

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



# BÀI TẬP

**PHÂN TÍCH THIẾT KẾ ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM**

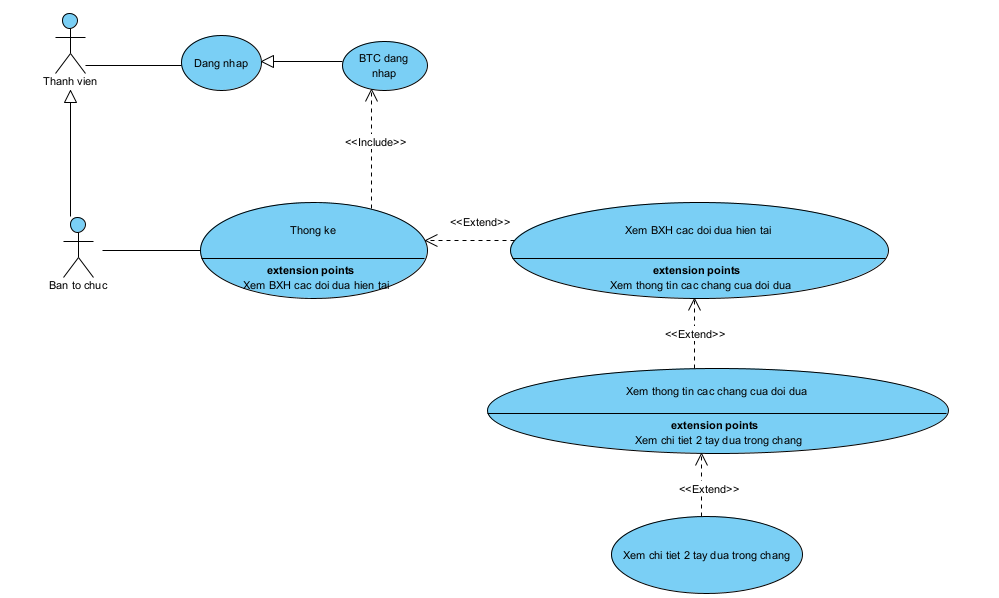
# MODULE: XEM BXH CÁC ĐỘI ĐUA(11.5)

**Họ và tên:** PHẠM CÔNG THÀNH

**Ngày sinh:** 23/10/2000

**Mã sinh viên:** B18DCCN619

1. **Usecase diagram**

****

* Trong chức năng này, ban tổ chức có thể tương tác với các giao diện:
  1. Đăng nhập -> thống nhất với UC đăng nhập
  2. Xem bảng xếp hạng các đội đua hiện tại -> đề xuất UC xem BXH các đội đua hiện tại
  3. Xem thông tin chi tiết đội đua -> đề xuất UC xem thông tin các chặng của đội đua
  4. Xem thông tin chi tiết chặng đua -> đề xuất UC xem chi tiết 2 tay đua trong chặng
  5. Các giao diện thống kê lần lượt theo dạng: giao diện sau là tùy chọn từ giao diện trước. Do đó, chúng có quan hệ mở rộng(Extend) lần lượt cái sau từ cái trước.
* Trong đó, các UC chi tiết được mô tả như sau:
  1. Xem bảng xếp hạng các đội đua hiện tại: UC này cho phép ban tổ chức xem danh sách các đội đua trong mùa giải hiện tại
  2. Xem các chặng đua của một đội: UC này cho phép ban tổ chức xem thông tin các chặng đua của một đội đua trong mùa giải
  3. Xem chi tiết 2 tay đua: UC này cho phép ban tổ chức xem kết quả chi tiết 2 tay đua trong một chặng đua

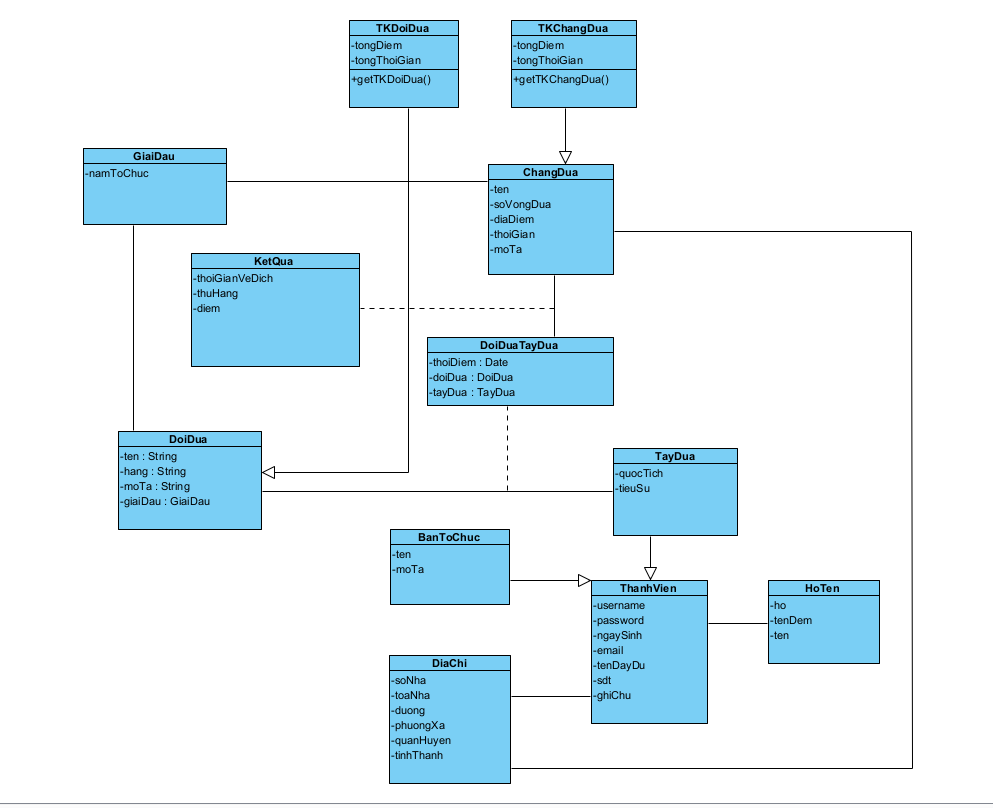
1. **Viết kịch bản**

**a. Kịch bản cho modul xem bảng xếp hạng các đội đua**

|  |  |
| --- | --- |
| Use case | Xem BXH các đội đua |
| Actor | Ban tổ chức |
| Tiền điều kiện | Ban tổ chức đã đăng nhập thành công, đã có đầy đủ thông tin giải đấu, đội đua, chặng đua và tay đua. |
| Hậu điều kiện | Ban tổ chức hoàn thành việc xem bảng xếp hạng các đội đua hiện tại |
| Kịch bản chính | 1. Sau khi đăng nhập, 2. Giao diện chọn thống kê hiện lên. Có danh sách các loại thống kê: xem bảng xếp hạng tay đua, xem bảng xếp hạng đội đua. 3. Ban tổ chức chọn chức năng xem bảng xếp hạng các đội đua, bảng xếp hạng các đội đua năm 2022 4. Giao diện bảng xếp hạng các đội đua năm 2022 hiện lên:  |  |  |  |  |  | | --- | --- | --- | --- | --- | | TT | Tên đội đua | Hãng | Tổng điểm | Tổng thời gian(phút) | | 1 | Honda-hd1 | Honda | 100 | 600 | | 2 | Mazda-md2 | Mazda | 90 | 650 | | 3 | Toyota-tyt3 | Toyota | 80 | 700 |  1. Ban tổ chức click vào dòng Honda-hd1 2. Giao diện các chặng đua của đội đua Honda-hd1 được hiện lên:  |  |  |  |  | | --- | --- | --- | --- | | TT | Tên chặng | Tổng điểm | Tổng thời gian(phút) | | 1 | Hà Giang – Hà Tĩnh | 45 | 100 | | 2 | Hà Tĩnh – Đà Nẵng | 35 | 200 | | 3 | Đà Nẵng – Cà Mau | 20 | 300 |  1. Ban tổ chức click vào dòng Hà Giang – Hà Tĩnh 2. Giao diện chi tiết 2 tay đua tham gia trong chặng đua Hà Giang – Hà Tĩnh hiện lên  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | TT | Tên tay đua | Tên chặng | Thời gian về đích | Thứ hạng | Điểm | | 1 | Phạm Công Thành | Hà Giang – Hà Tĩnh | 40 | 1 | 25 | | 2 | Phạm Công Bình | Hà Giang – Hà Tĩnh | 60 | 2 | 20 | |
| Ngoại lệ | 4. Năm 2022 không có đội đua nào thi đấu |

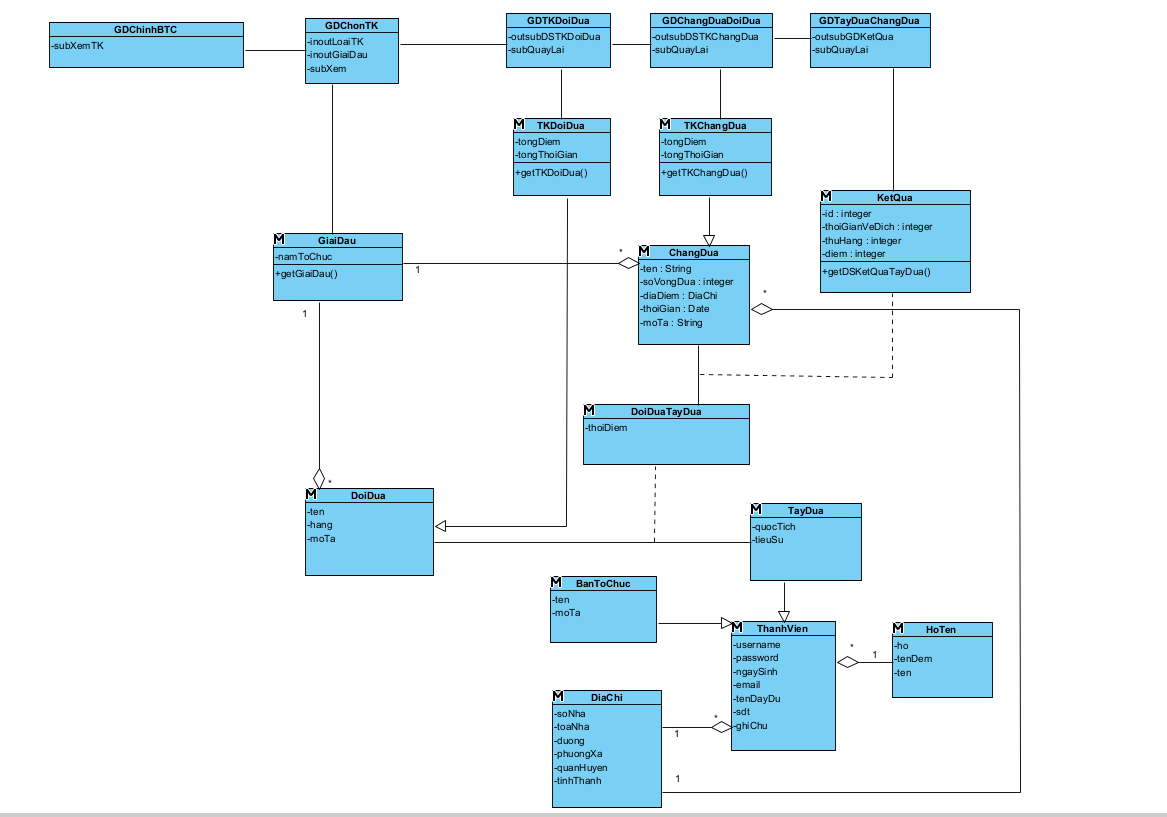
1. **Trích lớp thực thể và biểu đồ lớp thực thể liên quan đến modul xem bảng xếp hạng các đội đua**

* **Bước 1: Mô tả hệ thống trong một đoạn văn**
  1. Module xem bảng xếp hạng các đội đua cho phép ban tổ chức có thể xem thống kê bảng xếp hạng các đội đua của giải đấu hiện tại, xem thống kê các chặng đua trong một đội đua và xem chi tiết 2 tay đua trong từng chặng đua.
* **Bước 2: Trích các danh từ xuất hiện trong đoạn văn**
  1. Các danh từ liên quan đến người: ban tổ chức, tay đua, đội đua.
  2. Các danh từ liên quan đến vật: giải đấu, chặng đua.
  3. Các danh từ liên quan đến thông tin: thống kê bảng xếp hạng các đội đua, thống kê các chặng đua, xem chi tiết 2 tay đua trong từng chặng đua
* **Bước 3: Đánh giá và lựa chọn các danh từ làm lớp thực thể hoặc thuộc tính**
  1. Các danh từ trừu tượng: Module xem bảng xếp hạng, xem thống kê -> loại.
  2. Các danh từ liên quan đến người:
     + Thành viên hệ thống -> ThanhVien (trừu tượng): tên, tên đăng nhập, mật khẩu, ngày sinh, địa chỉ, email, điện thoại, ghi chú.
     + Ban tổ chức -> lớp BanToChuc kế thừa lớp Thành viên, thêm thuộc tính tên, mô tả.
     + Tay đua -> lớp TayDua kế thừa lớp Thành viên, thêm thuộc tính quốc tịch, tiểu sử.
  3. Các danh từ liên quan đến vật:
     + Giải đấu -> lớp GiaiDau: năm tổ chức.
     + Chặng đua -> lớp ChangDua: tên, số vòng đua, địa điểm, thời gian, mô tả.
  4. Các danh từ liên quan đến thông tin:
     + Đội đua -> lớp DoiDua: tên, hãng, mô tả.
     + Thống kê bảng xếp hạng các đội đua -> lớp TKDoiDua
     + Thống kê các chặng đua -> lớp TKChangDua
* **Bước 4: Xác định mối quan hệ số lượng giữa các thực thể**
  1. Giải đấu có nhiều chặng đua -> quan hệ giữa GiaiDau và ChangDua là 1-n
  2. Giải đấu có nhiều đội đua -> quan hệ giữa GiaiDau và DoiDua là 1-n
  3. Mỗi đội đua có nhiều tay đua, cùng một mùa giải, một tay đua có thể thi đấu cho nhiều đội đua ở nhiều thời điểm khác nhau -> DoiDua – TayDua là n-n -> đề xuất lớp DoiDuaTayDua xác định duy nhất một tay đua thuộc một đội đua nhất định.
  4. Mỗi DoiDuaTayDua có nhiều ChangDua, 1 ChangDua có nhiều DoiDuaTayDua -> đề xuất lớp KetQua xác định duy nhất kết quả của tay đua đã tham gia một chặng đua xác định
  5. Một địa chỉ có thể có nhiều thành viên, một thành viên có 1 địa chỉ -> DiaChi – ThanhVien là 1-n
  6. Một họ tên có thể có nhiều người chung, 1 người chỉ có 1 họ tên -> HoTen-ThanhVien là 1-n
  7. Ngoài ra, thông tin Họ và tên có thể tách thành một lớp thực thể riêng.
  8. Thông tin địa chỉ cũng có thể tách thành một lớp thực thể riêng.
* **Bước 5: Xác định quan hệ đối tượng giữa các thực thể**
  1. Chặng đua là thành phần của giải đấu
  2. Đội đua là thành phần của giải đấu
  3. Đội đua và tay đua tạo ra một DoiDuaTayDua duy nhất
  4. Chặng đua và DoiDuaTayDua tạo ra một kết quả duy nhất cho tay đua đối với một đội đua
  5. Tay đua là thành phần của chặng đua
  6. Thông tin địa chỉ nằm trong thông tin về thành viên, chặng đua.
  7. Thông tin họ và tên nằm trong thông tin thành viên.
* Kết quả thu được biểu đồ lớp thực thể pha phân tích module xem BXH đội đua:



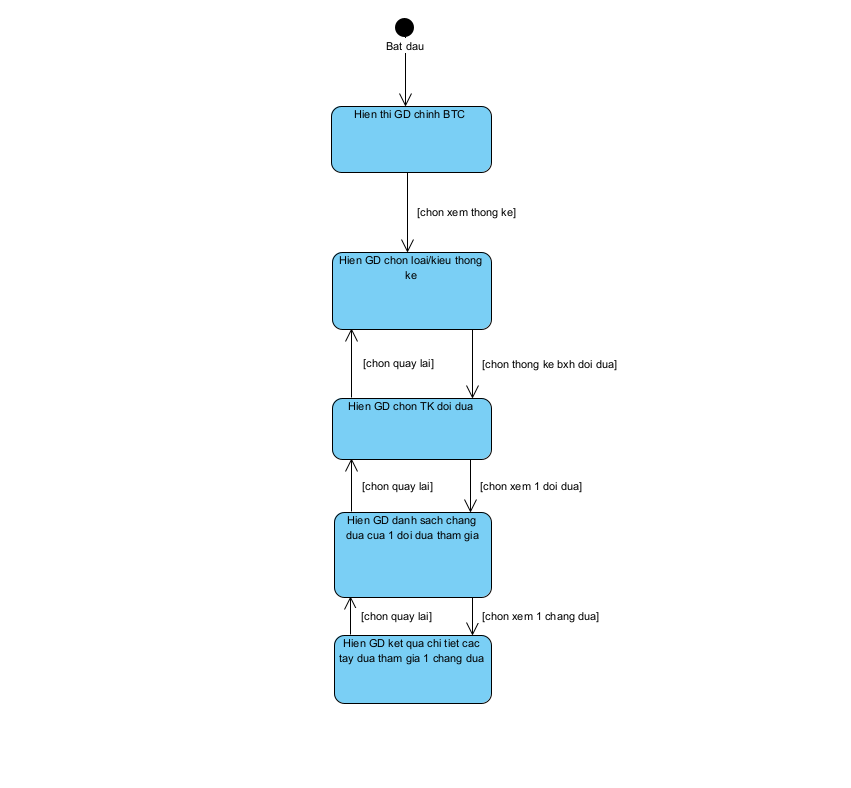
1. **Phân tích tĩnh cho modul xem BXH các đội đua**

* Ban đầu, từ giao diện chính của ban tổ chức -> đề xuất lớp GDChinhBTC, cần có:
  1. Chọn xem thống kê: kiểu submit
* Bước 2: Giao diện chọn loại thống kê và năm -> đề xuất lớp GDChonTK, cần có:
  1. Danh sách loại thống kê: vừa input, vừa output
  2. Danh sách giải đấu theo các năm: vừa input, vừa output
  3. Nút xem: kiểu submit
* Để có danh sách giải đấu theo các năm, cần xử lí trước dưới hệ thống:
  1. Lấy danh sách giải đấu theo các năm
  2. Input: Không có
  3. Output: Tất cả giải đấu của các năm trước cho đến hiện tại
  4. Đề xuất phương thức getGiaiDau(), gán cho lớp GiaiDau
* Bước 4, giao diện thống kê bảng xếp hạng các đội đua -> đề xuất lớp GDTKDoiDua, cần có:
  1. Danh sách thống kê các đội đua: output, submit
* Để có dữ liệu thống kê, cần xử lý dưới hệ thống:
  1. Lấy dữ liệu các đội đua từ giải đấu đã chọn
  2. Input: giải đấu
  3. Output: danh sách TKDoiDua
  4. Đề xuất phương thức getTKDoiDua(), gán cho lớp TKDoiDua
* Bước 6, giao diện các chặng đua mà đội đua tham gia -> GDChangDuaDoiDua, cần có
  1. Danh sách các chặng đua mà đội đua đó tham gia + điểm và thời gian: kiểu output, submit.
* Để có danh sách các điểm, thời gian của chặng đua mà đội đua đó tham gia, cần xử lý:
  1. Lấy danh sách các chặng đua + kết quả của đội đua đó( tổng điểm, tổng thời gian)
  2. Input: Đội đua, giải đấu
  3. Output: danh sách TKChangDua
  4. Đề xuất phương thức getTKChangDua(), gán cho lớp TKChangDua
* Bước 8, giao diện danh sách các tay đua của chặng đua -> Lớp GDTayDuaChangDua cần:
  1. Danh sách các tay đua trong chặng đua: output, submit.
* Để có danh sách các tay đua tham gia chặng đua đó, cần xử lý:
  1. Lấy danh sách điểm,thứ hạng, thời gian về đích các tay đua tham gia chặng đua đó
  2. Input: Tay đua, chặng đua
  3. Output: danh sách kết quả các tay đua tham gia chặng đua đó(KetQua)
  4. Đề xuất phương thức getDSKetQuaTayDua(), gán cho lớp KetQua
* Kết quả thu được biểu đồ lớp phân tích modul xem bảng xếp hạng đội đua



