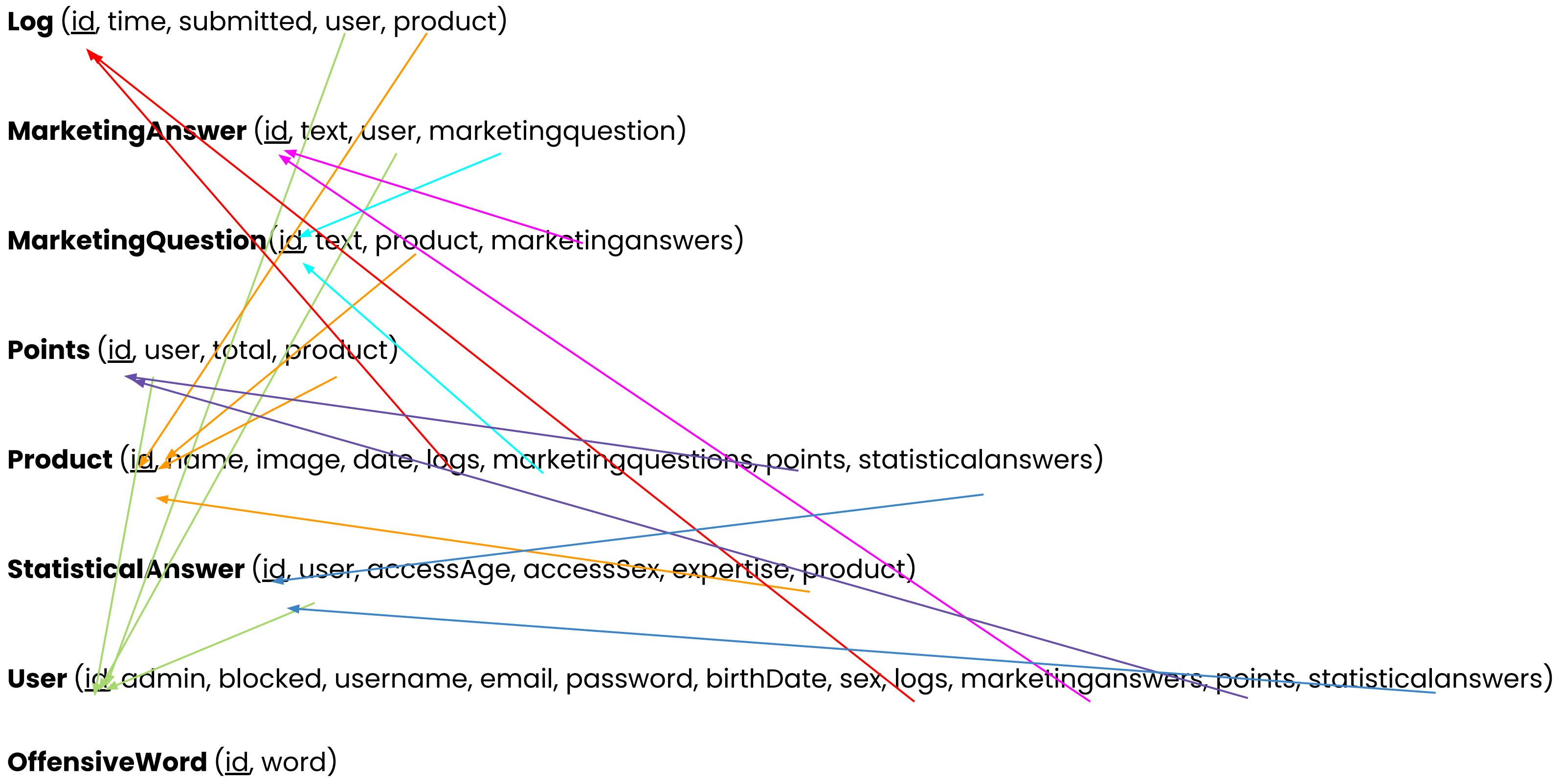
Completed

Thanks for your answer Marco!

