LARGE SCALE AND MULTI-STRUCTURED DATABASES

TASK 2

***OrgaBet***

*RICCARDO POLINI*

*GERARDO ALVARO*

*MARCO BONGIOVANNI*

*GIULIO SILVESTRI*

**Introduction**

*OrgaBet*, acronym of Organize your Bets, is a web-application that acts as an aggregator for sports betting odds, suggesting to the user which bookmakers offer the best odds for the desired events.

When a visitor lands on the homepage, a brief overview of the application is presented. The visitor may then decide to register (or log-in if already registered) in order to use the functionalities of the application.

Once logged in, the user is presented with a homepage containing the *hot matches* of the day (the most played by the other users). The application will have a side panel in which the sport and competition may be selected by the user in order to display all the playable matches for the current date, with the respective average odds. If the user wishes to bet on a certain result, he/she will select the desired odd and the event will be added to “My Coupon”.

“My Coupon” is a recap of all selected events and will be displayed in a side panel. The list will show the total odds for each available bookmaker so that the user may choose the most convenient one to bet on. The coupon may be saved by the user: if this happens, the list is saved in an archive accessible through the personal profile.

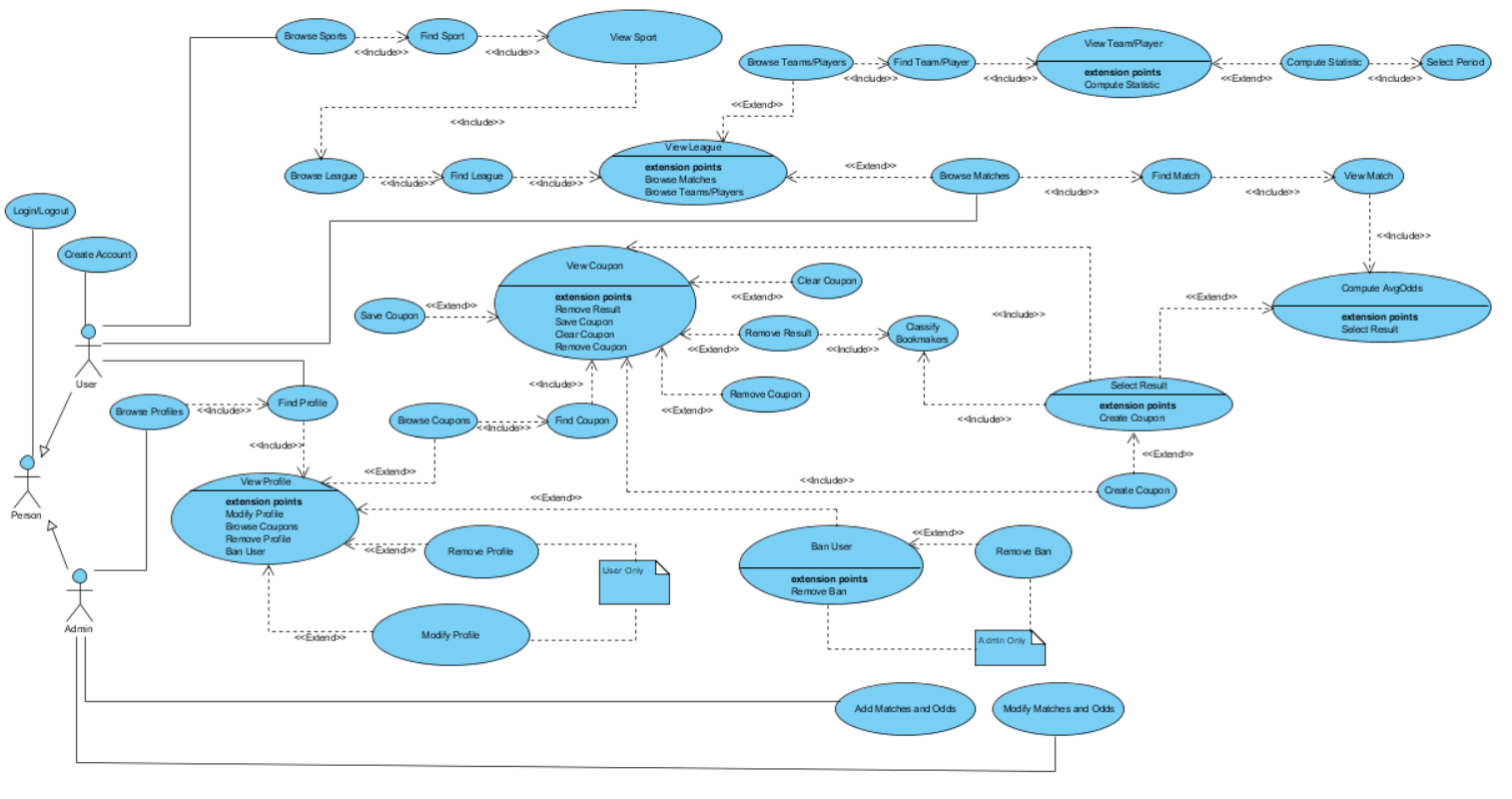
Some sports statistics are also available in a specific panel, accessible to users. Here one can browse various analytics about Matches, Teams or Players.