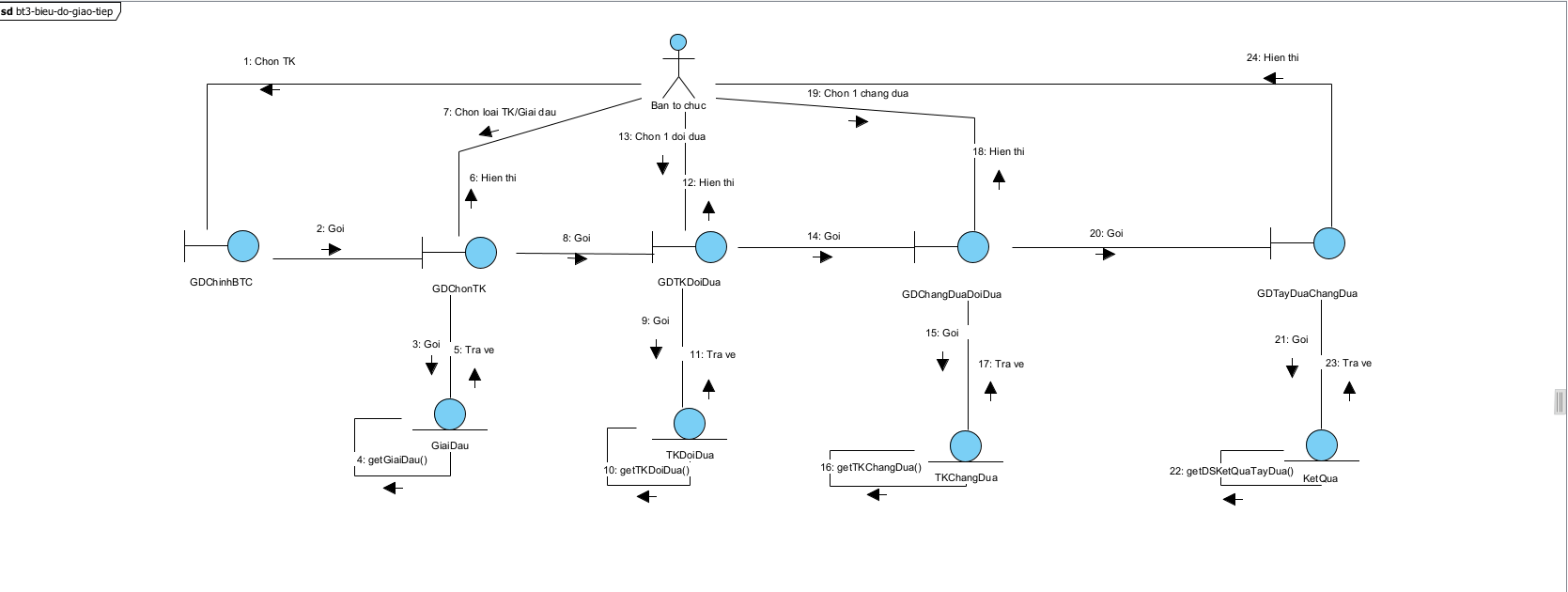
1. **Biểu đồ trạng thái module xem BXH các đội đua**

* Từ giao diện chính của ban tổ chức, khi chọn chức năng xem thống kê(xem bảng xếp hạng), hệ thống sẽ chuyển sang giao diện chọn kiểu thống kê
* Tại giao diện chọn kiểu thống kê, chọn loại thống kê: xem bảng xếp hạng các đội đua, hệ thống chuyển sang giao diện thống kê đội đua(bảng xếp hạng đội đua)
* Tại giao diện thống kê đội đua, khi chọn vào 1 đội đua, hệ thống chuyển sang giao diện thống kê các chặng đua mà đội đua đó tham gia.
* Tại giao diện thống kê chặng đua, chọn một chặng đua, hệ thống chuyển sang giao diện thông tin kết quả chi tiết của từng tay đua đã tham gia chặng đua đó
* Tại tất cả các giao diện, nếu chọn quay lại, hệ thống sẽ quay lại giao diện trước đó.
* Kết quả thu được biểu đồ trạng thái modul xem bảng xếp hạng đội đua

****

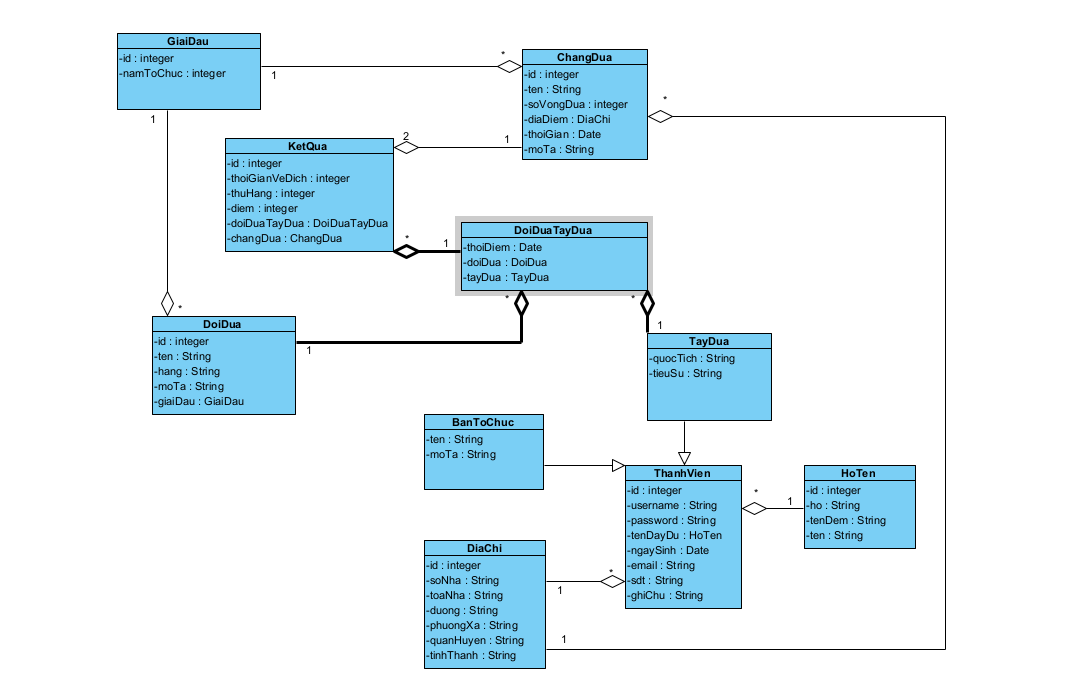
1. **Biểu đồ giao tiếp module xem BXH các đội đua**

* Kịch bản v.2 của modul diễn ra như sau:
  1. Tại giao diện chính của ban tổ chức, sau khi đăng nhập, ban tổ chức chọn xem thống kê
  2. Lớp GDChinhBTC gọi lớp GDChonTK
  3. Lớp GDChonTK gọi lớp GiaiDau yêu cầu lấy ra danh sách các giải đấu
  4. Lớp GiaiDau tìm danh sách các giải đấu
  5. Lớp GiaiDau trả kết quả lại cho lớp GDChonTK
  6. Lớp GDChonTK hiển thị cho BTC
  7. BTC chọn thống kê đội đua(xem BXH đội đua)
  8. Lớp GDChonTK gọi lớp GDTKDoiDua
  9. Lớp GDTKDoiDua hiển thị cho BTC
  10. BTC chọn 1 giải đấu muốn thống kê
  11. Lớp GDTKDoiDua gọi lớp TKDoiDua yêu cầu lấy dữ liệu
  12. Lớp TKDoiDua lấy dữ liệu thống kê
  13. Lớp TKDoiDua trả kết quả cho lớp GDTKDoiDua
  14. Lớp GDTKDoiDua hiển thị cho BTC
  15. BTC Chọn 1 đội đua
  16. Lớp GDTKDoiDua gọi lớp GDChangDuaDoiDua
  17. Lớp GDChangDuaDoiDua gọi lớp TKChangDua yêu cầu lấy dữ liệu
  18. Lớp TKChangDua lấy dữ liệu thống kê chặng đua mà đội đua tham gia
  19. Lớp TKChangDua trả lại kết quả cho lớp GDChangDuaDoiDua
  20. Lớp GDChangDuaDoiDua hiển thị cho BTC
  21. BTC click một chặng đua
  22. Lớp GDChangDuaDoiDua gọi lớp GDTayDuaChangDua
  23. Lớp GDTayDuaChangDua gọi lớp KetQua yêu cầu lấy dữ liệu
  24. Lớp KetQua lấy dữ liệu thống kê kết quả các tay đua tham gia chặng đua đó
  25. Lớp KetQua trả kết quả cho lớp GDTayDuaChangDua
  26. Lớp GDTayDuaChangDua hiển thị cho BTC



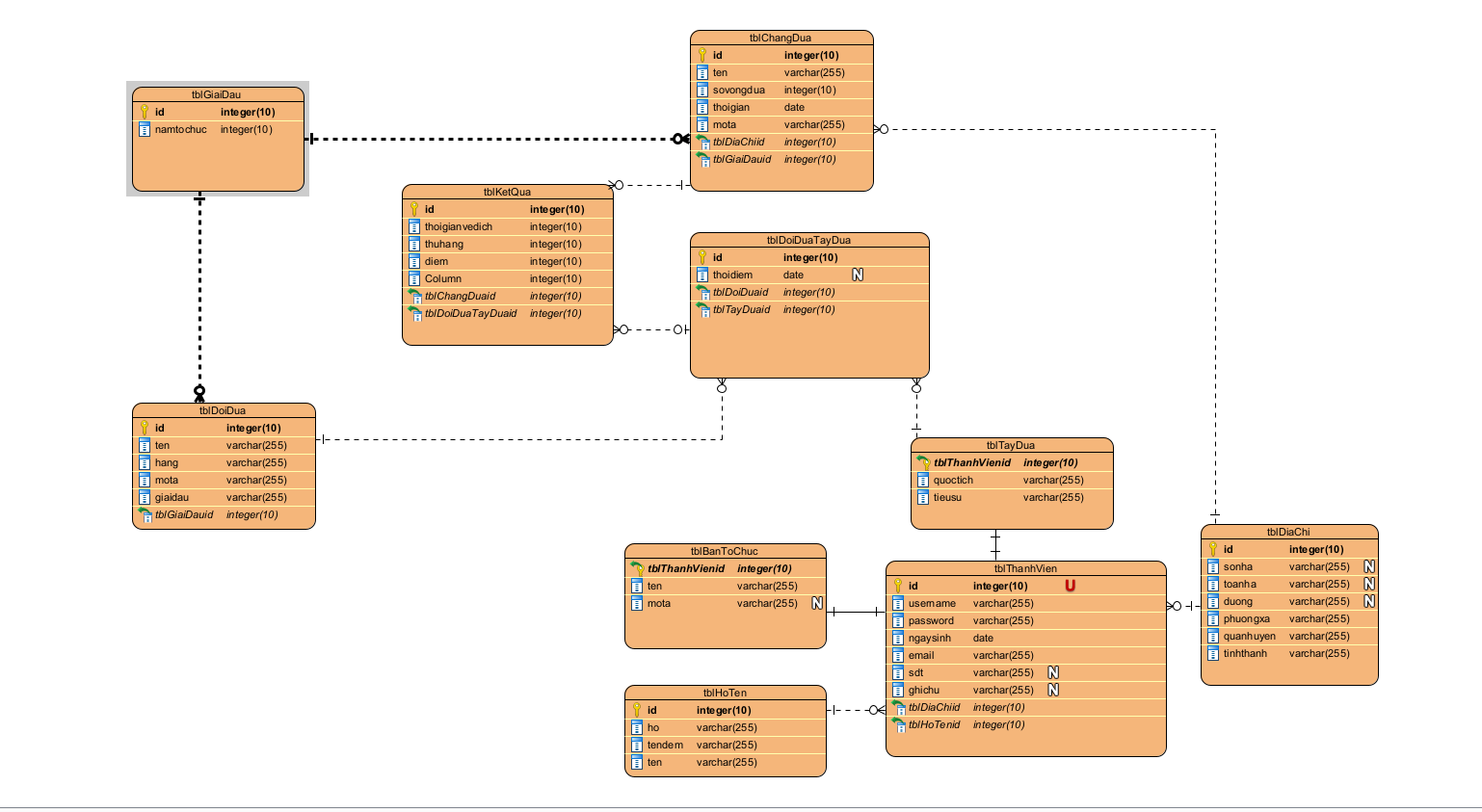
1. **Thiết kế lớp thực thể**

* Bước 1: Các lớp được bổ sung thuộc tính id, trừ các lớp:
  + TK
  + TayDua, BanToChuc(do kế thừa lớp ThanhVien)
* Bước 2: Thuộc tính của các lớp được bổ sung thuộc tính theo kiểu của ngôn ngữ lập trình Java.
* Bước 3:
  + Quan hệ DoiDua – TayDua -> DoiDuaTayDua chuyển thành DoiDuaTayDua chứa DoiDua và TayDua
  + Quan hệ DoiDuaTayDua - ChangDua -> KetQua chuyển thành KetQua chứa DoiDuaTayDua và ChangDua.
* Kết quả thu được biểu đồ thiết kế lớp thực thể:

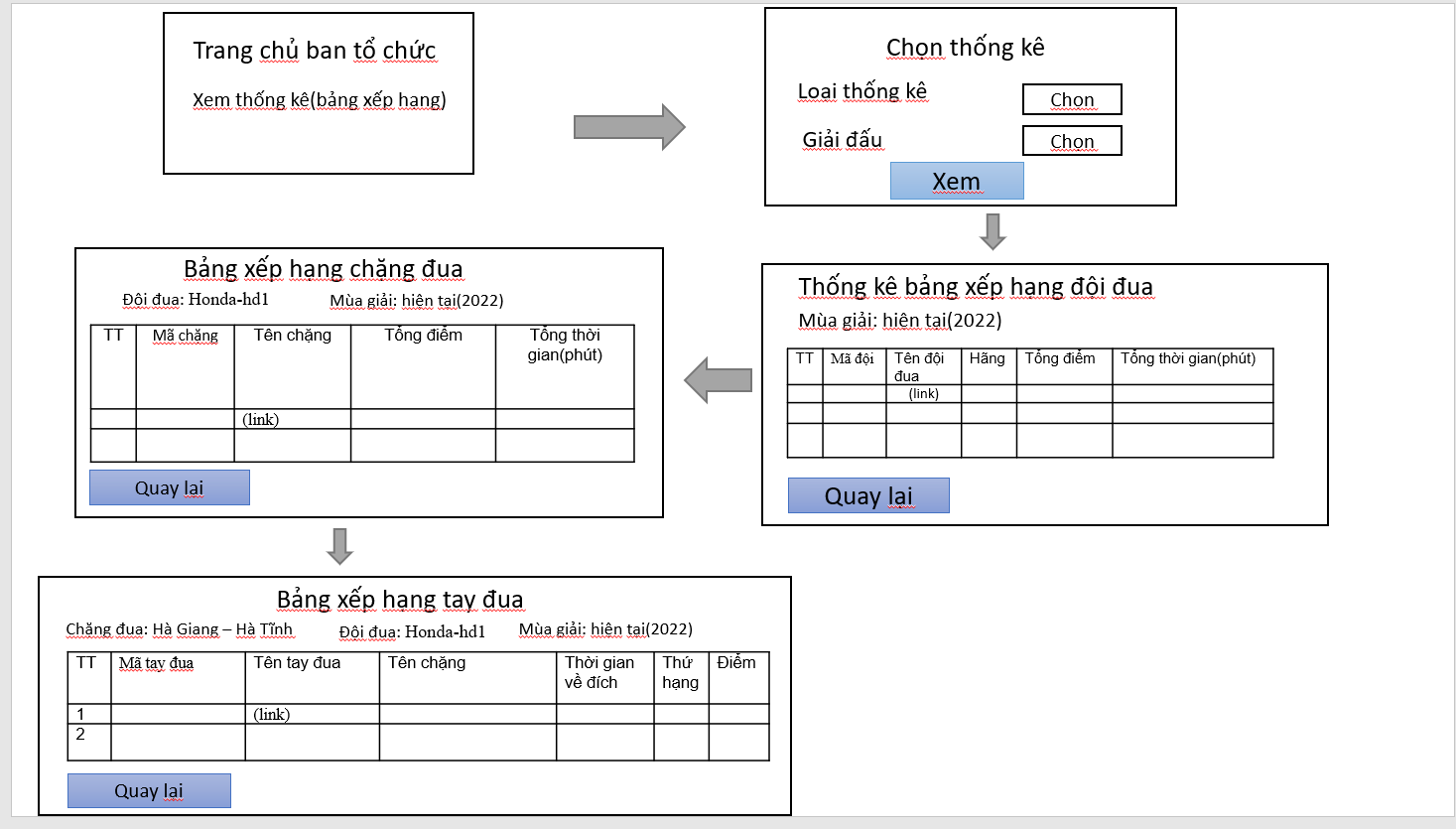


1. **Thiết kế cơ sở dữ liệu**

* Bước 1: Mỗi lớp thực thể đề xuất bảng tương ứng:
  + Lớp GiaiDau -> bảng tblGiaiDau
  + Lớp DoiDua -> bảng tblDoiDua
  + Lớp ChangDua -> bảng tblChangDua
  + Lớp TayDua -> bảng tblTayDua
  + Lớp BanToChuc -> bảng tblBanToChuc
  + Lớp DiaChi -> bảng tblDiaChi
  + Lớp HoTen -> bảng tblHoTen
  + Lớp ThanhVien-> bảng tblThanhVien
  + Lớp DoiDuaTayDua -> bảng tblDoiDuaTayDua
  + Lớp KetQua -> bảng tblKetQua
* Bước 2: Đưa thuộc tính không phải đối tượng của lớp thực thể thành thuộc tính của bảng tương ứng:
* Bước 3: Chuyến quan hệ số lượng giữa các lớp thực thể thành quan hệ số lượng giữa các bảng:
  + 1 tblGiaiDau – n tblDoiDua
  + 1 tblGiaiDau – n tblChangDua
  + 1 tblDoiDua – n tblDoiDuaTayDua
  + 1 tblTayDua– n tblDoiDuaTayDua
  + 1 tblDoiDuaTayDua – n tblKetQua
  + 1 tblChangDua – n tblKetQua
  + 1 tblDiaChi– n tblThanhVien
  + 1 tblDiaChi – n tblChangDua
  + 1 tblHoTen– n tblThanhVien
* Bước 4: Bổ sung các thuộc tính khóa:
  + Khóa chính được thiết lập với thuộc tính id của các bảng tương ứng: trừ các bảng tblTK, tblBanToChuc, tblTayDua.
  + Khóa ngoại được thiết lập cho các bảng:
    - 1 tblGiaiDau – n tblDoiDua -> bảng tblDoiDua có khóa ngoại tblGiaidauid
    - 1 tblGiaiDau – n tblChangDua -> bảng tblChangDua có khóa ngoại tblGiaidauid
    - 1 tblDoiDua – n tblDoiDuaTayDua -> bảng tblDoiDuaTayDua có khóa ngoại tblDoiduaid
    - 1 tblTayDua– n tblDoiDuaTayDua -> bảng tblDoiDuaTayDua có khóa ngoại tblTayduaid
    - 1 tblDoiDuaTayDua – n tblKetQua -> bảng tblKetQua có khóa ngoại tblDoiduatayduaid
    - 1 tblChangDua – n tblKetQua -> bảng tblKetQua có khóa ngoại tblchangduaid
    - 1 tblDiaChi– n tblThanhVien -> bảng tblThanhVien có khóa ngoại tbldiachiid
    - 1 tblDiaChi – n tblChangDua -> bảng tblChangDua có khóa ngoại tbldiachiid
    - 1 tblHoTen– n tblThanhVien -> bảng tblThanhVien có khóa ngoại tblhotenid
* Bước 5: Các thuộc tính dẫn xuất:
  + Các thuộc tính của các lớp thống kê -> loại bỏ hết các bảng thống kê.