ER schema


Answers
OffensiveWord
id
word
1:1
StatisticalAnswer
1:1
id
accessAge
accessSex
expertise
id
text
MarketingQuestion
1:1
Enquires
0:N
Refers to
Replies
1:1
1:N
0:N
id
name
image
date
MarketingAnswer
id
text
id
total
Points
1:1
Refers to
Product
1:1
1:1
0:N
0:N
Answers
Accumulates
Traces
0:N
1:1
0:N
User
0:N
Traces
1:1
Log
id
time
submitted
0:N
id
username
password
email
sex
birthdate
blocked
admin

Logical Model



Log (id, time, submitted, user, product)

MarketingAnswer (id, text, user, marketingquestion)

MarketingQuestion (id, text, product, marketinganswers)

Points (id, user, total, product)

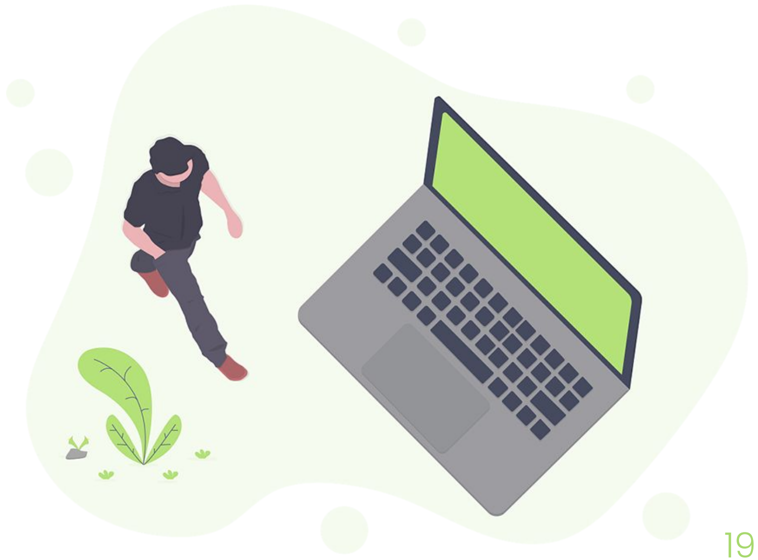
Product (id, name, image, date, logs, marketingquestions, points, statisticalanswers)

StatisticalAnswer (id, user, accessAge, accessSex, expertise, product)

User (id, admin, blocked, username, email, password, birthDate, sex, logs, marketinganswers, points, statisticalanswers)

OffensiveWord (id, word)

Components' Overview



Business Components

Entities

- User
- Points
- Log
- OffensiveWord
- Product
- MarketingQuestion
- MarketingAnswer
- StatisticalAnswer

Services

- UserServiceBean
- PointServiceBean
- LogServiceBean
- OffensiveWordServiceBean
- ProductServiceBean
- MarketingQuestionServiceBean
- MarketingAnswerServiceBean
- StatisticalAnswerServiceBean

Client Components (user)

User web module

- Servlets (controllers)
  - CheckLogin
  - CheckRegistration
  - GoToHomeUser
  - GoToCompileQuestionnaire
  - AnswerQuestionnaire
  - CancelQuestionnaire
  - GoToLeaderboard
  - Logout
- View (pages)
  - blocked.html
  - creation.html
  - duplicate.html
  - greetings.html
  - home_user.html
  - Index.html
  - leaderboard.html
  - questionnaire.html

Client Components (admin)

Admin web module

- Servlets (controllers)
  - CheckLogin
  - GoToHomeAdmin
  - DispatcherAdmin
  - DeleteQuestionnaire
  - InsertProduct
  - Logout

- View (pages)
  - index.html
  - home_admin.html
  - insertion.html
  - inspection.html
  - deletion.html

“ Entities

What follows is a brief overview of the entities we implemented in the EJB module of the project.

For each of them, the attributes are listed.

Entities

User

int	id
byte	admin
byte	blocked
String	username
String	password
String	email
Date	birthdate
String	sex

Product

int	id
String	name
byte	image
Date	date

Entities

StatisticalAnswer

int	<u>id</u>
byte	accessSex
byte	accessAge
String	expertise

MarketingAnswer

int	<u>id</u>
String	text

MarketingQuestion

int	<u>id</u>
String	text

Entities

Point

int	id
Int	total

OffensiveWord

int	id
String	word

Log

int	id
Timestamp	time
byte	submitted

“ Services

What follows is a brief overview of the services we implemented in the EJB module of the project.

For each of them, the most relevant methods are listed. While the name of the method and the return type are specified for each service, the parameters are omitted for brevity.

Services

Product service

```
Product        find(...)
List<Product>  findAll()
Product        findProductOfTheDay(...)
List<Product>  findPastProducts()
Int            createProduct(...)
               updateProduct(...)
               deleteProduct(...)
```

Log service

```
Log        find(...)
List<Log>  findAll()
Boolean    isLogPresent(...)
Int        createLog(...)
           deleteLog(...)
```

Services

MarketingQuestion service

MarketingQuestion	find(...)
List<MarketingQuestion>	findAll( )
Int	createMarketingQuestion(...)

MarketingAnswer service

MarketingAnswer	find(...)
List<MarketingAnswer>	findAll( )
Int	createMarketingAnswer(...)