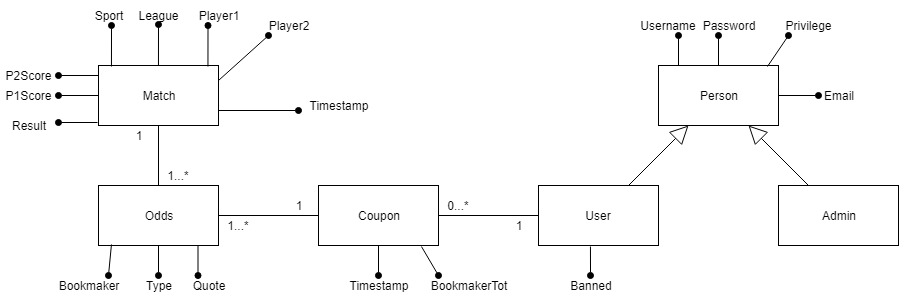
**Functional Requirements**

* The application is available only to registered users.
* The application acts as an aggregator for sports betting odds.
* The application presents the *hot matches* of the day.
* For each sport, the application presents all the events of the day for the available competitions in the available nations.
* For each event, the application displays the average odds of the available bookmakers.
* A User selects a result to bet on and this is added to “My Coupon”.
* A User may remove a selected result from “My Coupon”.
* A User may clear “My Coupon”.
* For each element added to “My Coupon”, the application shows the most convenient bookmakers for betting on the selected events, if any.
* A User may save the “My Coupon” in the personal profile.
* A User may decide to filer certain bookmakers in order to remove their odds for the application.
* A User may delete his/her profile.
* A User can browse the saved coupons in his profile
* A User may edit his/her personal profile.
* A User may remove a previously saved coupon from the personal archive.
* An Admin can view all the traffic of the website.
* An Admin can view the profiles of registered users, including their coupon archive.
* An Admin can add/remove bookmakers from the system.
* An Admin can ban users who violate the Terms & Agreements.

**Non-functional requirements**

* The application’s interface must be user-friendly.
* The application must have a low response time.
* The application will store information in a non-relational Database (MongoDB).
* The application must guarantee data availability.
* The application must be easily scalable.
* The application must be reliable: no system crashes, exceptions are handled etc.
* Admins of the application periodically monitor the behaviour of Users in order to guarantee that they comply to the Terms & Agreements.

****

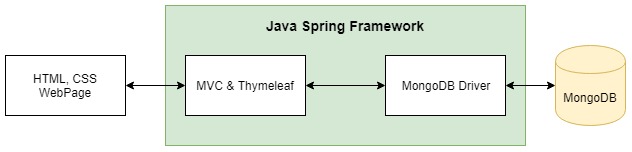
**Class Diagram**

**Data Model**

|  |
| --- |
| **<User>**  {  username:  password:  email:  name:  surname:  privilege:  banned:  coupons: [  {  timestamp:  bookmakerTot: [  {  bookmaker:  quoteTot:  }  ]  bets: [  {  homeTeam:  awayTeam:  result:  avgOdds:  odds: [  {  bookmaker:  odd:  }  ]  }  ]  }  ]  } |

|  |
| --- |
| **<Match>**  {  sport:  division:  date:  time:  homeTeam:  awayTeam:  fullTimeHomeG:  fullTimeAwayG:  fullTimeResult:  odds: [  {  type:  quotes: [  {  bookmaker:  odd:  }  ]  }  ]    //other sport-dependant data in our dataset  } |

**Software Architecture**

****