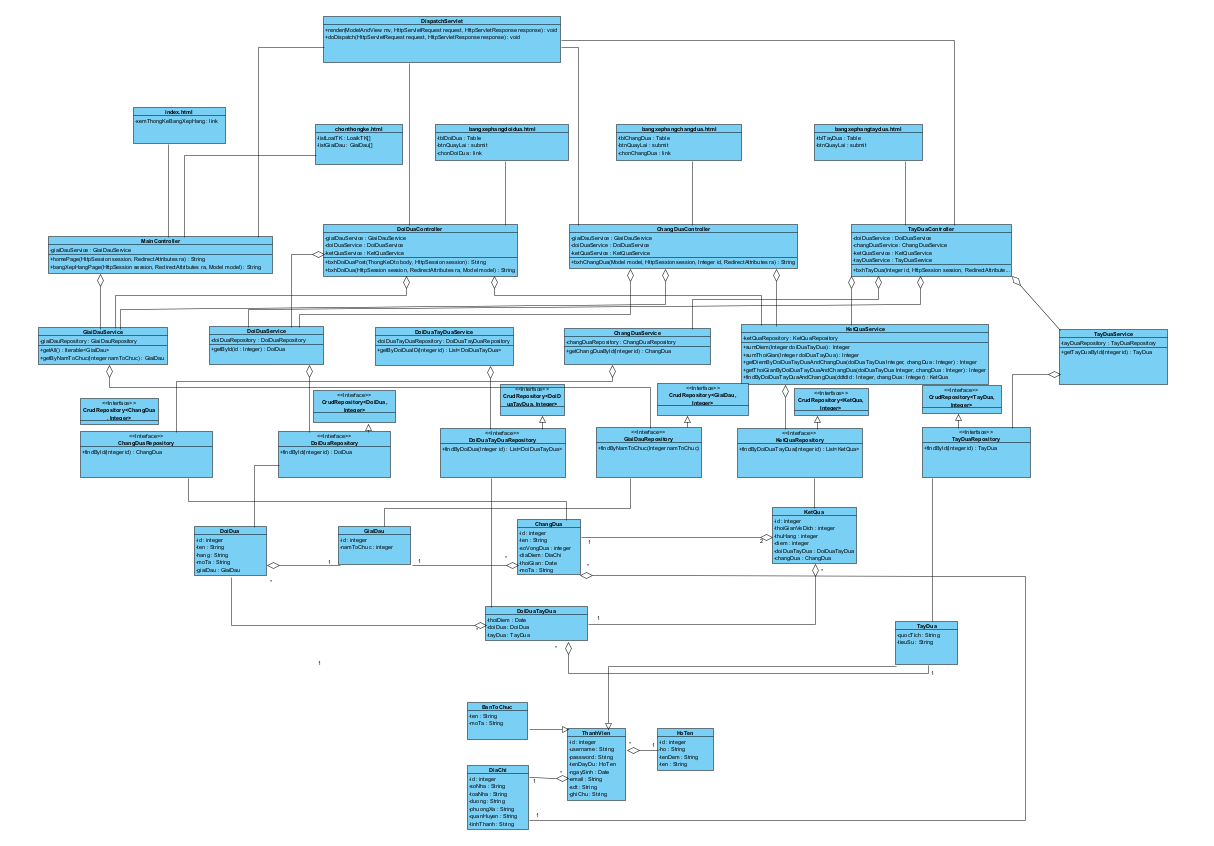


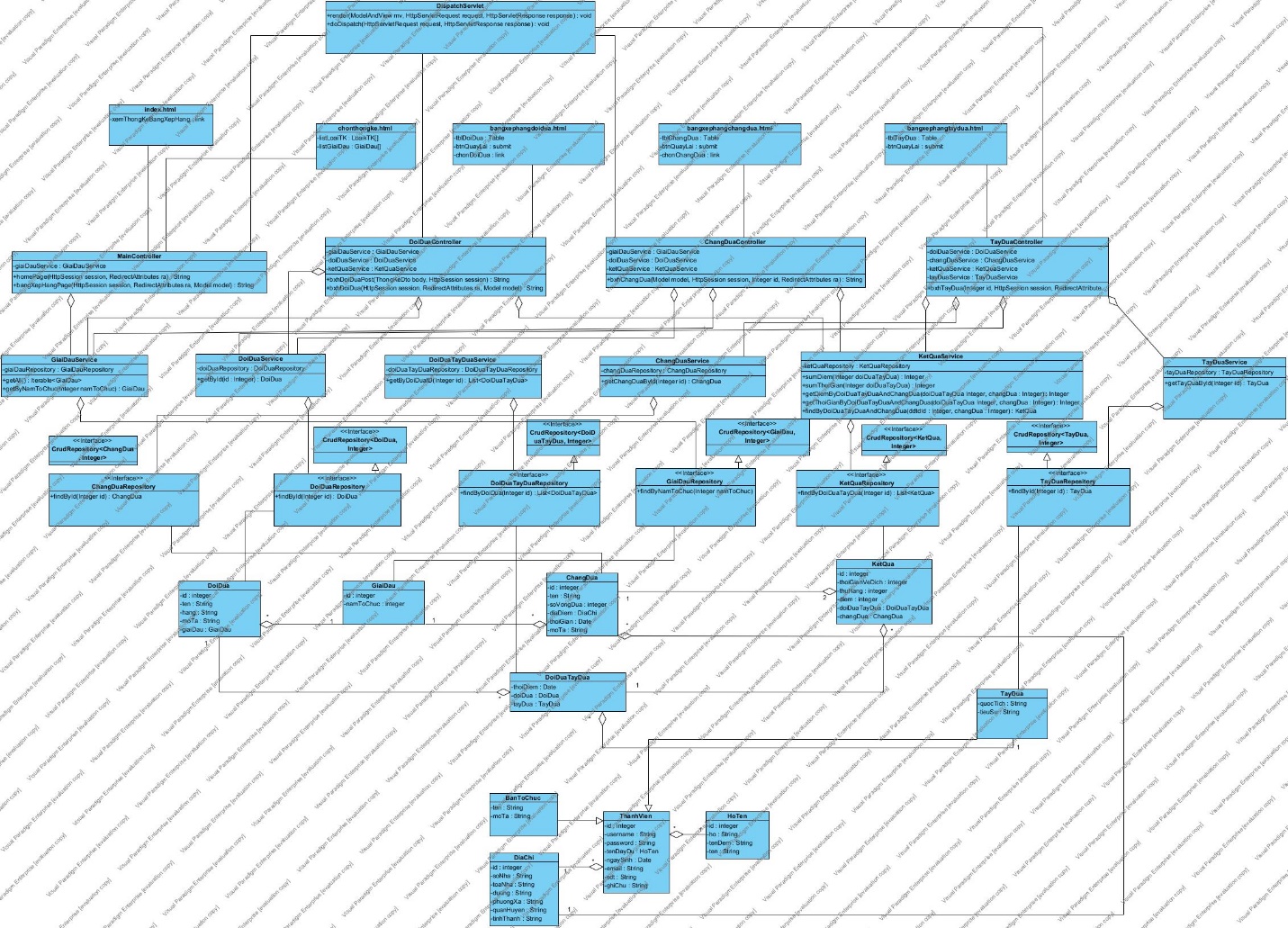
1. **Thiết kế tĩnh giao diện web**

****

1. **Biểu đồ thiết kế lớp thực thể**

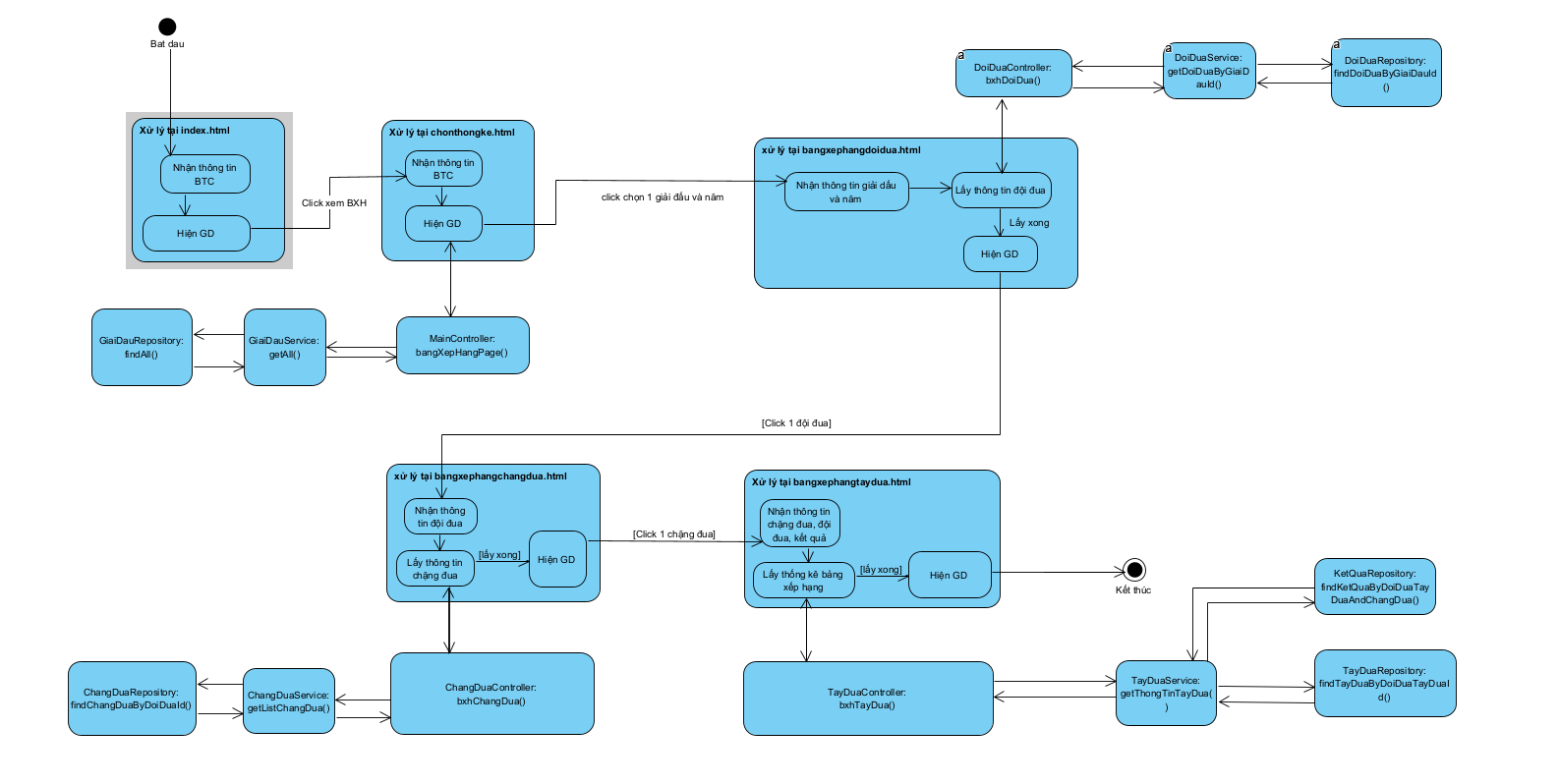
* Giao diện thiết kế module xem bảng xếp hạng các đội đua được thiết kế như hình dưới:
  + Tầng giao diện html sử dụng template engine là thymeleaf gồm các trang: index.html, chonthongke.html, bangxephangdoidua.html, bangxephangchangdua.html, bangxephangtaydua.html.
  + Tầng nhận request từ client và định tuyến router gồm các lớp MainController.java, DoiDuaController.java, ChangDuaController.java, TayDuaController.java.
  + Tầng xử lý logic có các lớp GiaiDauService.java, DoiDuaService.java, ChangDuaService.java, TayDuaService.java, KetQuaService.java, DoiDuaTayDuaService.java.
  + Tầng truy cập dữ liệu có các interface GiaiDauRepository.java, DoiDuaRepository.java, ChangDuaRepository.java, KetQuaRepository.java, TayDuaRepository.java kế thừa từ CrudRepository
  + Các thự thể có annotation @Entity map vào các table trong csdl: GiaiDau, KetQua, DoiDua, ChangDua, TayDua, ThanhVien, BanToChuc, DoiDuaTayDua.
  + Tất cả xử lý từ client qua server sẽ được thông qua một lớp DispatcherServlet – cho phép xử lý tất cả các HTTP request và response gồm nhiều method nhưng em liệt kê ra 2 method chính đại diện là:
    - [**render**](https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/web/servlet/DispatcherServlet.html#render(org.springframework.web.servlet.ModelAndView,jakarta.servlet.http.HttpServletRequest,jakarta.servlet.http.HttpServletResponse))([ModelAndView](https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/web/servlet/ModelAndView.html) mv, [HttpServletRequest](https://jakarta.ee/specifications/platform/9/apidocs/jakarta/servlet/http/HttpServletRequest.html) request, [HttpServletResponse](https://jakarta.ee/specifications/platform/9/apidocs/jakarta/servlet/http/HttpServletResponse.html) response) => bước cuối cùng xử lý request xong và hiểu thị ra view
    - [**doDispatch**](https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/web/servlet/DispatcherServlet.html#doDispatch(jakarta.servlet.http.HttpServletRequest,jakarta.servlet.http.HttpServletResponse))([HttpServletRequest](https://jakarta.ee/specifications/platform/9/apidocs/jakarta/servlet/http/HttpServletRequest.html) request, [HttpServletResponse](https://jakarta.ee/specifications/platform/9/apidocs/jakarta/servlet/http/HttpServletResponse.html) response) => xử lý tất cả các http method(get, post, put, patch, delete)

****

* Em có Backup ảnh diagram vì sợ size quá nhỏ không thấy rõ chữ:
* 

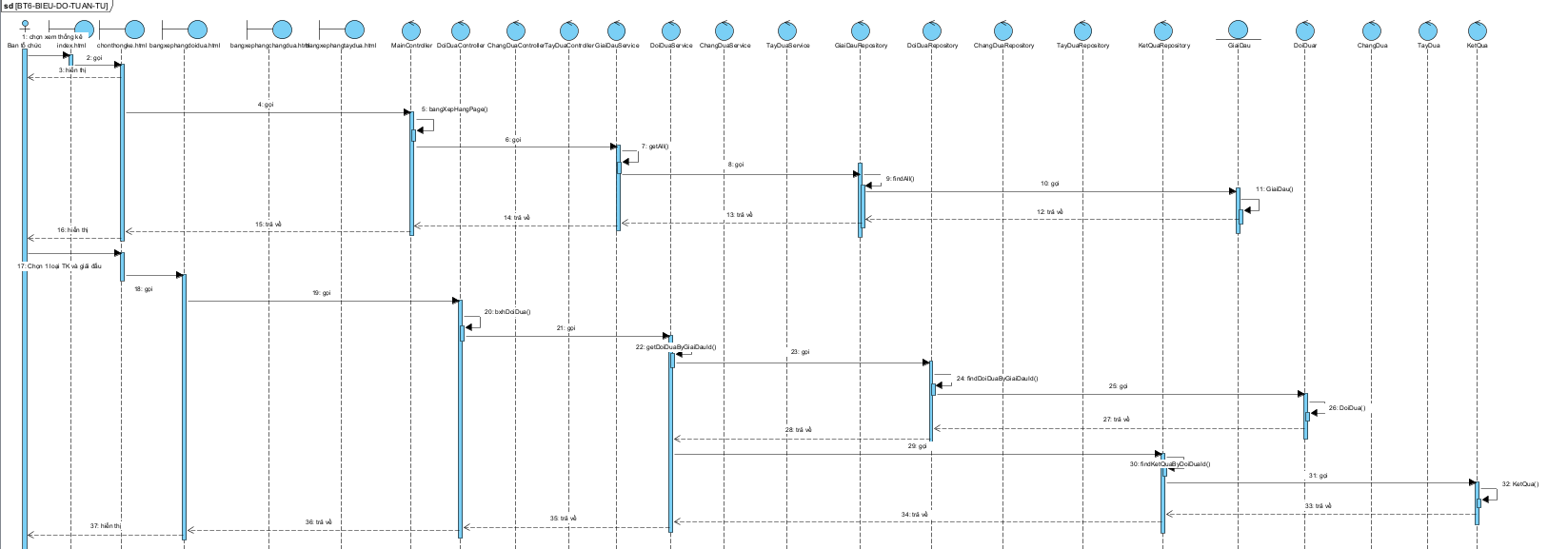
1. **Biểu đồ hoạt động của modul**

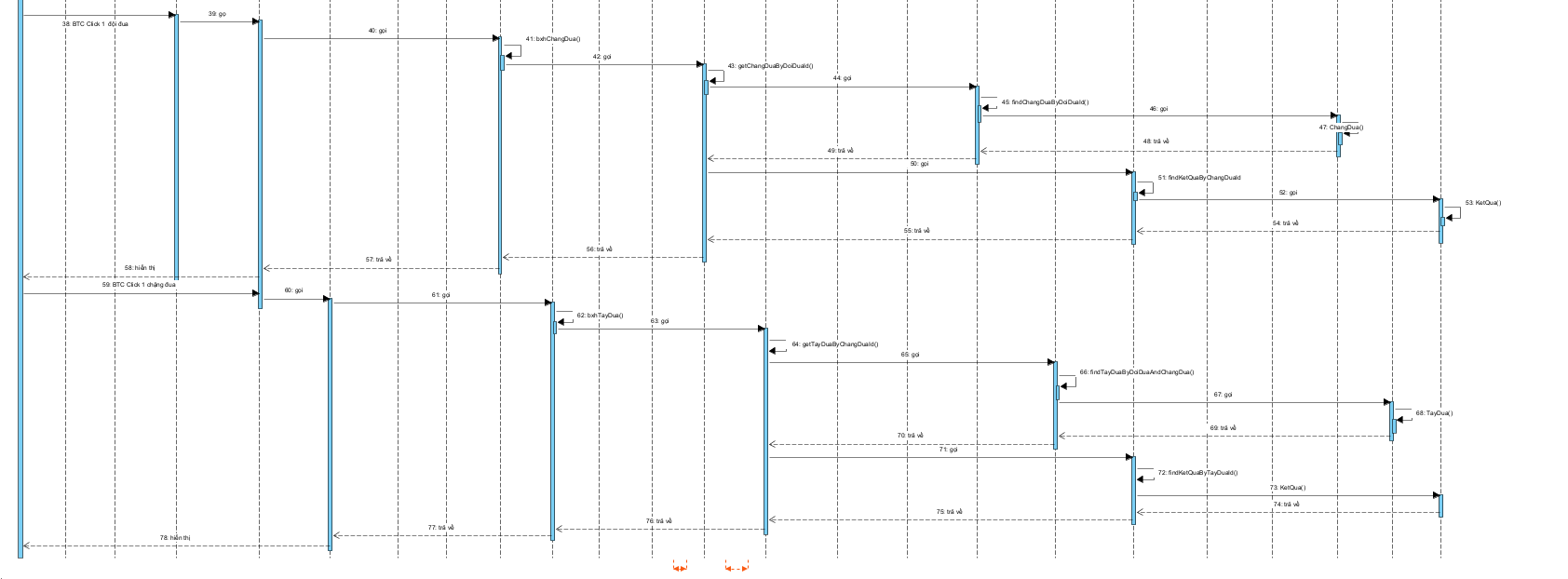
* Kịch bản các hoạt động được diễn ra như sau(vì lớp font controller là DispatchServlet được xây dựng dưới core của framework spring sẽ luôn nhận request và mapping route cho các controller trả về giao diện người dùng nên em sẽ không đề cập vào kịch bản chính)
  1. Tại trang index.html, sau khi đăng nhập, ban tổ chức chọn chức năng xem thống kê
  2. Trang index.html gọi trang chonthongke.html
  3. Trang chonthongke.html hiển thị cho ban tổ chức
  4. Ban tổ chức click xem xem bảng xếp hạng đội đua và giải đấu hiện tại(2022)
  5. Trang chonthongke.html gọi trang bangxephangdoidua.html
  6. Trang bangxephangdoidua.html gọi lớp DoiDuaController yêu cầu lấy danh sách các đội đua theo năm tổ chức
  7. Lớp DoiDuaController gọi hàm bxhDoiDua()
  8. Hàm bxhDoiDua() thực hiện và gọi lớp DoiDuaService yêu cầu lấy dữ liệu
  9. Lớp DoiDuaService gọi hàm getDoiDuaByGiaiDauId()
  10. Hàm getDoiDuaByGiaiDauId() gọi Interface DoiDuaRepository yêu cầu lấy dữ liệu và xử lý logic
  11. Interface DoiDuaRepository gọi hàm findDoiDuaByGiaiDauId()
  12. Hàm findDoiDuaByGiaiDauId () gọi lớp DoiDua đóng gói thông tin
  13. Lớp DoiDua đóng gói thông tin thực thể
  14. Lớp DoiDua trả về kết quả cho hàm findDoiDuaByGiaiDauId ()
  15. Hàm findDoiDuaByGiaiDauId () trả kết quả cho hàm getDoiDuaByGiaiDauId ()
  16. Hàm getDoiDuaByGiaiDauId () trả về kết quả cho hàm bxhDoiDua()
  17. Hàm bxhDoiDua() trả lại kết quả cho trang bangxephangdoidua.html
  18. Trang bangxephangdoidua.html hiển thị thông tin cho ban tổ chức
  19. Ban tổ chức click vào một đội đua
  20. Trang bangxephangdoidua.html gọi trang bangxephangchangdua.html
  21. Trang bangxephangchangdua.html gọi lớp ChangDuaController yêu cầu lấy danh sách các chặng đua mà đội đua đó tham gia
  22. Lớp ChangDuaController gọi hàm bxhChangDua()
  23. Hàm bxhChangDua() thực hiện và gọi lớp DoiDuaService yêu cầu lấy dữ liệu
  24. Lớp DoiDuaService gọi hàm getListChangDuaById()
  25. Hàm getListChangDuaById () gọi Interface DoiDuaRepository yêu cầu lấy dữ liệu và xử lý logic
  26. Interface DoiDuaRepository gọi hàm findListChangDuaById()
  27. Hàm findListChangDuaById () gọi lớp DoiDua đóng gói thông tin
  28. Lớp DoiDua đóng gói thông tin thực thể
  29. Lớp DoiDua trả về kết quả cho hàm findListChangDuaById ()
  30. Hàm findListChangDuaById() trả kết quả cho hàm getListChangDuaById()
  31. Hàm getListChangDuaById() trả về kết quả cho hàm bxhChangDua()
  32. Hàm bxhChangDua() trả lại kết quả cho trang bangxephangchangdua.html
  33. Trang bangxephangchangdua.html hiển thị thông tin cho ban tổ chức
  34. Ban tổ chức click vào một chặng đua
  35. Trang bangxephangchangdua.html gọi trang bangxephangtaydua.html
  36. Trang bangxephangtaydua.html gọi lớp TayDuaController yêu cầu lấy danh sách các tay đua thuộc đội đua đã chọn tham gia chặng đua đó
  37. Lớp TayDuaController gọi hàm bxhTayDua()
  38. Hàm bxhTayDua() thực hiện và gọi lớp ChangDuaService, DoiDuaService, KetQuaService, TayDuaService yêu cầu lấy dữ liệu về các thông tin của tay đua thuộc đội đua và tham gia chặng đua đó kèm theo thông tin kết quả(điểm, thời gian về đích) của từng tay đua
  39. Các lớp ChangDuaService, DoiDuaService, KetQuaService, TayDuaService gọi các hàm getTayDua(), getChangDua(), getDoiDua(), … để gọi các dữ liệu riêng biệt để xử lý logic
  40. Các hàm getTayDua(), getChangDua(), getDoiDua() gọi các interface ChangDuaRepository, DoiDuaRepository, KetQuaRepository, TayDuaRepository để yêu cầu lấy dữ liệu
  41. Các interface gọi các hàm findTayDuaById(), findKetQuaByDoiDuaTayDuaAndChangDua(),…
  42. các hàm findTayDuaById(), findKetQuaByDoiDuaTayDuaAndChangDua() gọi các lớp thực thể ChangDua, KetQua, DoiDua… để lấy dữ liệu
  43. Các lớp thực thể ChangDua, KetQua, DoiDua… trả về kết quả cho các interface ChangDuaRepository, DoiDuaRepository, KetQuaRepository, TayDuaRepository.
  44. Các hàm của các interface ChangDuaRepository, DoiDuaRepository, KetQuaRepository, TayDuaRepository trả về kết quả cho các hàm Service tương ứng để xử lý logic
  45. Các hàm xử lý logic ở các service sẽ trả về cho bxhTayDua()
  46. HàmbxhTayDua() trả lại kết qua cho trang bangxephangtaydua.html
  47. Hàm getListChangDuaById() trả về kết quả cho hàm bxhChangDua()
  48. Hàm bxhChangDua() trả lại kết quả cho trang bangxephangchangdua.html
  49. Trang bangxephangchangdua.html hiển thị thông tin cho ban tổ chức
* Biểu đồ hoạt động:

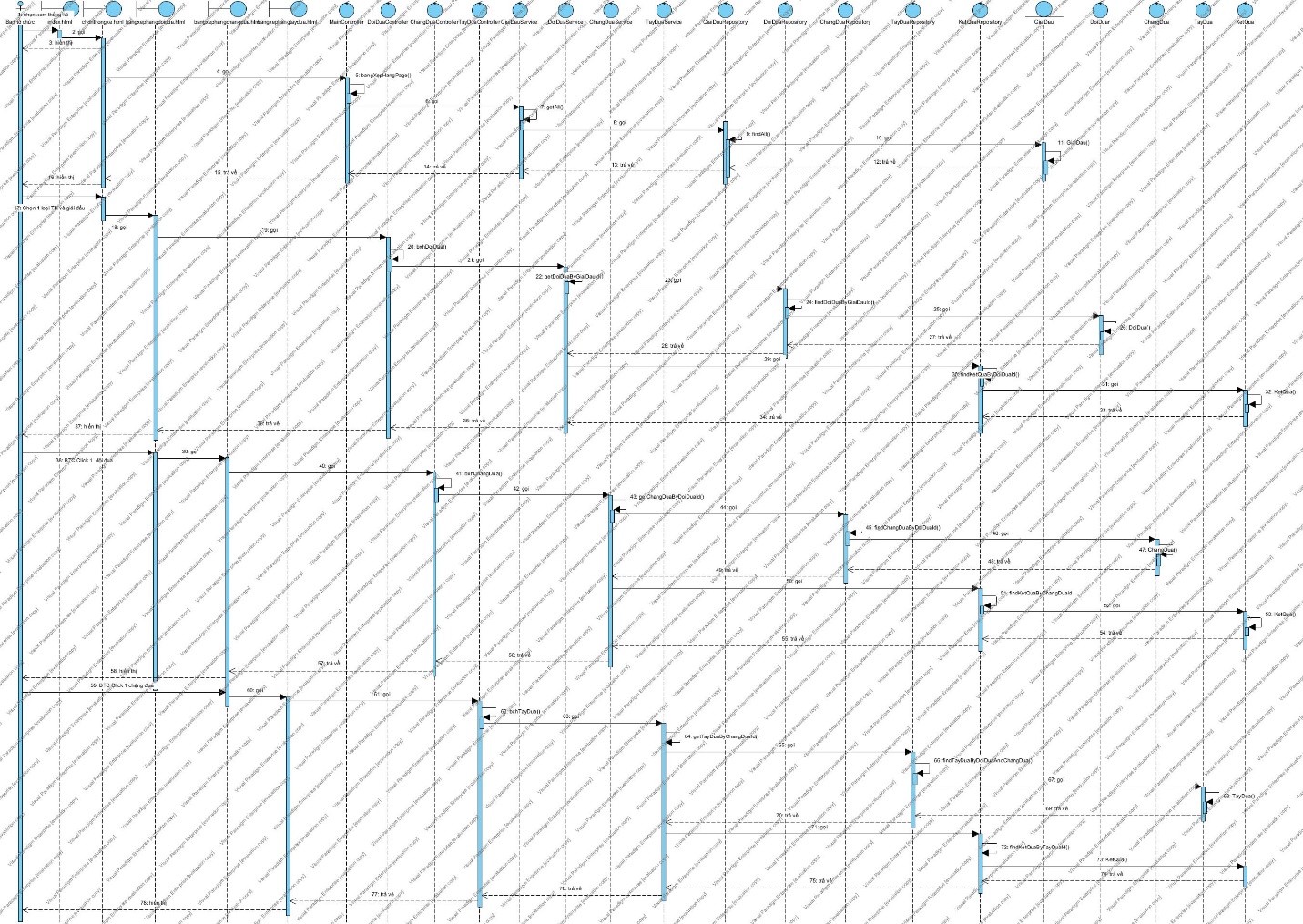
****

🡺 Biểu đồ tuần tự: Vì quá to nên em cắt thành 2 ảnh nhỏ

Nửa ảnh phần trên:

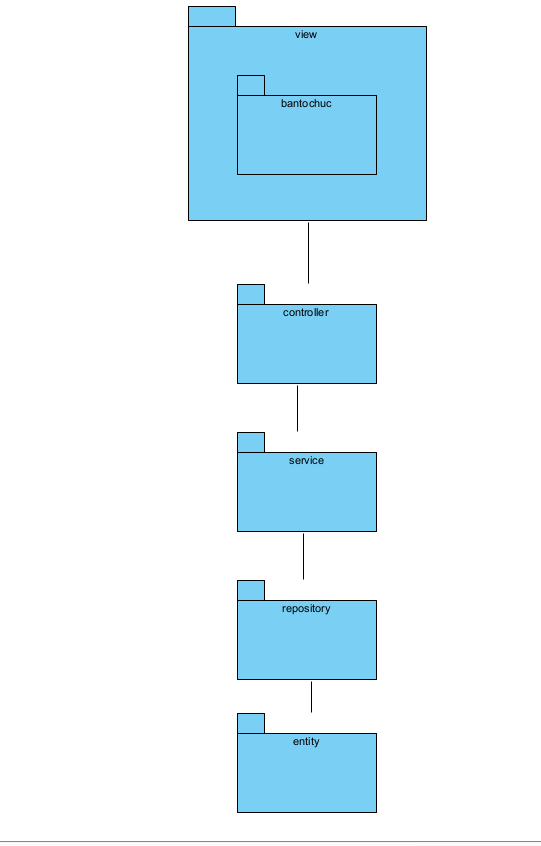




Ảnh backup: 

1. **Biểu đồ triển khai**

* Package view chứa package bantochuc trong đó có các trang html liên quan đến chức năng của ban tổ chức
* Package controller chứa các lớp controller nhận request từ client và trả về response cho client.
* Package service chứa các lớp service thực hiện xử lý logic business.
* Package repository chứa các interface repository kế thừa từ CrudRepository truy cập trực tiếp vào csdl.

****

1. **Cài đặt và triển khai hệ thống**
   1. **Code(Em sẽ chia code thành các modul nhỏ gồm các class Controller, Service, Entity, Dto(optional) và Inteface Repository**

AuthController.java

@Controller  
@RequestMapping("/auth")  
public class AuthController {  
 @Autowired private AuthService authService;  
  
 @GetMapping("/login")  
 public String loginPage(Model model) {  
 model.addAttribute("thanhVien", new ThanhVien());  
 return "login";  
 }  
  
 @PostMapping("/login")  
 public String loginAction(RedirectAttributes ra, @RequestParam("username") String username, @RequestParam("password") String password, HttpSession session) {  
 boolean isLoginSuccess = authService.authLogin(username, password);  
 if(isLoginSuccess) {  
 session.setAttribute("username", username);  
 return "redirect:/";  
 }  
 ra.addFlashAttribute("message", "Tài khoản hoặc mật khẩu không chính xác!");  
 return "redirect:/auth/login";  
 }  
}

AuthService.java

@Service  
public class AuthService {  
 @Autowired  
 private ThanhVienRepository thanhVienRepo;  
  
 public boolean authLogin(String username, String password) {  
 Optional<ThanhVien> tv = thanhVienRepo.findByUsername(username);  
 if(tv.isPresent() && tv.get().getPassword().equals(password)) {  
 return true;  
 }  
 return false;  
 }  
}

BanToChuc.java

@Entity  
@Table(name = "tblbantochuc")  
public class BanToChuc {  
 @Id  
 @Column(name = "tbl\_thanh\_vien\_id")  
 private Integer id;  
  
 @Column(name = "ten", nullable = false)  
 String ten;  
 @Column(name = "mo\_ta", nullable = false)  
 String moTa;  
  
 @OneToOne(cascade = CascadeType.*MERGE*, fetch = FetchType.*EAGER*)  
 @MapsId  
 @JoinColumn(name = "tbl\_thanh\_vien\_id")  
 private ThanhVien thanhVien;  
  
 public Integer getId() {  
 return id;  
 }  
  
 public void setId(Integer id) {  
 this.id = id;  
 }  
  
 public String getTen() {  
 return ten;  
 }  
  
 public void setTen(String ten) {  
 this.ten = ten;  
 }  
  
 public String getMoTa() {  
 return moTa;  
 }  
  
 public void setMoTa(String moTa) {  
 this.moTa = moTa;  
 }  
  
 public ThanhVien getThanhVien() {  
 return thanhVien;  
 }  
  
 public void setThanhVien(ThanhVien thanhVien) {  
 this.thanhVien = thanhVien;  
 }  
  
 @Override  
 public String toString() {  
 return "BanToChuc{" +  
 "id=" + id +  
 ", ten='" + ten + '\'' +  
 ", moTa='" + moTa + '\'' +  
 ", thanhVien=" + thanhVien +  
 '}';  
 }  
}

BanToChucRepository.java

public interface BanToChucRepository extends CrudRepository<BanToChuc, Integer> {  
}

ChangDua.java

@Entity  
@Table(name = "tblchangdua")  
public class ChangDua {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 Integer id;  
  
 @Column(name = "ten", nullable = false)  
 String ten;  
  
 @Column(name = "so\_vong\_dua", nullable = false)  
 Integer soVongDua;  
  
 @Column(name = "thoi\_gian", nullable = false)  
 Date thoiGian;  
  
 @Column(name = "mo\_ta", nullable = false)  
 String moTa;  
  
 @Column(name = "dia\_diem")  
 String diaDiem;  
  
 @ManyToOne(cascade = CascadeType.*MERGE*, fetch = FetchType.*EAGER*)  
 @JoinColumn(name = "tbl\_giai\_dau\_id")  
 GiaiDau giaiDau;  
  
 @LazyCollection(LazyCollectionOption.*FALSE*)  
 @OneToMany(mappedBy = "changDua", fetch = FetchType.*EAGER*,cascade = CascadeType.*PERSIST*)  
 List<KetQua> listKetQua;  
  
 public Integer getId() {  
 return id;  
 }  
  
 public void setId(Integer id) {  
 this.id = id;  
 }  
  
 public String getTen() {  
 return ten;  
 }  
  
 public void setTen(String ten) {  
 this.ten = ten;  
 }  
  
 public Integer getSoVongDua() {  
 return soVongDua;  
 }  
  
 public void setSoVongDua(Integer soVongDua) {  
 this.soVongDua = soVongDua;  
 }  
  
 public Date getThoiGian() {  
 return thoiGian;  
 }  
  
 public void setThoiGian(Date thoiGian) {  
 this.thoiGian = thoiGian;  
 }  
  
 public String getMoTa() {  
 return moTa;  
 }  
  
 public void setMoTa(String moTa) {  
 this.moTa = moTa;  
 }  
  
 public String getDiaDiem() {  
 return diaDiem;  
 }  
  
 public void setDiaDiem(String diaDiem) {  
 this.diaDiem = diaDiem;  
 }  
  
 public GiaiDau getGiaiDau() {  
 return giaiDau;  
 }  
  
 public void setGiaiDau(GiaiDau giaiDau) {  
 this.giaiDau = giaiDau;  
 }  
  
 public List<KetQua> getListKetQua() {  
 return listKetQua;  
 }  
  
 public void setListKetQua(List<KetQua> listKetQua) {  
 this.listKetQua = listKetQua;  
 }  
  
 @Override  
 public String toString() {  
 return "ChangDua{" +  
 "id=" + id +  
 ", ten='" + ten + '\'' +  
 ", soVongDua=" + soVongDua +  
 ", thoiGian=" + thoiGian +  
 ", moTa='" + moTa + '\'' +  
 ", diaDiem='" + diaDiem + '\'' +  
 ", giaiDau=" + giaiDau +  
 ", listKetQua=" + listKetQua +  
 '}';  
 }  
}

ChangDuaController.java

@Controller  
@RequestMapping("bang-xep-hang-chang-dua")  
public class ChangDuaController {  
  
 @Autowired  
 DoiDuaService doiDuaService;  
 @Autowired  
 GiaiDauService giaiDauService;  
  
 @Autowired  
 KetQuaService ketQuaService;  
  
 @GetMapping("/{id}")  
 public String bxhChangDua(Model model, HttpSession session, @PathVariable("id") Integer id, RedirectAttributes ra) {  
 if(session.getAttribute("username") == null) {  
 ra.addFlashAttribute("requireLogin", "Vui lòng đăng nhập để xem bảng xếp hạng chặng đua");  
 return "redirect:/auth/login";  
 } else {  
 session.setAttribute("doiDuaId", id);  
 Optional<DoiDua> doiDua = doiDuaService.getById(id);  
 Set<DoiDuaTayDua> listDDTD = doiDua.get().getListDoiDuaTayDua();  
 Integer namToChuc = (Integer) session.getAttribute("namToChuc");  
 Optional<GiaiDau> giaiDau = giaiDauService.getByNamToChuc(namToChuc);  
 Set<ChangDua> listChangDua = giaiDau.get().getListChangDua();  
 List<ChangDuaDto> listChangDuaDto = new ArrayList<>();  
 for(ChangDua changDua: listChangDua) {  
 Integer tongDiem = 0;  
 Integer tongThoiGian = 0;  
 for(DoiDuaTayDua doiDuaTayDua: listDDTD) {  
 tongDiem += ketQuaService.getDiemByDoiDuaTayDuaAndChangDua(doiDuaTayDua.getId(), changDua.getId());  
 tongThoiGian += ketQuaService.getThoiGianByDoiDuaTayDuaAndChangDua(doiDuaTayDua.getId(), changDua.getId());  
 }  
 listChangDuaDto.add(new ChangDuaDto(changDua.getId(), changDua.getTen(), tongDiem, tongThoiGian));  
 }  
 Collections.*sort*(listChangDuaDto, new Comparator<ChangDuaDto>() {  
 @Override  
 public int compare(ChangDuaDto o1, ChangDuaDto o2) {  
 return o2.getTongDiem() - o1.getTongDiem();  
 }  
 });  
 model.addAttribute("listChangDuaDto",listChangDuaDto);  
 model.addAttribute("namToChuc", namToChuc);  
 model.addAttribute("tenDoiDua", doiDua.get().getTen());  
 return "bangxephangchangdua";  
 }  
 }  
  
}

ChangDuaDto.java

public class ChangDuaDto {  
 private Integer id;  
 String ten;  
 Integer tongDiem;  
 Integer tongThoiGian;  
  
 public ChangDuaDto(Integer id, String ten, Integer tongDiem, Integer tongThoiGian) {  
 this.id = id;  
 this.ten = ten;  
 this.tongDiem = tongDiem;  
 this.tongThoiGian = tongThoiGian;  
 }  
  
 public Integer getId() {  
 return id;  
 }  
  
 public void setId(Integer id) {  
 this.id = id;  
 }  
  
 public String getTen() {  
 return ten;  
 }  
  
 public void setTen(String ten) {  
 this.ten = ten;  
 }  
  
 public Integer getTongDiem() {  
 return tongDiem;  
 }  
  
 public void setTongDiem(Integer tongDiem) {  
 this.tongDiem = tongDiem;  
 }  
  
 public Integer getTongThoiGian() {  
 return tongThoiGian;  
 }  
  
 public void setTongThoiGian(Integer tongThoiGian) {  
 this.tongThoiGian = tongThoiGian;  
 }  
}

ChangDuaRepository.java

public interface ChangDuaRepository extends CrudRepository<ChangDua, Integer> {  
  
 public Optional<ChangDua> findById(Integer id);  
}

ChangDuaService.java

@Service  
public class ChangDuaService {  
 @Autowired ChangDuaRepository changDuaRepository;  
  
 public Optional<ChangDua> getChangDuaById(Integer id) {  
 return changDuaRepository.findById(id);  
 }  
  
}

DoiDua.java

@Entity  
@Table(name = "tbldoidua")  
public class DoiDua {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Integer id;  
  
 @Column(name = "ten", nullable = false)  
 String ten;  
  
 @Column(name = "hang", nullable = false)  
 String hang;  
  
 @Column(name = "mo\_ta", nullable = false)  
 String moTa;  
  
 @ManyToOne  
 @JoinColumn(name = "tbl\_giai\_dau\_id")  
 GiaiDau giaiDau;  
  
 @LazyCollection(LazyCollectionOption.*FALSE*)  
 @OneToMany(mappedBy = "doiDua", fetch = FetchType.*EAGER*,cascade = CascadeType.*PERSIST*)  
 Set<DoiDuaTayDua> listDoiDuaTayDua;  
  
 public Integer getId() {  
 return id;  
 }  
  
 public void setId(Integer id) {  
 this.id = id;  
 }  
  
 public String getTen() {  
 return ten;  
 }  
  
 public void setTen(String ten) {  
 this.ten = ten;  
 }  
  
 public String getHang() {  
 return hang;  
 }  
  
 public void setHang(String hang) {  
 this.hang = hang;  
 }  
  
 public String getMoTa() {  
 return moTa;  
 }  
  
 public void setMoTa(String moTa) {  
 this.moTa = moTa;  
 }  
  
 public GiaiDau getGiaiDau() {  
 return giaiDau;  
 }  
  
 public void setGiaiDau(GiaiDau giaiDau) {  
 this.giaiDau = giaiDau;  
 }  
  
 public Set<DoiDuaTayDua> getListDoiDuaTayDua() {  
 return listDoiDuaTayDua;  
 }  
  
 public void setListDoiDuaTayDua(Set<DoiDuaTayDua> listDoiDuaTayDua) {  
 this.listDoiDuaTayDua = listDoiDuaTayDua;  
 }  
  
}

DoiDuaController.java

@Controller  
@RequestMapping("bang-xep-hang-doi-dua")  
public class DoiDuaController {  
 @Autowired GiaiDauService giaiDauService;  
 @Autowired DoiDuaService doiDuaService;  
 @Autowired  
 DoiDuaTayDuaService doiDuaTayDuaService;  
 @Autowired  
 KetQuaService ketQuaService;  
  
 @PostMapping("")  
 public String bxhDoiDuaPost(@RequestBody ThongKeDto body, HttpSession session) {  
 Integer namToChuc = body.getGiaiDau();  
 Optional<GiaiDau> giaiDau = giaiDauService.getByNamToChuc(namToChuc);  
 Set<DoiDua> listDoiDua = giaiDau.get().getListDoiDua();  
 session.setAttribute("listDoiDua", listDoiDua);  
 session.setAttribute("namToChuc", namToChuc);  
 return "bangxephangdoidua";  
 }  
  
 @GetMapping("")  
 public String bxhDoiDua(HttpSession session, RedirectAttributes ra, Model model) {  
 System.*out*.println(session);  
 if(session.getAttribute("username") == null) {  
 ra.addFlashAttribute("requireLogin", "Vui lòng đăng nhập để xem bảng xếp hạng!");  
 return "redirect:/auth/login";  
 } else {  
 Integer namToChuc = (Integer) session.getAttribute("namToChuc");  
 Set<DoiDua> listDoiDua = giaiDauService.getByNamToChuc(namToChuc).get().getListDoiDua();  
 List<DoiDuaDto> listDoiDuaDto = new ArrayList<>();  
 for(DoiDua dd: listDoiDua) {  
 Optional<DoiDua> doiDua = doiDuaService.getById(dd.getId());  
 System.*out*.println(doiDua);  
 Set<DoiDuaTayDua> listDDTD = doiDua.get().getListDoiDuaTayDua();  
 Integer tongDiem = 0;  
 Integer tongThoiGian = 0;  
 for(DoiDuaTayDua ddtd: listDDTD) {  
 System.*out*.println(ddtd);  
 Integer sumDiem = ketQuaService.sumDiem(ddtd.getId());  
 Integer sumThoiGian = ketQuaService.sumThoiGian(ddtd.getId());  
 tongDiem += sumDiem;  
 tongThoiGian += sumThoiGian;  
 }  
 listDoiDuaDto.add(new DoiDuaDto(dd.getId(), dd.getTen(), dd.getHang(), tongDiem, tongThoiGian));  
 }  
 Collections.*sort*(listDoiDuaDto, new Comparator<DoiDuaDto>() {  
 @Override  
 public int compare(DoiDuaDto o1, DoiDuaDto o2) {  
 return o2.getTongDiem() - o1.getTongDiem();  
 }  
 });  
 model.addAttribute("namToChuc", namToChuc);  
 model.addAttribute("listDoiDuaDto", listDoiDuaDto);  
 return "bangxephangdoidua";  
 }  
 }  
  
}

DoiDuaDto.java

public class DoiDuaDto {  
 private Integer id;  
  
 String ten;  
  
 String hang;  
  
 Integer tongDiem;  
  
 Integer tongThoiGian;  
  
 public Integer getId() {  
 return id;  
 }  
  
 public void setId(Integer id) {  
 this.id = id;  
 }  
  
 public String getTen() {  
 return ten;  
 }  
  
 public void setTen(String ten) {  
 this.ten = ten;  
 }  
  
 public String getHang() {  
 return hang;  
 }  
  
 public void setHang(String hang) {  
 this.hang = hang;  
 }  
  
 public Integer getTongDiem() {  
 return tongDiem;  
 }  
  
 public void setTongDiem(Integer tongDiem) {  
 this.tongDiem = tongDiem;  
 }  
  
 public Integer getTongThoiGian() {  
 return tongThoiGian;  
 }  
  
 public void setTongThoiGian(Integer tongThoiGian) {  
 this.tongThoiGian = tongThoiGian;  
 }  
  
 public DoiDuaDto(Integer id, String ten, String hang, Integer tongDiem, Integer tongThoiGian) {  
 this.id = id;  
 this.ten = ten;  
 this.hang = hang;  
 this.tongDiem = tongDiem;  
 this.tongThoiGian = tongThoiGian;  
 }  
  
}

DoiDuaRepository.java

public interface DoiDuaRepository extends CrudRepository<DoiDua, Integer> {  
 public Optional<DoiDua> findById(Integer id);  
}

DoiDuaService.java

@Service  
public class DoiDuaService {  
 @Autowired DoiDuaRepository doiDuaRepository;  
  
 public Optional<DoiDua> getById(Integer id) {  
 return doiDuaRepository.findById(id);  
 }  
  
}

DoiDuaTayDua.java

@Entity  
@Table(name = "tbldoiduataydua")  
public class DoiDuaTayDua {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 Integer id;  
  