Services

StatisticalAnswer service

StatisticalAnswer	find(...)
List<StatisticalAnswer>	findAll( )
StatisticalAnswer	createStatisticalAnswer(...)

Points service

Points	find(...)
List<Points>	findAll( )

OffensiveWord service

OffensiveWord	find(...)
List<OffensiveWord>	findAll( )

Services

User service

User	find(...)
List<User>	findAll( )
User	checkCredentials(...)
User	createUser(...)
	deleteUser(...)
	blockUser(...)

“ Servlets

What follows is a brief overview of the servlets we implemented in the Web modules of the project.

For each of them, their purpose is explained briefly.

Servlets

User

CheckLogin

Servlet that processes the login of a user:

If the user exists and is not admin, adds info to the session and goes to home page, otherwise shows login page with an error message

CheckRegistration

Servlet that processes the registration of a new user:

If the user already exists, shows the login page with an error message, otherwise adds info to the session and goes to home page

Servlets

User

GoToHomeUser

Servlet that loads the home page of the user:

gets the user from the session and finds the product of the day with the associated reviews by all the users

GoToCompileQuestionnaire

Servlet that loads the questionnaire compilation page, if he hasn’t already done it

GoToLeaderboard

Servlet that loads the leaderboard page

Servlets

User

CancelQuestionnaire

Servlet that deletes the current answer of a questionnaire:

It creates the log of deletion and redirects to the home page

AnswerQuestionnaire

Servlet that performs the answering of a questionnaire:

It retrieves all the parameters/answers from the form, blocks the user if he used some of the offensive words, otherwise it stores the answers

Logout

Servlet that performs the logout of a user

Servlets

Admin

CheckLogin

Servlet that processes the login of a user:

If the user exists and it is not user, adds info to the session and goes to home page, otherwise shows the login page with an error message. Note that an Admin cannot register, it has to be inserted manually in the database.

GoToHomeAdmin

Servlet that loads the home page of the admin:

It gets the admin from the session, and shows his home page

Servlets

Admin

DispatcherAdmin

Servlet that dispatches the admin to the needed page:

- Insertion page
- Deletions page
- Inspection page

InsertProduct

Servlet that handles the insertion of a new product:

It gets all parameter from the form (mandatory) and, after checking again date validity, insert the new product and the related questions.

Servlets

Admin

DeleteQuestionnaire

Servlet that handles the deletion of a questionnaire (product):

Deletes the product and (in cascade) the related data

Triggers

```
CREATE DEFINER=`root`@`localhost` TRIGGER `marketinganswer_AFTER_INSERT`
AFTER INSERT ON `marketinganswer`
FOR EACH ROW
BEGIN
   DECLARE x INTEGER;

          SELECT productId INTO x                        -- select id of the product
          FROM marketingQuestion
          WHERE NEW.marketingquestionId=id;

          IF EXISTS ( SELECT *                            -- if already exists a row in points from that user and that product, update
                         FROM points p
                         WHERE p.userId=NEW.userId
                          AND p.productId=x)
          THEN
          UPDATE points
                    SET total=total+1
          WHERE userId=NEW.userId
            AND productId=x;

          ELSE                                                 -- else insert a new row
          INSERT INTO points(userId, productid, total)
          VALUES(NEW.userId, x, 1);
                     END IF;
END
```

This trigger computes automatically the points for each user, assigning **1 point** for each **marketing answer**

Triggers

```
CREATE DEFINER=`root`@`localhost` TRIGGER `statisticalanswer_AFTER_INSERT`
AFTER INSERT ON `statisticalanswer`
FOR EACH ROW
BEGIN
  DECLARE pointsToAdd integer;

  SELECT IF(NEW.age=0, 0, 2) + IF(NEW.expertise IS NULL, 0, 2) + IF(NEW.sex=0, 0, 2) INTO pointsToAdd
  FROM points
  WHERE NEW.userId=userId AND NEW.productId=productId;                -- set variable numberOfPointsToAdd checking the new
                                                                         statistical answer

            IF (pointsToAdd>0)
              AND EXISTS (   SELECT *                                  -- if already exists a row in points from that user and that product, update
                             FROM points P
                             WHERE P.userId=NEW.userId
                             AND P.productId=NEW.productId)
              THEN
                UPDATE Points
                  SET total=total+pointsToAdd
                WHERE userId=NEW.userId
                AND productId=NEW.productId;
              ELSE
                INSERT INTO Points(userId, productid, total)            -- else insert a new row
                VALUES(NEW.userId, NEW.productId, pointsToAdd);
END IF;
END
```

This trigger computes automatically the points for each user, assigning **2 points** for each **statistical answer**

END