 @Column(name = "thoi\_diem", nullable = false)  
 Date thoiDiem;  
  
 @ManyToOne(cascade = CascadeType.*MERGE*, fetch = FetchType.*EAGER*)  
 @JoinColumn(name = "tbl\_thanh\_vien\_id")  
 TayDua tayDua;  
  
 @ManyToOne(cascade = CascadeType.*MERGE*, fetch = FetchType.*EAGER*)  
 @JoinColumn(name = "tbl\_doi\_dua\_id")  
 DoiDua doiDua;  
  
 @OneToMany(mappedBy = "doiDuaTayDua", fetch = FetchType.*EAGER*,cascade = CascadeType.*PERSIST*)  
 List<KetQua> listKetQua;  
  
 public Integer getId() {  
 return id;  
 }  
  
 public void setId(Integer id) {  
 this.id = id;  
 }  
  
 public Date getThoiDiem() {  
 return thoiDiem;  
 }  
  
 public void setThoiDiem(Date thoiDiem) {  
 this.thoiDiem = thoiDiem;  
 }  
  
 public TayDua getTayDua() {  
 return tayDua;  
 }  
  
 public void setTayDua(TayDua tayDua) {  
 this.tayDua = tayDua;  
 }  
  
 public DoiDua getDoiDua() {  
 return doiDua;  
 }  
  
 public void setDoiDua(DoiDua doiDua) {  
 this.doiDua = doiDua;  
 }  
  
 public List<KetQua> getListKetQua() {  
 return listKetQua;  
 }  
  
 public void setListKetQua(List<KetQua> listKetQua) {  
 this.listKetQua = listKetQua;  
 }  
  
}

DoiDuaTayDuaRepository.java

@Repository  
public interface DoiDuaTayDuaRepository extends CrudRepository<DoiDuaTayDua, Integer> {  
 public Optional<List<DoiDuaTayDua>> findByDoiDua(Integer id);  
}

DoiDuaTayDuaService.java

@Service  
public class DoiDuaTayDuaService {  
 @Autowired DoiDuaTayDuaRepository doiDuaTayDuaRepository;  
  
 public Optional<List<DoiDuaTayDua>> getByDoiDuaID(Integer id) {  
 return doiDuaTayDuaRepository.findByDoiDua(id);  
 }  
  
}

GiaiDau.java

@Entity  
@Table(name = "tblgiaidau")  
public class GiaiDau {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Integer id;  
  
 @Column(name = "nam\_to\_chuc")  
 private Integer namToChuc;  
  
 @LazyCollection(LazyCollectionOption.*FALSE*)  
 @OneToMany(mappedBy = "giaiDau",fetch = FetchType.*EAGER*, cascade = CascadeType.*PERSIST*)  
 Set<ChangDua> listChangDua;  
  
 @LazyCollection(LazyCollectionOption.*FALSE*)  
 @OneToMany(mappedBy = "giaiDau", fetch = FetchType.*EAGER*,cascade = CascadeType.*PERSIST*)  
 Set<DoiDua> listDoiDua;  
  
 @Override  
 public String toString() {  
 return "GiaiDau{" +  
 "id=" + id +  
 ", namToChuc=" + namToChuc +  
 '}';  
 }  
  
 public Integer getId() {  
 return id;  
 }  
  
 public void setId(Integer id) {  
 this.id = id;  
 }  
  
 public Integer getNamToChuc() {  
 return namToChuc;  
 }  
  
 public void setNamToChuc(Integer namToChuc) {  
 this.namToChuc = namToChuc;  
 }  
  
 public Set<ChangDua> getListChangDua() {  
 return listChangDua;  
 }  
  
 public void setListChangDua(Set<ChangDua> listChangDua) {  
 this.listChangDua = listChangDua;  
 }  
  
 public Set<DoiDua> getListDoiDua() {  
 return listDoiDua;  
 }  
  
 public void setListDoiDua(Set<DoiDua> listDoiDua) {  
 this.listDoiDua = listDoiDua;  
 }  
}

GiaiDauRepository.java

public interface GiaiDauRepository extends CrudRepository<GiaiDau, Integer> {  
 public Optional<GiaiDau> findByNamToChuc(Integer namToChuc);  
}

GiaiDauService.java

@Service  
public class GiaiDauService {  
 @Autowired GiaiDauRepository giaiDauRepository;  
  
 public Iterable<GiaiDau> getAll() {  
 return giaiDauRepository.findAll();  
 }  
  
 public Optional<GiaiDau> getByNamToChuc(Integer namToChuc) {  
 return giaiDauRepository.findByNamToChuc(namToChuc);  
 }  
  
}

KetQua.java

@Entity  
@Table(name = "tblketqua")  
public class KetQua {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Integer id;  
  
 @Column(name = "thoi\_gian\_ve\_dich", nullable = false)  
 Integer thoiGianVeDich;  
  
 @Column(name = "thu\_hang", nullable = false)  
 Integer thuHang;  
  
 @Column(name = "diem", nullable = false)  
 Integer diem;  
  
 @ToStringExclude  
 @ManyToOne(cascade = CascadeType.*MERGE*, fetch = FetchType.*EAGER*)  
 @JoinColumn(name = "tbl\_doi\_dua\_tay\_dua\_id")  
 DoiDuaTayDua doiDuaTayDua;  
  
 @ManyToOne(cascade = CascadeType.*MERGE*, fetch = FetchType.*EAGER*)  
 @JoinColumn(name = "tbl\_chang\_dua\_id")  
 ChangDua changDua;  
  
 public Integer getId() {  
 return id;  
 }  
  
 public void setId(Integer id) {  
 this.id = id;  
 }  
  
 public Integer getThoiGianVeDich() {  
 return thoiGianVeDich;  
 }  
  
 public void setThoiGianVeDich(Integer thoiGianVeDich) {  
 this.thoiGianVeDich = thoiGianVeDich;  
 }  
  
 public Integer getThuHang() {  
 return thuHang;  
 }  
  
 public void setThuHang(Integer thuHang) {  
 this.thuHang = thuHang;  
 }  
  
 public Integer getDiem() {  
 return diem;  
 }  
  
 public void setDiem(Integer diem) {  
 this.diem = diem;  
 }  
  
 public DoiDuaTayDua getDoiDuaTayDua() {  
 return doiDuaTayDua;  
 }  
  
 public void setDoiDuaTayDua(DoiDuaTayDua doiDuaTayDua) {  
 this.doiDuaTayDua = doiDuaTayDua;  
 }  
  
 public ChangDua getChangDua() {  
 return changDua;  
 }  
  
 public void setChangDua(ChangDua changDua) {  
 this.changDua = changDua;  
 }  
  
}

KetQuaRepository.java

@Repository  
public interface KetQuaRepository extends CrudRepository<KetQua, Integer> {  
 public Optional<Set<KetQua>> findByDoiDuaTayDua(Integer id);  
  
 @Query(value = "SELECT sum(diem) FROM KetQua where doiDuaTayDua.id = ?1")  
 public Integer sumDiem(Integer doiDuaTayDua);  
  
 @Query(value = "SELECT sum(thoiGianVeDich) FROM KetQua where doiDuaTayDua.id = ?1")  
 public Integer sumThoiGian(Integer doiDuaTayDua);  
  
 @Query(value = "SELECT diem FROM KetQua where doiDuaTayDua.id = ?1 and changDua.id = ?2")  
 public Integer getDiemByDoiDuaTayDuaAndChangDua(Integer doiDuaTayDua, Integer changDua);  
  
 @Query(value = "SELECT thoiGianVeDich FROM KetQua where doiDuaTayDua.id = ?1 and changDua.id = ?2")  
 public Integer getThoiGianByDoiDuaTayDuaAndChangDua(Integer doiDuaTayDua, Integer changDua);  
  
 @Query(value = "SELECT t FROM KetQua t where t.doiDuaTayDua.id = ?1 and t.changDua.id = ?2")  
 public Optional<KetQua> findByDoiDuaTayDuaAndChangDua(Integer ddtdId, Integer changDuaId);  
  
}

KetQuaService.java

@Service  
public class KetQuaService {  
  
 @Autowired KetQuaRepository ketQuaRepository;  
 public Optional<Set<KetQua>> getByDoiDuaTayDua(Integer id) {  
 return ketQuaRepository.findByDoiDuaTayDua(id);  
 }  
  
 public Integer sumDiem(Integer doiDuaTayDua) {  
 return ketQuaRepository.sumDiem(doiDuaTayDua);  
 }  
 public Integer sumThoiGian(Integer doiDuaTayDua) {  
 return ketQuaRepository.sumThoiGian(doiDuaTayDua);  
 }  
  
 public Integer getDiemByDoiDuaTayDuaAndChangDua(Integer doiDuaTayDua, Integer changDua) {  
 return ketQuaRepository.getDiemByDoiDuaTayDuaAndChangDua(doiDuaTayDua, changDua);  
 }  
 public Integer getThoiGianByDoiDuaTayDuaAndChangDua(Integer doiDuaTayDua, Integer changDua) {  
 return ketQuaRepository.getThoiGianByDoiDuaTayDuaAndChangDua(doiDuaTayDua, changDua);  
 }  
  
 public Optional<KetQua> findByDoiDuaTayDuaAndChangDua(Integer ddtdId, Integer changDuaId) {  
 return ketQuaRepository.findByDoiDuaTayDuaAndChangDua(ddtdId, changDuaId);  
 }  
}

TayDua.java

@Entity  
@Table(name = "tbltaydua")  
public class TayDua {  
 @Id  
 @Column(name = "tbl\_thanh\_vien\_id")  
 Integer id;  
  
 @Column(name = "quoc\_tich", nullable = false)  
 String quocTich;  
 @Column(name = "tieu\_su", nullable = false)  
 String tieuSu;  
  
 @OneToOne  
 @MapsId  
 @JoinColumn(name = "tbl\_thanh\_vien\_id")  
 ThanhVien thanhVien;  
  
 @OneToMany(mappedBy = "tayDua", fetch = FetchType.*EAGER*,cascade = CascadeType.*PERSIST*)  
 List<DoiDuaTayDua> listDoiDuaTayDua;  
  
 public Integer getId() {  
 return id;  
 }  
  
 public void setId(Integer id) {  
 this.id = id;  
 }  
  
 public String getQuocTich() {  
 return quocTich;  
 }  
  
 public void setQuocTich(String quocTich) {  
 this.quocTich = quocTich;  
 }  
  
 public String getTieuSu() {  
 return tieuSu;  
 }  
  
 public void setTieuSu(String tieuSu) {  
 this.tieuSu = tieuSu;  
 }  
  
 public ThanhVien getThanhVien() {  
 return thanhVien;  
 }  
  
 public void setThanhVien(ThanhVien thanhVien) {  
 this.thanhVien = thanhVien;  
 }  
  
 public List<DoiDuaTayDua> getListDoiDuaTayDua() {  
 return listDoiDuaTayDua;  
 }  
  
 public void setListDoiDuaTayDua(List<DoiDuaTayDua> listDoiDuaTayDua) {  
 this.listDoiDuaTayDua = listDoiDuaTayDua;  
 }  
  
}

TayDuaController.java

@Controller  
@RequestMapping("bang-xep-hang-tay-dua")  
public class TayDuaController {  
 @Autowired TayDuaService tayDuaService;  
 @Autowired  
 ChangDuaService changDuaService;  
  
 @Autowired  
 DoiDuaService doiDuaService;  
  
 @Autowired  
 KetQuaService ketQuaService;  
  
 @GetMapping("/{id}")  
 public String bxhTayDua(@PathVariable("id") Integer id, HttpSession session, RedirectAttributes ra, Model model) {  
 if(session.getAttribute("username") == null) {  
 ra.addFlashAttribute("requireLogin", "Vui lòng đăng nhập để xem bảng xếp hạng tay đua!");  
 return "redirect:/auth/login";  
 }  
 Optional<ChangDua> changDua = changDuaService.getChangDuaById(id);  
 Integer doiDuaId = (Integer) session.getAttribute("doiDuaId");  
 Optional<DoiDua> doiDua = doiDuaService.getById(doiDuaId);  
 Set<DoiDuaTayDua> listDoiDuaTayDua = doiDua.get().getListDoiDuaTayDua();  
  
 List<TayDuaDto> listTayDuaDto = new ArrayList<>();  
 for(DoiDuaTayDua doiDuaTayDua: listDoiDuaTayDua) {  
 System.*out*.println(doiDuaTayDua.getId());  
 System.*out*.println(changDua.get().getId());  
 Optional<KetQua> ketQua = ketQuaService.findByDoiDuaTayDuaAndChangDua(doiDuaTayDua.getId(), changDua.get().getId());  
 System.*out*.println(ketQua);  
 listTayDuaDto.add(new TayDuaDto(doiDuaTayDua.getTayDua().getId(), doiDuaTayDua.getTayDua().getThanhVien().getHoTen(), ketQua.get().getChangDua().getTen(), ketQua.get().getThoiGianVeDich(), ketQua.get().getThuHang(), ketQua.get().getDiem()));  
 }  
 Collections.*sort*(listTayDuaDto, new Comparator<TayDuaDto>() {  
 @Override  
 public int compare(TayDuaDto o1, TayDuaDto o2) {  
 return o2.diem - o1.diem;  
 }  
 });  
 Integer namToChuc = (Integer) session.getAttribute("namToChuc");  
 model.addAttribute("namToChuc", namToChuc);  
 model.addAttribute("listTayDuaDto", listTayDuaDto);  
 model.addAttribute("changDuaId", id);  
 for(TayDuaDto tayDuaDto: listTayDuaDto) {  
 System.*out*.println(tayDuaDto);  
 }  
 return "bangxephangtaydua";  
  
 }  
  
}

TayDuaDto.java

public class TayDuaDto {  
 private Integer id;  
 String tenTayDua;  
 String tenChang;  
 Integer thoiGianVeDich;  
 Integer thuHang;  
 Integer diem;  
  
 public TayDuaDto(Integer id, String tenTayDua, String tenChang, Integer thoiGianVeDich, Integer thuHang, Integer diem) {  
 this.id = id;  
 this.tenTayDua = tenTayDua;  
 this.tenChang = tenChang;  
 this.thoiGianVeDich = thoiGianVeDich;  
 this.thuHang = thuHang;  
 this.diem = diem;  
 }  
  
 public Integer getId() {  
 return id;  
 }  
  
 public void setId(Integer id) {  
 this.id = id;  
 }  
  
 public String getTenTayDua() {  
 return tenTayDua;  
 }  
  
 public void setTenTayDua(String tenTayDua) {  
 this.tenTayDua = tenTayDua;  
 }  
  
 public String getTenChang() {  
 return tenChang;  
 }  
  
 public void setTenChang(String tenChang) {  
 this.tenChang = tenChang;  
 }  
  
 public Integer getThoiGianVeDich() {  
 return thoiGianVeDich;  
 }  
  
 public void setThoiGianVeDich(Integer thoiGianVeDich) {  
 this.thoiGianVeDich = thoiGianVeDich;  
 }  
  
 public Integer getThuHang() {  
 return thuHang;  
 }  
  
 public void setThuHang(Integer thuHang) {  
 this.thuHang = thuHang;  
 }  
  
 public Integer getDiem() {  
 return diem;  
 }  
  
 public void setDiem(Integer diem) {  
 this.diem = diem;  
 }  
  
}

TayDuaRepository.java

public interface TayDuaRepository extends CrudRepository<TayDua, Integer> {  
  
 public Optional<TayDua> findById(Integer id);  
}

TayDuaService.java

@Service  
public class TayDuaService {  
 @Autowired TayDuaRepository tayDuaRepository;  
  
 public Optional<TayDua> getTayDua(Integer id) {  
 return tayDuaRepository.findById(id);  
 }  
  
}

ThanhVien.java

@Entity  
@Table(name = "tblthanhvien")  
public class ThanhVien {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Integer id;  
  
 @Column(name = "username", nullable = false)  
 String username;  
  
 @Column(name = "password", nullable = false)  
 String password;  
  
 @Column(name = "ngay\_sinh", nullable = false)  
 Date ngaySinh;  
  
 @Column(name = "email", nullable = false)  
 String email;  
  
 @Column(name = "sdt")  
 String sdt;  
  
 @Column(name = "ghi\_chu")  
 String ghiChu;  
  
 @Column(name = "ho\_ten")  
 String hoTen;  
  
 @Column(name = "dia\_chi")  
 String diaChi;  
  
 @OneToOne(mappedBy = "thanhVien", cascade = CascadeType.*ALL*)  
 @PrimaryKeyJoinColumn  
 private TayDua tayDua;  
  
 @OneToOne(mappedBy = "thanhVien", cascade = CascadeType.*ALL*)  
 @PrimaryKeyJoinColumn  
 private BanToChuc banToChuc;  
  
 public Integer getId() {  
 return id;  
 }  
  
 public void setId(Integer id) {  
 this.id = id;  
 }  
  
 public String getUsername() {  
 return username;  
 }  
  
 public void setUsername(String username) {  
 this.username = username;  
 }  
  
 public String getPassword() {  
 return password;  
 }  
  
 public void setPassword(String password) {  
 this.password = password;  
 }  
  
 public Date getNgaySinh() {  
 return ngaySinh;  
 }  
  
 public void setNgaySinh(Date ngaySinh) {  
 this.ngaySinh = ngaySinh;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public String getSdt() {  
 return sdt;  
 }  
  
 public void setSdt(String sdt) {  
 this.sdt = sdt;  
 }  
  
 public String getGhiChu() {  
 return ghiChu;  
 }  
  
 public void setGhiChu(String ghiChu) {  
 this.ghiChu = ghiChu;  
 }  
  
 public String getHoTen() {  
 return hoTen;  
 }  
  
 public void setHoTen(String hoTen) {  
 this.hoTen = hoTen;  
 }  
  
 public String getDiaChi() {  
 return diaChi;  
 }  
  
 public void setDiaChi(String diaChi) {  
 this.diaChi = diaChi;  
 }  
  
 public TayDua getTayDua() {  
 return tayDua;  
 }  
  
 public void setTayDua(TayDua tayDua) {  
 this.tayDua = tayDua;  
 }  
  
 public BanToChuc getBanToChuc() {  
 return banToChuc;  
 }  
  
 public void setBanToChuc(BanToChuc banToChuc) {  
 this.banToChuc = banToChuc;  
 }  
  
}

ThanhVienRepository.java

@Repository  
public interface ThanhVienRepository extends CrudRepository<ThanhVien, Integer> {  
 public Optional<ThanhVien> findByUsername(String username);  
}

ThanhVienService.java

public class ThanhVienService{  
 @Autowired private ThanhVienRepository thanhVienRepo;  
  
}

MainController.java

@Controller  
public class MainController {  
 @Autowired private GiaiDauService giaiDauService;  
  
 @GetMapping("")  
 public String homePage(HttpSession session, RedirectAttributes ra) {  
 if(session.getAttribute("username") == null) {  
 ra.addFlashAttribute("requireLogin", "Vui lòng đăng nhập để xem bảng xếp hạng!");  
 return "redirect:/auth/login";  
 } else {  
 return "index";  
 }  
 }  
  
 @GetMapping("/chon-thong-ke")  
 public String bangXepHangPage(HttpSession session, RedirectAttributes ra, Model model) {  
 if(session.getAttribute("username") == null) {  
 ra.addFlashAttribute("requireLogin", "Vui lòng đăng nhập để xem bảng xếp hạng!");  
 return "redirect:/auth/login";  
 } else {  
 Iterable<GiaiDau> listGiaiDau = giaiDauService.getAll();  
 model.addAttribute("listGiaiDau", listGiaiDau);  
 return "chonthongke";  
 }  
 }  
}

Các page html dùng template engine của spring boot là thymeleaf:

Login.html

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Đăng nhập</title>  
 <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap.min.css}"/>  
</head>  
<body>  
<div class="container-fluid text-center">  
 <div><h2>Đăng nhập</h2></div>  
 <div th:if="${requireLogin}" class="alert alert-danger">  
 [[${requireLogin}]]  
 </div>  
 <form th:action="@{/auth/login}" th:object="${thanhVien}" method="post" style="max-width: 500px; margin: 0 auto">  
 <div class="border border-secondary rounded p-3">  
 <div class="form-group row">  
 <label class="col-sm-4 col-form-label">  
 Tên đăng nhập:  
 </label>  
 <div class="col-sm-8">  
 <input type="text" th:field="\*{username}" class="form-control" required />  
 </div>  
 </div>  
 <div class="form-group row">  
 <label class="col-sm-4 col-form-label">  
 Mật khẩu:  
 </label>  
 <div class="col-sm-8">  
 <input type="password" th:field="\*{password}" class="form-control" required/>  
 </div>  
 </div>  
 <div th:if="${message}" style="color: #f00" class="mt-2">  
 [[${message}]]  
 </div>  
 <div class="text-center">  
 <button type="submit" class="btn btn-primary">  
 Đăng nhập  
 </button>  
 </div>  
 </div>  
 </form>  
</div>  
</body>  
</html>

Index.html

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Home page</title>  
 <!-- Import bootstrap -->  
 <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap.min.css}"/>  
</head>  
<body>  
 <div class="container-fluid text-center">  
 <div class="h1">Module bang xep hang doi dua</div>  
 <a class="h2" th:href="@{/chon-thong-ke}">Xem thống kê(bảng xếp hạng)</a>  
 </div>  
</body>  
</html>

Chonthongke.html

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Xem bảng xếp hạng</title>  
 <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap.min.css}"/>  
 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>  
 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>  
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>  
</head>  
<body>  
 <div class="container-fluid">  
 <div class="border border-secondary rounded p-3" style="width: 21%">  
 <div class="h2">Xem bảng xếp hạng</div>  
 <div>  
 <div class="form-group row">  
 <label class="col-4">  
 Loại thống kê  
 </label>  
 <div class="col-1">  
 <div class="dropdown">  
 <button class="btn btn-secondary dropdown-toggle" type="button" id="dropdownMenu2" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">  
 Chọn  
 </button>  
 <div class="dropdown-menu" aria-labelledby="dropdownMenu2">  
 <button class="dropdown-item" type="button" id="bxhDoiDua" onclick="xemBXHDoiDua()">Xem bảng xếp hạng đội đua</button>  
<!-- <button class="dropdown-item" type="button" id="bxhTayDua" onclick="xemBXHTayDua()">Xem bảng xếp hạng tay đua</button>-->  
 </div>  
 </div>  
 </div>  
 </div>  
 <div class="form-group row">  
 <label class="col-4">  
 Giải đấu  
 </label>  
 <div class="col-1">  
 <div class="dropdown">  
 <button class="btn btn-secondary dropdown-toggle" type="button" id="dropdownMenu3" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">  
 Chọn  
 </button>  
 <div class="dropdown-menu" aria-labelledby="dropdownMenu3">  
 <button th:each="giaidau: ${listGiaiDau}" class="dropdown-item" th:attr="onclick=|handleGiaiDau('${giaidau.namToChuc}')|" type="button">[[${giaidau.namToChuc}]]</button>  
 </div>  
 </div>  
 </div>  
 </div>  
 <div class="text-danger mb-2 d-none" id="msg-error">  
 Xin chọn đủ các thông tin trên!  
 </div>  
 <button class="btn btn-primary" onclick="handleXemBXH()">Xem</button>  
 </div>  
 </div>  
 </div>  
  
  
</body>  
<script>  
 let is\_choose = {  
 loaiTK: false,  
 giaiDau: false  
 }  
 function xemBXHDoiDua() {  
 let toggle = document.getElementById("dropdownMenu2");  
 toggle.innerText = "Xem bảng xếp hạng đội đua";  
 is\_choose.loaiTK = true;  
 }  
 // function xemBXHTayDua() {  
 // let toggle = document.getElementById("dropdownMenu2");  
 // toggle.innerText = "Xem bảng xếp hạng tay đua";  
 // }  
 function handleGiaiDau(namToChuc) {  
 let toggle = document.getElementById("dropdownMenu3");  
 toggle.innerText = namToChuc;  
 is\_choose.giaiDau = true;  
 }  
  
 function handleXemBXH() {  
 if(!is\_choose.loaiTK || !is\_choose.giaiDau) {  
 document.getElementById("msg-error").classList.add("d-block");  
 } else {  
 document.getElementById("msg-error").classList.remove("d-block");  
 let data = {};  
 data.loaiTK = document.getElementById("dropdownMenu2").innerText;  
 data.giaiDau = Number(document.getElementById("dropdownMenu3").innerText);  
 console.log(data)  
 fetch("http://localhost:8080/bang-xep-hang-doi-dua/", {  
 method: 'POST',  
 headers: {  
 'Content-Type': 'application/json',  
 },  
 body: JSON.stringify(data)  
 })  
 .then(data => {  
 console.log("data", data);  
 return window.location.href = "/bang-xep-hang-doi-dua";  
 })  
 .catch(e => {  
 console.log("e", e);  
 })  
 }  
 }  
</script>  
</html>

Bangxemhangtaydua.html

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Bang xep hang tay dua</title>  
 <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap.min.css}"/>  
</head>  
<body>  
 <div class="container-fluid text-center">  
 <div class="mb-3">  
 <h1>BXH tay đua</h1>  
 <h2>Mùa giải: [[${namToChuc}]]</h2>  
 </div>  
 <div>  
 <table class="table table-bordered table-striped">  
 <thead>  
 <tr>  
 <th>TT</th>  
 <th>Mã</th>  
 <th>Tên tay đua</th>  
 <th>Tên chặng</th>  
 <th>thời gian về đích</th>  
 <th>thứ hạng</th>  
 <th>điểm</th>  
 </tr>  
  
 </thead>  
 <tbody>  
 <th:block th:each="tayDua, iter: ${listTayDuaDto}">  
 <tr>  
 <td>[[${iter.index}]]</td>  
 <td>[[${tayDua.id}]]</td>  
 <td>[[${tayDua.tenTayDua}]]</td>  
 <td>[[${tayDua.tenChang}]]</td>  
 <td>[[${tayDua.thoiGianVeDich}]]</td>  
 <td>[[${tayDua.thuHang}]]</td>  
 <td>[[${tayDua.diem}]]</td>  
 </tr>  
 </th:block>  
 </tbody>  
 </table>  
 <a class="btn btn-primary" th:href="@{'/bang-xep-hang-chang-dua/' + ${changDuaId}}" role="button" style="position: absolute; left: 15px" >Quay lại</a>  
 </div>  
 </div>  
</body>  
</html>

Bangxemhangdoidua.html

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap.min.css}"/>  
 <title>BXH Đội đua</title>  
</head>  
<body>  
 <div class="container-fluid text-center">  
 <div class="mb-3">  
 <h1>BXH Đội đua</h1>  
 <h2>Mùa giải: [[${namToChuc}]]</h2>  
 </div>  
 <div>  
 <table class="table table-bordered table-striped">  
 <thead>  
 <tr>  
 <th>TT</th>  
 <th>Mã đội</th>  
 <th>Tên đội</th>  
 <th>Hãng</th>  
 <th>Tổng điểm</th>  
 <th>Tổng thời gian(phút)</th>  
 <th></th>  
 </tr>  
 </thead>  
 <tbody>  
 <th:block th:each="doiDua, iter: ${listDoiDuaDto}">  
 <tr>  
 <td>[[${iter.index}]]</td>  
 <td>[[${doiDua.id}]]</td>  
 <td>[[${doiDua.ten}]]</td>  
 <td>[[${doiDua.hang}]]</td>  
 <td>[[${doiDua.tongDiem}]]</td>  
 <td>[[${doiDua.tongThoiGian}]]</td>  
 <td>  
 <a th:href="@{'/bang-xep-hang-chang-dua/' + ${doiDua.id}}">Chọn</a>  
 </td>  
 </tr>  
 </th:block>  
 </tbody>  
 </table>  
 <a class="btn btn-primary" href="/chon-thong-ke" role="button" style="position: absolute; left: 15px">Quay lại</a>  
 </div>  
  
 </div>  
  
</body>  
</html>

Bangxemhangchangdua.html

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap.min.css}"/>  
 <title>BXH Chang dua</title>  
</head>  
<body>  
<div class="container-fluid text-center">  
 <div class="mb-3">  
 <h1>BXH chặng đua</h1>  
 <h2>Mùa giải: [[${namToChuc}]]</h2>  
 </div>  
 <div>  
 <table class="table table-bordered table-striped">  
 <thead>  
 <tr>  
 <th>TT</th>  
 <th>Mã chặng</th>  
 <th>Tên chặng</th>  
 <th>Tổng điểm</th>  
 <th>Tổng thời gian(phút)</th>  
 <th></th>  
 </tr>  
 </thead>  
 <tbody>  
 <th:block th:each="changDua, iter: ${listChangDuaDto}">  
 <tr>  
 <td>[[${iter.index}]]</td>  
 <td>[[${changDua.id}]]</td>  
 <td>[[${changDua.ten}]]</td>  
 <td>[[${changDua.tongDiem}]]</td>  
 <td>[[${changDua.tongThoiGian}]]</td>  
 <td>  
 <a th:href="@{'/bang-xep-hang-tay-dua/' + ${changDua.id}}">Chọn</a>  
 </td>  
 </tr>  
 </th:block>  
 </tbody>  
 </table>  
 <a class="btn btn-primary" href="/bang-xep-hang-doi-dua/" role="button" style="position: absolute; left: 15px">Quay lại</a>  
 </div>  
  
</div>  
</body>  
</html>

Sql:

Tblbantochuc.sql

create table tblbantochuc  
(  
 tbl\_thanh\_vien\_id int not null  
 primary key,  
 mo\_ta varchar(255) not null,  
 ten varchar(255) not null,  
 constraint FKfrm84jfovm7rj9cp78d3yed2h  
 foreign key (tbl\_thanh\_vien\_id) references tblthanhvien (id)  
);

tblchangdua.sql

create table tblchangdua  
(  
 id int auto\_increment  
 primary key,  
 dia\_diem varchar(255) null,  
 mo\_ta varchar(255) not null,  
 so\_vong\_dua int not null,  
 ten varchar(255) not null,  
 thoi\_gian datetime(6) not null,  
 tbl\_giai\_dau\_id int null,  
 constraint FKv4xudce6c37j3shlma3yy1cy  
 foreign key (tbl\_giai\_dau\_id) references tblgiaidau (id)  
);

tbldoidua.sql

create table tbldoidua  
(  
 id int auto\_increment  
 primary key,  
 hang varchar(255) not null,  
 mo\_ta varchar(255) not null,  
 ten varchar(255) not null,  
 tbl\_giai\_dau\_id int null,  
 constraint FKl1m55xaqeoiqnf2xvpkoiqkcr  
 foreign key (tbl\_giai\_dau\_id) references tblgiaidau (id)  
);

tbldoidua.sql

create table tbldoiduataydua  
(  
 id int auto\_increment  
 primary key,  
 thoi\_diem datetime(6) not null,  
 tbl\_doi\_dua\_id int null,  
 tbl\_thanh\_vien\_id int null,  
 constraint FK5ady64wam6bhyd0jnta2v90qk  
 foreign key (tbl\_doi\_dua\_id) references tbldoidua (id),  
 constraint FKmqyljmewidyo7hxsk2hvgrm2h  
 foreign key (tbl\_thanh\_vien\_id) references tbltaydua (tbl\_thanh\_vien\_id)  
);

tblgiaidau.sql

create table tblgiaidau  
(  
 id int auto\_increment  
 primary key,  
 nam\_to\_chuc int null  
);

tblketqua.sql

create table tblketqua  
(  
 id int auto\_increment  
 primary key,  
 diem int not null,  
 thoi\_gian\_ve\_dich int not null,  
 thu\_hang int not null,  
 tbl\_chang\_dua\_id int null,  
 tbl\_doi\_dua\_tay\_dua\_id int null,  
 constraint FKgnpkiin4f38iynmm1isp5f7dr  
 foreign key (tbl\_chang\_dua\_id) references tblchangdua (id),  
 constraint FKlr7vpfoe5gqu83g1eb6x9dpc6  
 foreign key (tbl\_doi\_dua\_tay\_dua\_id) references tbldoiduataydua (id)  
);

tbltaydua.sql

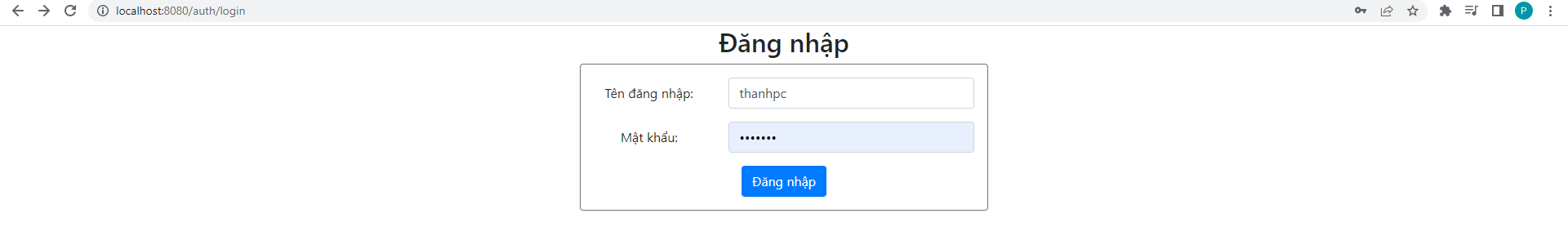
create table tbltaydua  
(  
 tbl\_thanh\_vien\_id int not null  
 primary key,  
 quoc\_tich varchar(255) not null,  
 tieu\_su varchar(255) not null,  
 constraint FKtq4h7qd16g0jh5rwd80dd306u  
 foreign key (tbl\_thanh\_vien\_id) references tblthanhvien (id)  
);

tblthanhvien.sql

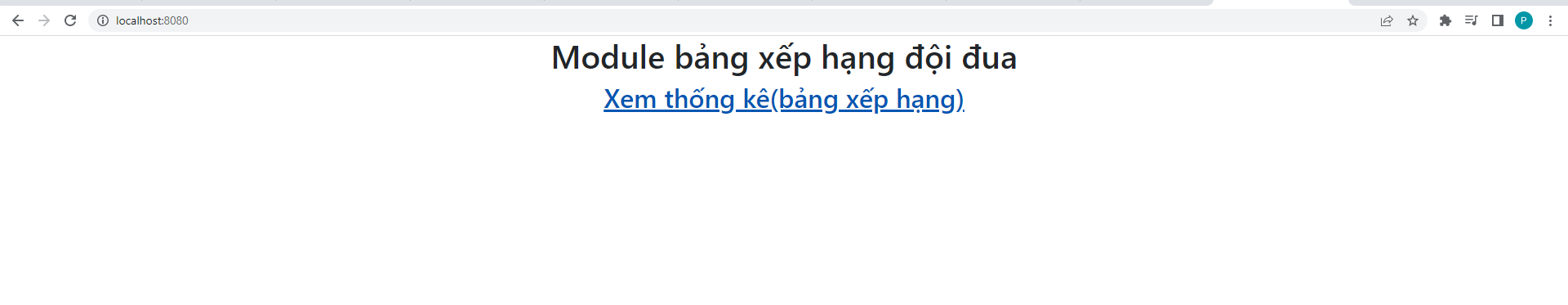
create table tblthanhvien  
(  
 id int auto\_increment  
 primary key,  
 dia\_chi varchar(255) null,  
 email varchar(255) not null,  
 ghi\_chu varchar(255) null,  
 ho\_ten varchar(255) null,  
 ngay\_sinh datetime(6) not null,  
 password varchar(255) not null,  
 sdt varchar(255) null,  
 username varchar(255) not null  
);

* **Kết quả cài đặt(happy case)**

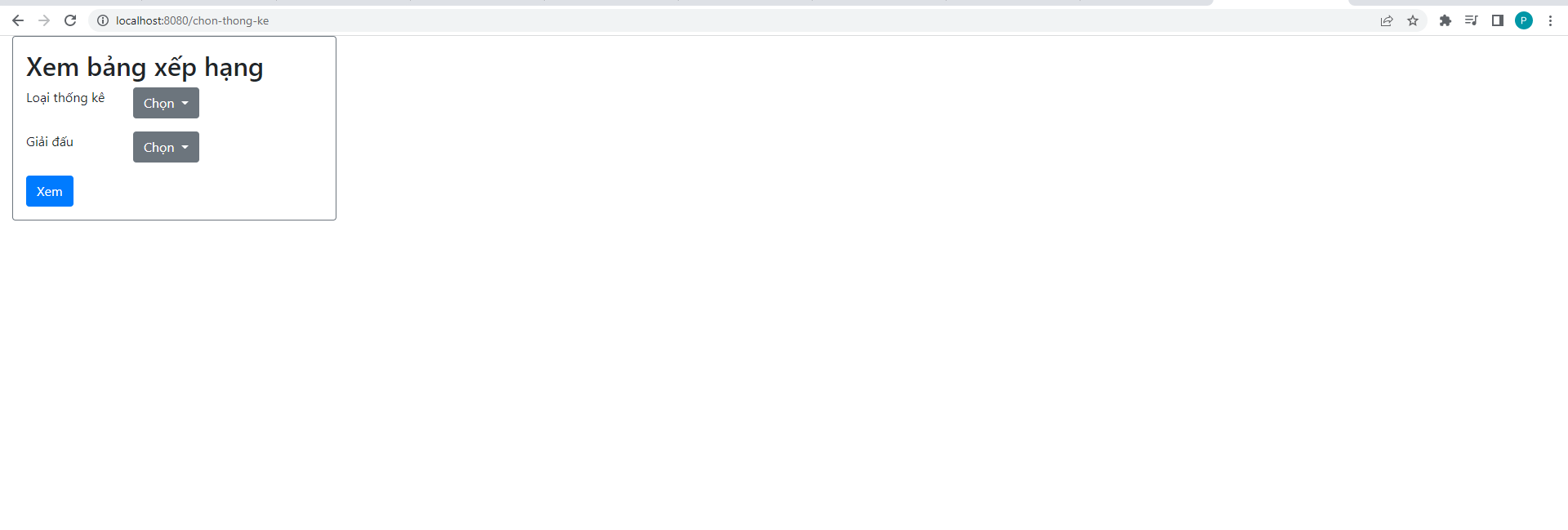
Trang đăng nhập:

* + ****

Đăng nhập thành công vào giao diện chính



Ấn vào link xem thống kê(bảng xếp hạng):



Chọn xem bảng xếp hạng đội đua và giải đấu năm 2022(hiện tại)



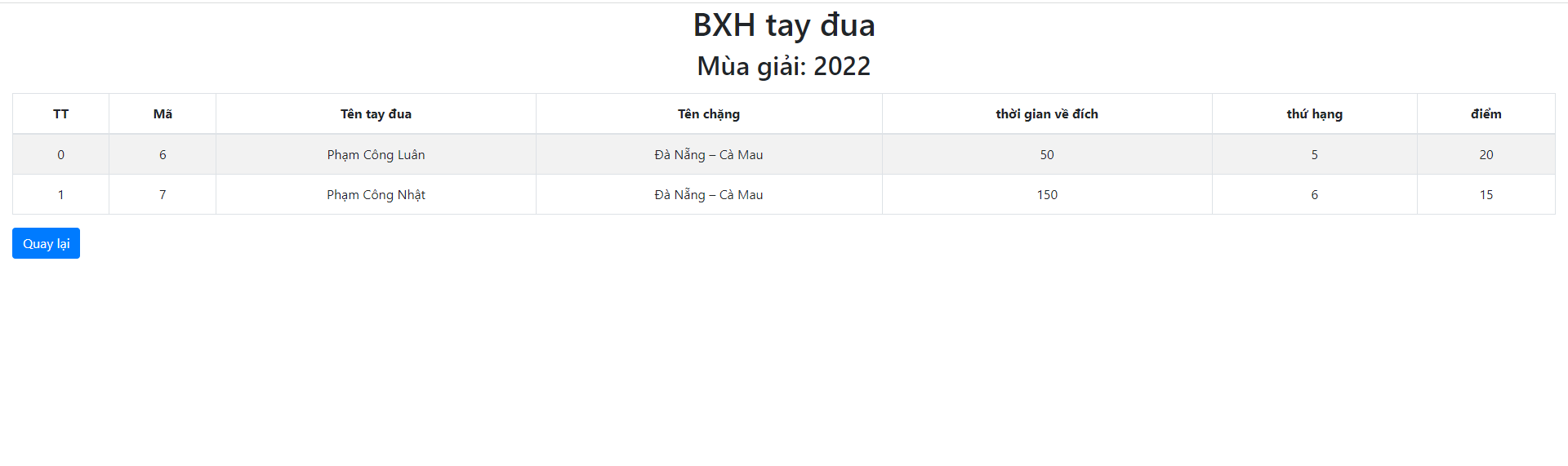
Danh sách các đội đua mùa giải 2022:



Ấn chọn 1 đội đua sẽ hiện ra danh sách các chặng đua mà đội đua đó tham gia:

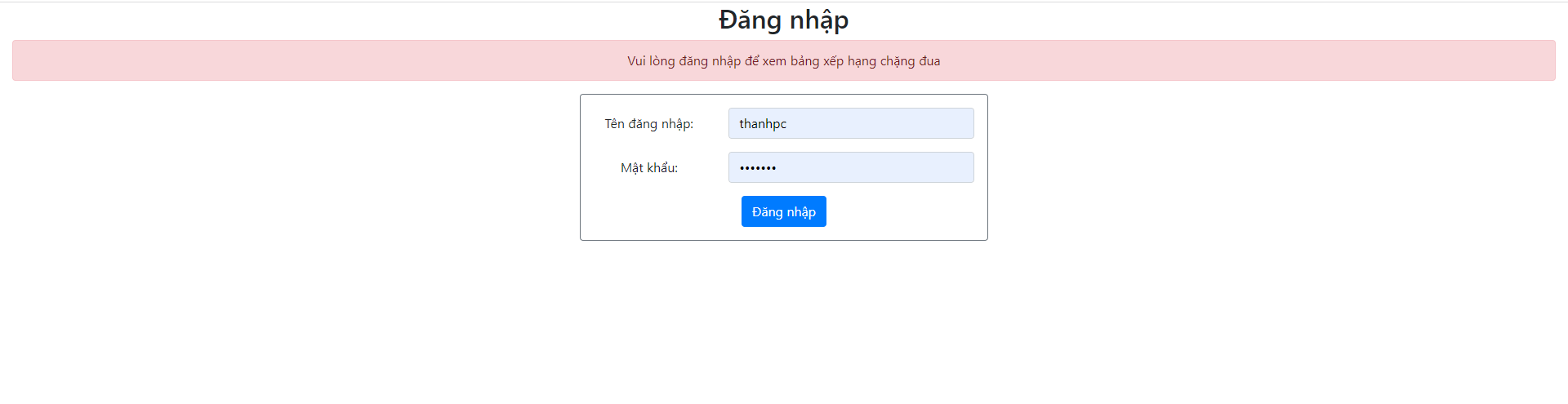


Ấn vào 1 chặng đua sẽ hiện ra danh sách 2 tay đua của đội đua đã tham gia chặng đua đó:

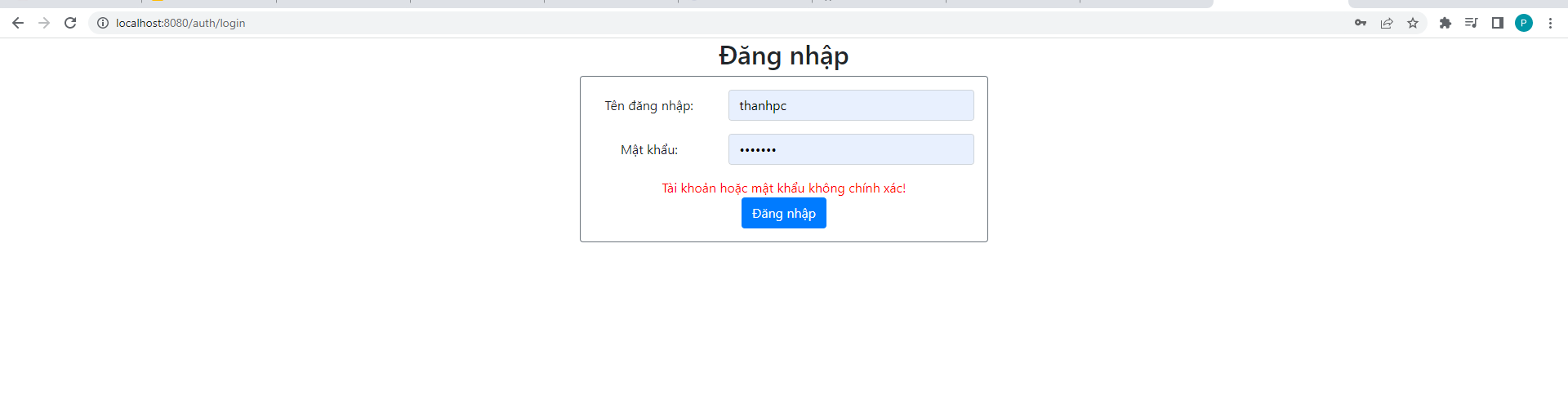


Xử lý lỗi: (handle exception)

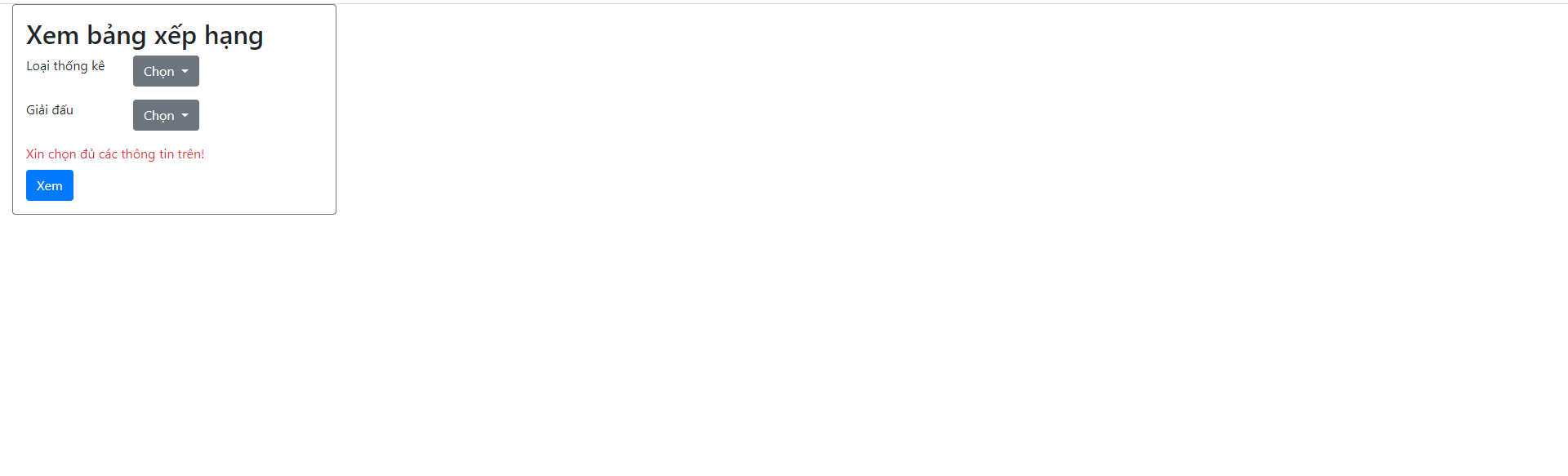
Nếu chưa đăng nhập mà xem một thống kê bất kì thì sẽ hiển thị lỗi:



Nếu đăng nhập sai tài khoản hoặc mật khẩu:



Nếu chọn thiếu 1 thông tin khi ấn xem bảng xếp hạng(loại thống kê hoặc năm tổ chức)



* 1. **Viết unit test cho các service:**

|  |  |  |
| --- | --- | --- |
| Lớp điều khiển | Phương thức | Trường hợp test |
| AuthService | authLogin(String username, String password) | Login với tài khoản mật khẩu chính xác |
| Login với tài khoản mật khẩu không chính xác |
| ChangDuaService | getChangDuaById(Integer id) | Không tìm thấy chặng đua với id truyền vào |
| Tìm thấy chặng đua với id truyền vào |
| DoiDuaService | getDoiDuaById(Integer id) | Không tìm thấy đội đua với id truyền vào |
| Tìm thấy đội đua với id truyền vào |
| GiaiDauService | getAll() | Lấy được danh sách giải đấu |
| getByNamToChuc(Integer namToChuc) | Không tìm thấy năm tổ chức với giá trị namToChuc truyền vào |
| Tìm thấy năm tổ chức với giá trị namToChuc truyền vào |
| KetQuaService | sumDiem(Integer doiDuaTayDua) | Lấy được tổng số điểm của 2 tay đua của 1 đội đua tham gia 1 chặng đua |
| sumThoiGian(Integer doiDuaTayDua) | Lấy được tổng thời gian của 2 tay đua của 1 đội đua tham gia 1 chặng đua |
| getDiemByDoiDuaTayDuaAndChangDua(Integer doiDuaTayDua, Integer changDua) | Lấy thành công điểm của một tay đua |
| Lấy thất bại điểm của một tay đua |
| getThoiGianByDoiDuaTayDuaAndChangDua(Integer doiDuaTayDua, Integer changDua) | Lấy thành công thời gian của một tay đua |
| Lấy thất bại thời gian của một tay đua |
| findByDoiDuaTayDuaAndChangDua(Integer ddtd, Integer changDua) | Lấy thành công thông tin kết quả của một tay đua |
| Lấy thất bại thông tin kết quả của một tay đua |
|  |  |
|  |  |  |

AuthServiceTests.java

public class AuthServiceTests {  
 @Autowired  
 AuthService authService;  
 @Test  
 void testLoginSuccessShouldReturnTrue() {  
 String username = "thanhpc";  
 String password = "thanhpc";  
 boolean isLogin = authService.authLogin(username, password);  
 Assertions.*assertEquals*(isLogin, true);  
 }  
 @Test  
 void testLoginFailureShouldReturnFalse() {  
 String username = "thanhpcx";  
 String password = "thanhpc";  
 boolean isLogin = authService.authLogin(username, password);  
 Assertions.*assertEquals*(isLogin, false);  
 }  
}

ChangDuaServiceTests.java

@SpringBootTest  
public class ChangDuaServiceTests {  
 @Autowired  
 ChangDuaService changDuaService;  
  
 @Test  
 void getChangDuaByIdShouldSuccess() {  
 Optional<ChangDua> changDua = changDuaService.getChangDuaById(1);  
 System.*out*.println(changDua);  
 Assertions.*assertTrue*(changDua.get().getTen().equals("Hà Giang – Hà Tĩnh"));  
 }  
 @Test  
 void getChangDuaByIdShouldFailure() {  
 Optional<ChangDua> changDua = changDuaService.getChangDuaById(122);  
 Assertions.*assertTrue*(changDua.isPresent() == false);  
 }  
}

DoiDuaServiceTests.java

@SpringBootTest  
public class DoiDuaServiceTests {  
 @Autowired  
 DoiDuaService doiDuaService;  
  
 @Test  
 void getdoiDuaByIdShouldSuccess() {  
 Optional<DoiDua> doiDua = doiDuaService.getDoiDuaById(1);  
 System.*out*.println(doiDua);  
 Assertions.*assertTrue*(doiDua.get().getTen().equals("Honda-hd1"));  
 }  
 @Test  
 void getdoiDuaByIdShouldFailure() {  
 Optional<DoiDua> doiDua = doiDuaService.getDoiDuaById(122);  
 Assertions.*assertTrue*(doiDua.isPresent() == false);  
 }  
  
}

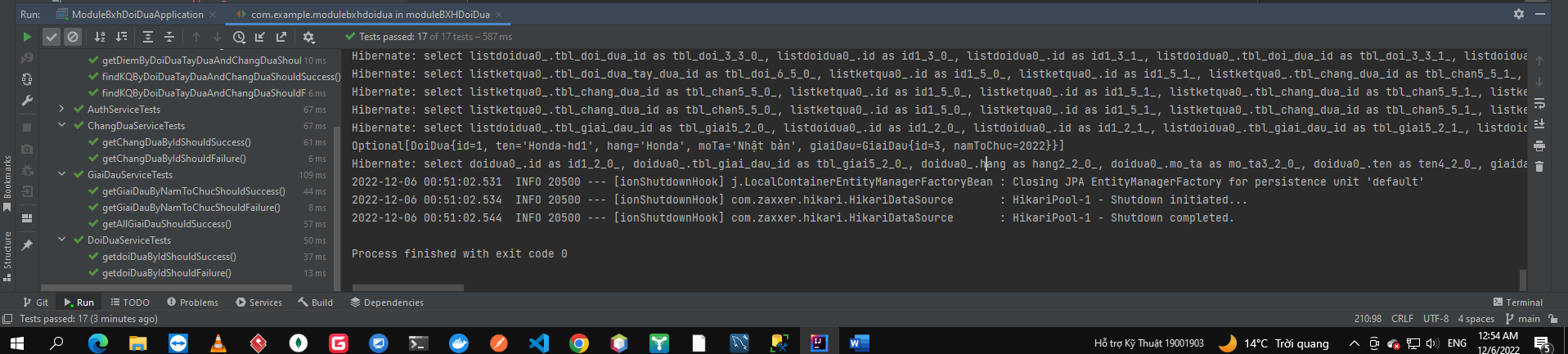
GiaiDauServiceTests.java

@SpringBootTest  
public class GiaiDauServiceTests {  
 @Autowired  
 GiaiDauService giaiDauService;  
  
 @Test  
 void getAllGiaiDauShouldSuccess() {  
 Iterable<GiaiDau> listGiaiDau = giaiDauService.getAll();  
 Assertions.*assertNotNull*(listGiaiDau);  
 for(GiaiDau giaiDau: listGiaiDau) {  
 System.*out*.println(giaiDau);  
 }  
 }  
  
 @Test  
 void getGiaiDauByNamToChucShouldSuccess() {  
 Optional<GiaiDau> giaiDau = giaiDauService.getByNamToChuc(2022);  
 Assertions.*assertTrue*(giaiDau.isPresent());  
 }  
 @Test  
 void getGiaiDauByNamToChucShouldFailure() {  
 Optional<GiaiDau> giaiDau = giaiDauService.getByNamToChuc(2023);  
 Assertions.*assertFalse*(giaiDau.isPresent());  
 }  
}

KetQuaServiceTests.java

@SpringBootTest  
public class KetQuaServiceTests {  
 @Autowired  
 KetQuaService ketQuaService;  
  
 @Test  
 void sumDiemShouldReturnIntegerNumberGreaterThanZero() {  
 Integer ddtdId = 1;  
 Integer sumDiem = ketQuaService.sumDiem(ddtdId);  
 System.*out*.println(sumDiem);  
 Assertions.*assertTrue*(sumDiem > 0);  
 }  
  
 @Test  
 void sumThoiGianShouldReturnIntegerNumberGreaterThanZero() {  
 Integer ddtdId = 1;  
 Integer sumThoiGian = ketQuaService.sumThoiGian(ddtdId);  
 System.*out*.println(sumThoiGian);  
 Assertions.*assertTrue*(sumThoiGian > 0);  
 }  
  
 @Test  
 void getDiemByDoiDuaTayDuaAndChangDuaShouldSuccess() {  
 Integer ddtd = 1;  
 Integer changDua = 1;  
 Integer diem = ketQuaService.getDiemByDoiDuaTayDuaAndChangDua(ddtd, changDua);  
 System.*out*.println(diem);  
 Assertions.*assertTrue*(diem > 0);  
 }  
  
 @Test  
 void getDiemByDoiDuaTayDuaAndChangDuaShouldFailure() {  
 Integer ddtd = 10;  
 Integer changDua = 122;  
 Integer diem = ketQuaService.getDiemByDoiDuaTayDuaAndChangDua(ddtd, changDua);  
 Assertions.*assertTrue*(diem == null);  
 }  
  
 @Test  
 void getThoiGianByDoiDuaTayDuaAndChangDuaShouldSuccess() {  
 Integer ddtd = 1;  
 Integer changDua = 1;  
 Integer tg = ketQuaService.getThoiGianByDoiDuaTayDuaAndChangDua(ddtd, changDua);  
 System.*out*.println(tg);  
 Assertions.*assertTrue*(tg > 0);  
 }  
  
 @Test  
 void getThoiGianByDoiDuaTayDuaAndChangDuaShouldFailure() {  
 Integer ddtd = 10;  
 Integer changDua = 122;  
 Integer tg = ketQuaService.getThoiGianByDoiDuaTayDuaAndChangDua(ddtd, changDua);  
 Assertions.*assertTrue*(tg == null);  
 }  
  
 @Test  
 void findKQByDoiDuaTayDuaAndChangDuaShouldSuccess() {  
 Integer ddtd = 1;  
 Integer changDua = 1;  
 Optional<KetQua> ketQua = ketQuaService.findByDoiDuaTayDuaAndChangDua(ddtd, changDua);  
 System.*out*.println(ketQua);  
 Assertions.*assertTrue*(ketQua.isPresent());  
 }  
  
 @Test  
 void findKQByDoiDuaTayDuaAndChangDuaShouldFailure() {  
 Integer ddtd = 10;  
 Integer changDua = 122;  
 Optional<KetQua> ketQua = ketQuaService.findByDoiDuaTayDuaAndChangDua(ddtd, changDua);  
 Assertions.*assertFalse*(ketQua.isPresent());  
 }  
}

Chạy tất cả xanh 17 test case:



1. **Rà soát và kiểm thử hệ thống**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã trường hợp kiểm thử** | **Mục đích kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| BXH\_1 | **Chức năng: Ban tổ chức xem bảng xếp hạng đội đua** | | |
| **Pre-condition: Ban tổ chức đăng nhập thành công vào hệ thống** | | |
| **Giao diện (Phần này viết các trường hợp kiểm thử cho giao diện chung và các giao diện cho các control)** | | |
| **Giao diện chung** | | |
| **Giao diện trang chủ** | | |
| Kiểm tra tổng  thể giao diện  màn hình | Kiểm tra về bố cục, font chữ,  chính tả, màu chữ | Các label, textbox cùng font chữ cỡ chữ, căn lề trái, có độ dài, rộng và khoảng cách bằng nhau, không xô lệch.  - Không có lỗi về chính tả, cấu trúc câu, ngữ pháp trên màn hình  - Form được bố trí hợp lý và dễ sử dụng |
| BXH\_2 | Kiểm tra bố cục giao diện | 1. Kiểm tra title của màn hình  2. Kiểm tra hiển thị thông tin button trên màn hình | 1. Hiển thị title của chức năng  2. Button  + Xem thống kê |
| BXH\_3 | Kiểm tra giao  diện khi thu  nhỏ, phóng to | 1. Nhấn phím Ctrl –  2. Nhấn phim Ctrl + | Màn hình thu nhỏ, phóng to tương ứng và không bị vỡ giao diện |
| BXH\_4 | Kiểm tra thứ tự  di chuyển trỏ  trên màn hình  khi nhấn phím  Tab | Forcus vào màn hình. Nhấn Tab liên tục | Con trỏ di chuyển từ text field sang nút bấm |
| BXH\_5 | Kiểm tra thứ tự  con trỏ di  chuyển ngược  lại trên màn  hình khi nhấn  Shift-Tab | Forcus vào màn hình. Nhấn  phím Shift-Tab liên tục | Con trỏ di chuyển lần lượt theo thứ tự: từ dưới lên trên, từ phải qua trái. |
| BXH\_6 | Kiểm tra thực  hiện chức năng  chính của màn  hình khi nhấn  Enter | Nhấn phím Enter | 1. Nếu chuộc ko focus vào  button nào thì Thực hiện chức năng của button chính  2. Nếu đang focus vào 1 button thì sẽ thực hiện chức năng của button |
|  | **Giao diện chọn thống kê** | | |
| BXH\_7 | Kiểm tra tổng  thể giao diện  màn hình | Kiểm tra về bố cục, font chữ,  chính tả, màu chữ | Các label, textbox cùng font chữ cỡ chữ, căn lề trái, có độ dài, rộng và khoảng cách bằng nhau, không xô lệch.  - Không có lỗi về chính tả, cấu trúc câu, ngữ pháp trên màn hình  - Form được bố trí hợp lý và dễ sử dụng |
| BXH\_8 | Kiểm tra bố cục  giao diện | 1. Kiểm tra title của màn hình  2. Kiểm tra focus của chuột  3. Kiểm tra hiển thị thông tin  các trường và button trên màn  hình | 1. Hiển thị title của chức năng  2. Focus được set vào trường đầu tiên có thể edit  3. Hiển thị đầy đủ các trường :  + Chọn Thống kê: Dropdown  list: TK Bảng xếp hạng đội đua  + Chọn giải đấu: Dropdownlist  4. Button  + Xem |
| BXH\_9 | Kiểm tra giao  diện khi thu  nhỏ, phóng to | 1. Nhấn phím Ctrl -  2. Nhấn phim Ctrl + | Màn hình thu nhỏ, phóng to tương ứng và không bị vỡ giao diện |
| BXH\_10 | Kiểm tra thứ tự  di chuyển trỏ  trên màn hình  khi nhấn phím  Tab | Forcus vào màn hình. Nhấn  Tab liên tục | Con trỏ di chuyển lần lượt theo thứ tự: Từ phải qua trái, từ trên xuống dưới. |
| BXH\_11 | Kiểm tra thứ tự  con trỏ di  chuyển ngược  lại trên màn  hình khi nhấn  Shift-Tab | Forcus vào màn hình. Nhấn  phím Shift-Tab liên tục | Con trỏ di chuyển lần lượt theo thứ tự: từ dưới lên trên, từ phải qua trái. |
| BXH\_12 | Kiểm tra thực  hiện chức năng  chính của màn  hình khi nhấn  Enter | Nhấn phím Enter | 1. Nếu chuộc ko focus vào  button nào thì Thực hiện chức  năng của button chính  2. Nếu đang focus vào 1 button thì sẽ thực hiện chức năng của button |
|  | **Giao diện xem bảng xếp hạng đội đua** | | |
| BXH\_13 | Kiểm tra tổng  thể giao diện  màn hình | Kiểm tra về bố cục, font chữ,  chính tả, màu chữ | Các label, textbox cùng font chữ cỡ chữ, căn lề trái, có độ dài, rộng và khoảng cách bằng nhau, không xô lệch.  - Không có lỗi về chính tả, cấu trúc câu, ngữ pháp trên màn hình  - Form được bố trí hợp lý và dễ sử dụng |
| BXH\_14 | Kiểm tra bố cục  giao diện | 1. Kiểm tra title của màn hình  2. Kiểm tra focus của chuột  3. Kiểm tra hiển thị thông tin  các trường và button trên màn  hình | 1. Hiển thị title của chức năng  2. Focus được set vào trường  đầu tiên có thể edit:  3. Hiển thị đầy đủ các trường :  + Mùa giải: Textview  + Table: TT, mã đội, tên đội, hãng, tổng điểm, tổng thời gian(phút)  4. Button  + Quay lại  5. Hyperlink  + Chọn 🡪 ở mỗi dòng của table |
| BXH\_15 | Kiểm tra giao  diện khi thu  nhỏ, phóng to | 1. Nhấn phím Ctrl -  2. Nhấn phim Ctrl + | Màn hình thu nhỏ, phóng to tương ứng và không bị vỡ giao diện |
| BXH\_16 | Kiểm tra thứ tự  di chuyển trỏ  trên màn hình  khi nhấn phím  Tab | Forcus vào màn hình. Nhấn  Tab liên tục | Con trỏ di chuyển lần lượt theo thứ tự: Từ phải qua trái, từ trên xuống dưới. |
| BXH\_17 | Kiểm tra thứ tự  con trỏ di  chuyển ngược  lại trên màn  hình khi nhấn  Shift-Tab | Forcus vào màn hình. Nhấn  phím Shift-Tab liên tục | Con trỏ di chuyển lần lượt theo  thứ tự: từ dưới lên trên, từ phải  qua trái. |
| BXH\_18 | Kiểm tra thực  hiện chức năng  chính của màn  hình khi nhấn  Enter | Nhấn phím Enter | 1. Nếu chuộc ko focus vào  button nào thì Thực hiện chức  năng của button chính  2. Nếu đang focus vào 1 button  thì sẽ thực hiện chức năng của  button |
|  | **Giao diện xem bảng xếp hạng chặng đua** | | |
| BXH\_19 | Kiểm tra tổng  thể giao diện  màn hình | Kiểm tra về bố cục, font chữ,  chính tả, màu chữ | Các label, textbox cùng font chữ cỡ chữ, căn lề trái, có độ dài, rộng và khoảng cách bằng nhau, không xô lệch.  - Không có lỗi về chính tả, cấu trúc câu, ngữ pháp trên màn hình  - Form được bố trí hợp lý và dễ sử dụng |
| BXH\_20 | Kiểm tra bố cục  giao diện | 1. Kiểm tra title của màn hình  2. Kiểm tra focus của chuột  3. Kiểm tra hiển thị thông tin  các trường và button trên màn  hình | 1. Hiển thị title của chức năng  2. Hiển thị đầy đủ các trường :  + Mùa giải: Textview  + Table: TT, Mã chặng, Tên chặng, Tổng điểm, Tổng thời gian(phút)  3. Hyperlink  + Chọn  4. Button  + Quay lại |
| BXH\_21 | Kiểm tra giao  diện khi thu  nhỏ, phóng to | 1. Nhấn phím Ctrl -  2. Nhấn phim Ctrl + | Màn hình thu nhỏ, phóng to tương ứng và không bị vỡ giao diện |
| BXH\_22 | Kiểm tra thứ tự di chuyển trỏ trên màn hình khi nhấn phím Tab | Forcus vào màn hình. Nhấn  Tab liên tục | Con trỏ di chuyển lần lượt theo thứ tự: Từ phải qua trái, từ trên xuống dưới. |
| BXH\_23 | Kiểm tra thứ tự  con trỏ di  chuyển ngược  lại trên màn  hình khi nhấn  Shift-Tab | Forcus vào màn hình. Nhấn  phím Shift-Tab liên tục | Con trỏ di chuyển lần lượt theo thứ tự: từ dưới lên trên, từ phải qua trái. |
| BXH\_24 | Kiểm tra thực  hiện chức năng  chính của màn  hình khi nhấn  Enter | Nhấn phím Enter | 1. Nếu chuộc ko focus vào  button nào thì Thực hiện chức  năng của button chính  2. Nếu đang focus vào 1 button  thì sẽ thực hiện chức năng của  button |
|  | **Giao diện Bảng xếp hạng tay đua** | | |
| BXH\_25 | Kiểm tra tổng  thể giao diện  màn hình | Kiểm tra về bố cục, font chữ,  chính tả, màu chữ | Các label, textbox cùng font chữ cỡ chữ, căn lề trái, có độ dài, rộng và khoảng cách bằng nhau, không xô lệch.  - Không có lỗi về chính tả, cấu trúc câu, ngữ pháp trên màn hình  - Form được bố trí hợp lý và dễ sử dụng |
| BXH\_26 | Kiểm tra bố cục  giao diện | 1. Kiểm tra title của màn hình  2. Kiểm tra focus của chuột  3. Kiểm tra hiển thị thông tin  các trường và button trên màn  hình | 1. Hiển thị title của chức năng  2. Hiển thị đầy đủ các trường :  + Mùa giải: Textview  + Table: TT, Mã, Tên tay đua, Tên chặng, thời gian về đích, thứ hạng, điểm.  3. Hyperlink  + Chọn  4. Button  + Quay lại |
| BXH\_27 | Kiểm tra giao  diện khi thu  nhỏ, phóng to | 1. Nhấn phím Ctrl -  2. Nhấn phim Ctrl + | Màn hình thu nhỏ, phóng to tương ứng và không bị vỡ giao diện |
| BXH\_28 | Kiểm tra thứ tự di chuyển trỏ trên màn hình khi nhấn phím Tab | Forcus vào màn hình. Nhấn  Tab liên tục | Con trỏ di chuyển lần lượt theo thứ tự: Từ phải qua trái, từ trên xuống dưới. |
| BXH\_29 | Kiểm tra thứ tự  con trỏ di  chuyển ngược  lại trên màn  hình khi nhấn  Shift-Tab | Forcus vào màn hình. Nhấn  phím Shift-Tab liên tục | Con trỏ di chuyển lần lượt theo  thứ tự: từ dưới lên trên, từ phải  qua trái. |
| BXH\_30 | Kiểm tra thực  hiện chức năng  chính của màn  hình khi nhấn  Enter | Nhấn phím Enter | 1. Nếu chuộc ko focus vào  button nào thì Thực hiện chức năng của button chính 2. Nếu đang focus vào 1 button thì sẽ thực hiện chức năng của button |
|  | Trường Chọn Giải Đấu, Dropdownlist | | |
|  | Kiểm tra giá trị  mặc định | 1. Trên màn hình Xem bảng xếp hạng  2. Kiểm tra giá trị mặc định  của trường: Chọn thống kê | Giá trị mặc định của Chọn  thống kê: TK xem bảng xếp hạng đội đua |
|  | Kiểm tra danh  sách các loại  thống kê | 1. Focus chuột vào dropdown  list Chọn thống kê | Hệ thống hiển thị danh sách  các loại Thống kê |

**Data test:**

* **Thành viên:**
  + INSERT INTO bxhdoidua.tblthanhvien (dia\_chi, email, ghi\_chu, ho\_ten, ngay\_sinh, password, sdt, username) VALUES ('Hà Tĩnh', 'thanhpham.23102000@gmail.com', 'Phạm Công Thành', 'Phạm Công Thành', '2000-10-23 16:28:59.000000', 'thanhpc', '0986249793', 'thanhpc')
  + INSERT INTO bxhdoidua.tblthanhvien (dia\_chi, email, ghi\_chu, ho\_ten, ngay\_sinh, password, sdt, username) VALUES ('Hà Nội', 'hnoiv@gmail.com', 'Phạm Công Bình', 'Phạm Công Bình', '2001-11-27 16:30:16.000000', 'binhpc', '0999846793', 'binhpc')
  + INSERT INTO bxhdoidua.tblthanhvien (dia\_chi, email, ghi\_chu, ho\_ten, ngay\_sinh, password, sdt, username) VALUES ('Nghệ An', 'btc2022@gmail.com', 'Đinh Nhật Lệ', 'Đinh Thị Nhật Lệ', '1987-11-27 16:31:30.000000', 'nhatledn', '0866497796', 'nhatledn')
* **Tay đua:**
  + INSERT INTO bxhdoidua.tbltaydua (tbl\_thanh\_vien\_id, quoc\_tich, tieu\_su) VALUES (1, 'Việt Nam', 'Tay đua vô địch aff1')
  + INSERT INTO bxhdoidua.tbltaydua (tbl\_thanh\_vien\_id, quoc\_tich, tieu\_su) VALUES (2, 'Việt Nam', 'Tay đua vô địch world cup')
* **Ban tổ chức:** 
  + INSERT INTO bxhdoidua.tblbantochuc (tbl\_thanh\_vien\_id, mo\_ta, ten) VALUES (3, 'Quản lý', 'VFF-CUP')
* **Giải đấu:**
  + INSERT INTO bxhdoidua.tblgiaidau (nam\_to\_chuc) VALUES (2019)
  + INSERT INTO bxhdoidua.tblgiaidau (nam\_to\_chuc) VALUES (2020)
  + INSERT INTO bxhdoidua.tblgiaidau (nam\_to\_chuc) VALUES (2021)
  + INSERT INTO bxhdoidua.tblgiaidau (nam\_to\_chuc) VALUES (2022)
* **Đội đua:** 
  + INSERT INTO bxhdoidua.tbldoidua (hang, mo\_ta, ten, tbl\_giai\_dau\_id) VALUES ('Honda', 'Nhật bản', 'Honda-hd1', 3)
  + INSERT INTO bxhdoidua.tbldoidua (hang, mo\_ta, ten, tbl\_giai\_dau\_id) VALUES ('Mazda', 'Nhật bản', 'Mazda-md2', 3)
  + INSERT INTO bxhdoidua.tbldoidua (hang, mo\_ta, ten, tbl\_giai\_dau\_id) VALUES ('Toyota', 'Nhật bản', 'Toyota-tyt3', 3)
  + INSERT INTO bxhdoidua.tbldoidua (hang, mo\_ta, ten, tbl\_giai\_dau\_id) VALUES ('Lexus', 'Nhật bản', 'Lexus-lx1', 2)
* **Chặng đua:** 
  + INSERT INTO bxhdoidua.tblchangdua (dia\_diem, mo\_ta, so\_vong\_dua, ten, thoi\_gian, tbl\_giai\_dau\_id) VALUES ('Hà Giang – Hà Tĩnh', 'Đường gồ ghề', 2, 'Hà Giang – Hà Tĩnh', '2022-11-27 16:21:14.000000', 3)
  + INSERT INTO bxhdoidua.tblchangdua (dia\_diem, mo\_ta, so\_vong\_dua, ten, thoi\_gian, tbl\_giai\_dau\_id) VALUES ('Hà Tĩnh – Đà Nẵng', 'Nhiều cây xanh', 3, 'Hà Tĩnh – Đà Nẵng', '2022-07-27 16:21:46.000000', 3)
  + INSERT INTO bxhdoidua.tblchangdua (dia\_diem, mo\_ta, so\_vong\_dua, ten, thoi\_gian, tbl\_giai\_dau\_id) VALUES ('Đà Nẵng – Cà Mau', 'Thoáng mát', 2, 'Đà Nẵng – Cà Mau', '2022-05-04 20:22:11.000000', 3)
  + bxhdoidua.tblchangdua (dia\_diem, mo\_ta, so\_vong\_dua, ten, thoi\_gian, tbl\_giai\_dau\_id) VALUES ('Sài Gòn - Lào Cai', 'Có bão đi qua', 1, 'Sài Gòn - Lào Cai', '2021-11-03 16:22:52.000000', 2)
* **Đội đua tay đua:** 
  + INSERT INTO bxhdoidua.tbldoiduataydua (thoi\_diem, tbl\_doi\_dua\_id, tbl\_thanh\_vien\_id) VALUES ('2022-08-27 17:09:50.000000', 1, 1)
  + INSERT INTO bxhdoidua.tbldoiduataydua (thoi\_diem, tbl\_doi\_dua\_id, tbl\_thanh\_vien\_id) VALUES ('2022-06-27 17:10:05.000000', 1, 2)
* **Kết quả:**
  + INSERT INTO bxhdoidua.tblketqua (diem, thoi\_gian\_ve\_dich, thu\_hang, tbl\_chang\_dua\_id, tbl\_doi\_dua\_tay\_dua\_id) VALUES (30, 100, 1, 1, 1)
  + INSERT INTO bxhdoidua.tblketqua (diem, thoi\_gian\_ve\_dich, thu\_hang, tbl\_chang\_dua\_id, tbl\_doi\_dua\_tay\_dua\_id) VALUES (20, 200, 2, 1, 2)
* **Import tất cả data trên vào mysql:**
* INSERT INTO bxhdoidua.tblgiaidau (id, nam\_to\_chuc) VALUES (1, 2020);  
  INSERT INTO bxhdoidua.tblgiaidau (id, nam\_to\_chuc) VALUES (2, 2021);  
  INSERT INTO bxhdoidua.tblgiaidau (id, nam\_to\_chuc) VALUES (3, 2022);  
  INSERT INTO bxhdoidua.tblgiaidau (id, nam\_to\_chuc) VALUES (4, 2019);  
    
    
  INSERT INTO bxhdoidua.tblchangdua (id, dia\_diem, mo\_ta, so\_vong\_dua, ten, thoi\_gian, tbl\_giai\_dau\_id) VALUES (1, 'Hà Giang – Hà Tĩnh', 'Đường gồ ghề', 2, 'Hà Giang – Hà Tĩnh', '2022-11-27 16:21:14', 3);  
  INSERT INTO bxhdoidua.tblchangdua (id, dia\_diem, mo\_ta, so\_vong\_dua, ten, thoi\_gian, tbl\_giai\_dau\_id) VALUES (2, 'Hà Tĩnh – Đà Nẵng', 'Nhiều cây xanh', 3, 'Hà Tĩnh – Đà Nẵng', '2022-07-27 16:21:46', 3);  
  INSERT INTO bxhdoidua.tblchangdua (id, dia\_diem, mo\_ta, so\_vong\_dua, ten, thoi\_gian, tbl\_giai\_dau\_id) VALUES (3, 'Đà Nẵng – Cà Mau', 'Thoáng mát', 2, 'Đà Nẵng – Cà Mau', '2022-05-04 20:22:11', 3);  
  INSERT INTO bxhdoidua.tblchangdua (id, dia\_diem, mo\_ta, so\_vong\_dua, ten, thoi\_gian, tbl\_giai\_dau\_id) VALUES (4, 'Sài Gòn - Lào Cai', 'Có bão đi qua', 1, 'Sài Gòn - Lào Cai', '2021-11-03 16:22:52', 2);  
    
    
  INSERT INTO bxhdoidua.tblthanhvien (id, dia\_chi, email, ghi\_chu, ho\_ten, ngay\_sinh, password, sdt, username) VALUES (1, 'Hà Tĩnh', 'thanhpham.23102000@gmail.com', 'Phạm Công Thành', 'Phạm Công Thành', '2000-10-23 16:28:59', 'thanhpc', '0986249793', 'thanhpc');  
  INSERT INTO bxhdoidua.tblthanhvien (id, dia\_chi, email, ghi\_chu, ho\_ten, ngay\_sinh, password, sdt, username) VALUES (2, 'Hà Nội', 'hnoiv@gmail.com', 'Phạm Công Bình', 'Phạm Công Bình', '2001-11-27 16:30:16', 'binhpc', '0999846793', 'binhpc');  
  INSERT INTO bxhdoidua.tblthanhvien (id, dia\_chi, email, ghi\_chu, ho\_ten, ngay\_sinh, password, sdt, username) VALUES (3, 'Nghệ An', 'btc2022@gmail.com', 'Đinh Nhật Lệ', 'Đinh Thị Nhật Lệ', '1987-11-27 16:31:30', 'nhatledn', '0866497796', 'nhatledn');  
    
  INSERT INTO `bxhdoidua`.`tblthanhvien` (`id`, `dia\_chi`, `email`, `ghi\_chu`, `ho\_ten`, `ngay\_sinh`, `password`, `sdt`, `username`) VALUES ('4', 'Nghệ An', 'thanhpham.23102000@gmail.com', 'Phạm Công Tuấn', 'Phạm Công Tuấn', '2000-10-23 16:28:59.000000', 'tuanpc', '0866497796', 'tuanpc');  
  INSERT INTO `bxhdoidua`.`tblthanhvien` (`id`, `dia\_chi`, `email`, `ghi\_chu`, `ho\_ten`, `ngay\_sinh`, `password`, `sdt`, `username`) VALUES ('5', 'Nghệ An', 'thanhpham.23102000@gmail.com', 'Phạm Công Long', 'Phạm Công Long', '2000-10-23 16:28:59.000000', 'longpc', '0866497796', 'longpc');  
  INSERT INTO `bxhdoidua`.`tblthanhvien` (`id`, `dia\_chi`, `email`, `ghi\_chu`, `ho\_ten`, `ngay\_sinh`, `password`, `sdt`, `username`) VALUES ('6', 'Nghệ An', 'thanhpham.23102000@gmail.com', 'Phạm Công Luân', 'Phạm Công Luân', '2000-10-23 16:28:59.000000', 'luanpc', '0866497796', 'luanpc');  
  INSERT INTO `bxhdoidua`.`tblthanhvien` (`id`, `dia\_chi`, `email`, `ghi\_chu`, `ho\_ten`, `ngay\_sinh`, `password`, `sdt`, `username`) VALUES ('7', 'Nghệ An', 'thanhpham.23102000@gmail.com', 'Phạm Công Nhật', 'Phạm Công Nhật', '2000-10-23 16:28:59.000000', 'nhatpc', '0866497796', 'nhatpc');  
    
    
  INSERT INTO bxhdoidua.tbltaydua (tbl\_thanh\_vien\_id, quoc\_tich, tieu\_su) VALUES (1, 'Việt Nam', 'Tay đua vô địch aff1');  
  INSERT INTO bxhdoidua.tbltaydua (tbl\_thanh\_vien\_id, quoc\_tich, tieu\_su) VALUES (2, 'Việt Nam ', 'Tay đua vô địch world cup');  
    
  INSERT INTO `bxhdoidua`.`tbltaydua` (`tbl\_thanh\_vien\_id`, `quoc\_tich`, `tieu\_su`) VALUES ('4', 'Việt Nam ', 'Tay đua vô địch world cup');  
  INSERT INTO `bxhdoidua`.`tbltaydua` (`tbl\_thanh\_vien\_id`, `quoc\_tich`, `tieu\_su`) VALUES ('5', 'Việt Nam ', 'Tay đua vô địch euro');  
  INSERT INTO `bxhdoidua`.`tbltaydua` (`tbl\_thanh\_vien\_id`, `quoc\_tich`, `tieu\_su`) VALUES ('6', 'Việt Nam ', 'Tay đua vô địch copa america');  
  INSERT INTO `bxhdoidua`.`tbltaydua` (`tbl\_thanh\_vien\_id`, `quoc\_tich`, `tieu\_su`) VALUES ('7', 'Việt Nam ', 'Tay đua vô địch numberz');  
    
    
  INSERT INTO bxhdoidua.tbldoidua (id, hang, mo\_ta, ten, tbl\_giai\_dau\_id) VALUES (1, 'Honda', 'Nhật bản', 'Honda-hd1', 3);  
  INSERT INTO bxhdoidua.tbldoidua (id, hang, mo\_ta, ten, tbl\_giai\_dau\_id) VALUES (2, 'Mazda', 'Nhật bản', 'Mazda-md2', 3);  
  INSERT INTO bxhdoidua.tbldoidua (id, hang, mo\_ta, ten, tbl\_giai\_dau\_id) VALUES (3, 'Toyota', 'Nhật bản', 'Toyota-tyt3', 3);  
  INSERT INTO bxhdoidua.tbldoidua (id, hang, mo\_ta, ten, tbl\_giai\_dau\_id) VALUES (4, 'Lexus', 'Nhật bản', 'Lexus-lx1', 2);  
    
    
  INSERT INTO bxhdoidua.tbldoiduataydua (id, thoi\_diem, tbl\_doi\_dua\_id, tbl\_thanh\_vien\_id) VALUES (1, '2022-08-27 17:09:50', 1, 1);  
  INSERT INTO bxhdoidua.tbldoiduataydua (id, thoi\_diem, tbl\_doi\_dua\_id, tbl\_thanh\_vien\_id) VALUES (2, '2022-06-27 17:10:05', 1, 2);  
    
  INSERT INTO `bxhdoidua`.`tbldoiduataydua` (`id`, `thoi\_diem`, `tbl\_doi\_dua\_id`, `tbl\_thanh\_vien\_id`) VALUES ('3', '2022-06-27 17:10:05.000000', '2', '4');  
  INSERT INTO `bxhdoidua`.`tbldoiduataydua` (`id`, `thoi\_diem`, `tbl\_doi\_dua\_id`, `tbl\_thanh\_vien\_id`) VALUES ('4', '2022-06-27 17:10:05.000000', '2', '5');  
  INSERT INTO `bxhdoidua`.`tbldoiduataydua` (`id`, `thoi\_diem`, `tbl\_doi\_dua\_id`, `tbl\_thanh\_vien\_id`) VALUES ('5', '2022-06-27 17:10:05.000000', '3', '6');  
  INSERT INTO `bxhdoidua`.`tbldoiduataydua` (`id`, `thoi\_diem`, `tbl\_doi\_dua\_id`, `tbl\_thanh\_vien\_id`) VALUES ('6', '2022-06-27 17:10:05.000000', '3', '7');  
    
    
  INSERT INTO bxhdoidua.tblketqua (id, diem, thoi\_gian\_ve\_dich, thu\_hang, tbl\_chang\_dua\_id, tbl\_doi\_dua\_tay\_dua\_id) VALUES (1, 30, 100, 1, 1, 1);  
  INSERT INTO bxhdoidua.tblketqua (id, diem, thoi\_gian\_ve\_dich, thu\_hang, tbl\_chang\_dua\_id, tbl\_doi\_dua\_tay\_dua\_id) VALUES (2, 20, 200, 2, 1, 2);  
    
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('3', '20', '80', '1', '2', '3');  
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('4', '10', '120', '2', '2', '4');  
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('5', '10', '130', '1', '3', '5');  
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('6', '10', '170', '2', '3', '6');  
  UPDATE `bxhdoidua`.`tblketqua` SET `thoi\_gian\_ve\_dich` = '40' WHERE (`id` = '1');  
  UPDATE `bxhdoidua`.`tblketqua` SET `thoi\_gian\_ve\_dich` = '60' WHERE (`id` = '2');  
    
  -- update on 30/11/2022  
  UPDATE `bxhdoidua`.`tblketqua` SET `diem` = '15', `thoi\_gian\_ve\_dich` = '100' WHERE (`id` = '1');  
  UPDATE `bxhdoidua`.`tblketqua` SET `diem` = '5', `thoi\_gian\_ve\_dich` = '200' WHERE (`id` = '2');  
  UPDATE `bxhdoidua`.`tblketqua` SET `tbl\_doi\_dua\_tay\_dua\_id` = '1' WHERE (`id` = '3');  
  UPDATE `bxhdoidua`.`tblketqua` SET `diem` = '15', `tbl\_doi\_dua\_tay\_dua\_id` = '2' WHERE (`id` = '4');  
  UPDATE `bxhdoidua`.`tblketqua` SET `diem` = '25', `thoi\_gian\_ve\_dich` = '40', `tbl\_doi\_dua\_tay\_dua\_id` = '1' WHERE (`id` = '5');  
  UPDATE `bxhdoidua`.`tblketqua` SET `diem` = '20', `thoi\_gian\_ve\_dich` = '60', `tbl\_doi\_dua\_tay\_dua\_id` = '2' WHERE (`id` = '6');  
    
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('7', '15', '120', '3', '1', '3');  
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('8', '10', '180', '4', '1', '4');  
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('9', '20', '70', '3', '2', '3');  
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('10', '10', '130', '4', '2', '4');  
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('11', '20', '50', '3', '3', '3');  
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('12', '15', '100', '4', '3', '4');  
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('13', '15', '100', '5', '1', '5');  
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('14', '5', '200', '6', '1', '6');  
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('15', '15', '100', '5', '2', '5');  
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('16', '10', '150', '6', '2', '6');  
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('17', '20', '50', '5', '3', '5');  
  INSERT INTO `bxhdoidua`.`tblketqua` (`id`, `diem`, `thoi\_gian\_ve\_dich`, `thu\_hang`, `tbl\_chang\_dua\_id`, `tbl\_doi\_dua\_tay\_dua\_id`) VALUES ('18', '15', '150', '6', '3', '6');  
    
    
  INSERT INTO bxhdoidua.tblbantochuc (tbl\_thanh\_vien\_id, mo\_ta, ten) VALUES (3, 'Quản lý', 'VFF-CUP');