

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



# BÀI TẬP

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

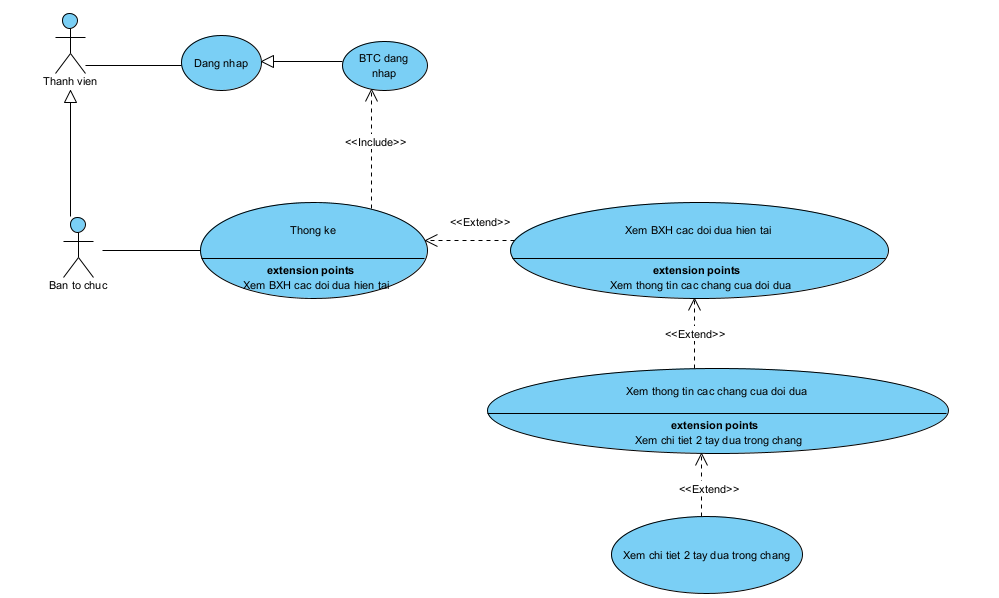
# MODULE: XEM BXH CÁC ĐỘI ĐUA(11.5)

**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

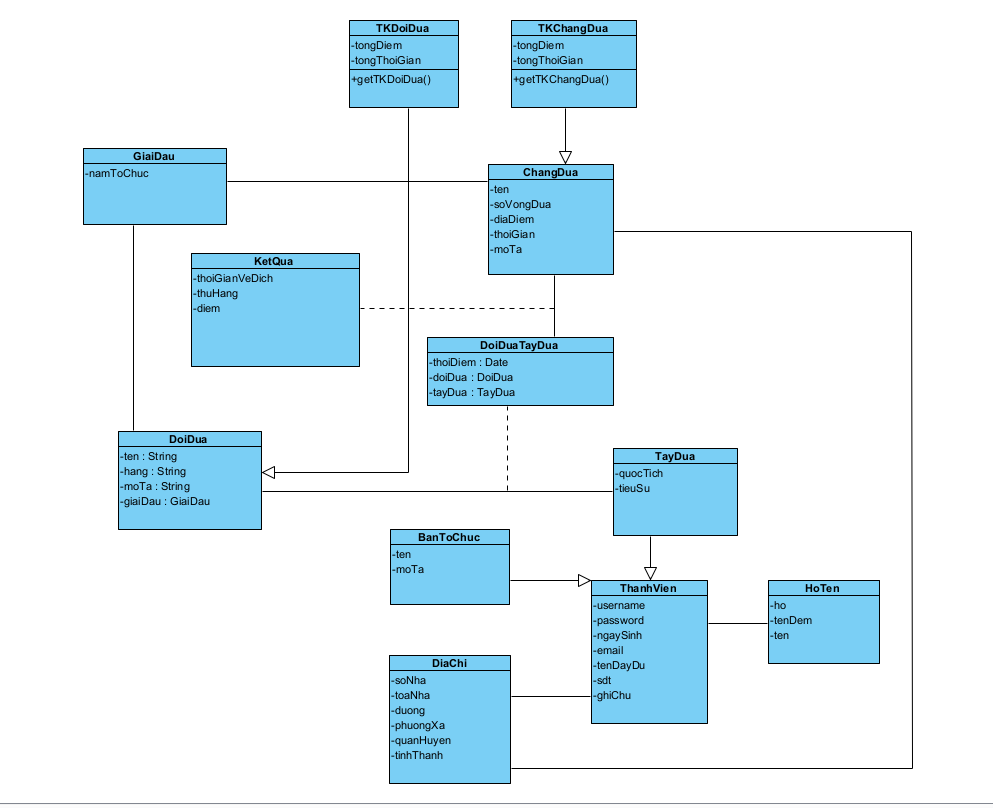
1. **Viết kịch bản**

**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 | 50 | 100 | | 2 | Hà Tĩnh – Đà Nẵng | 30 | 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 | 30 | | 2 | Phạm Công Bình | Hà Giang – Hà Tĩnh | 60 | 2 | 20 | |
| Ngoại lệ | 4. Năm 2022 không có đội đua nào thi đấu |

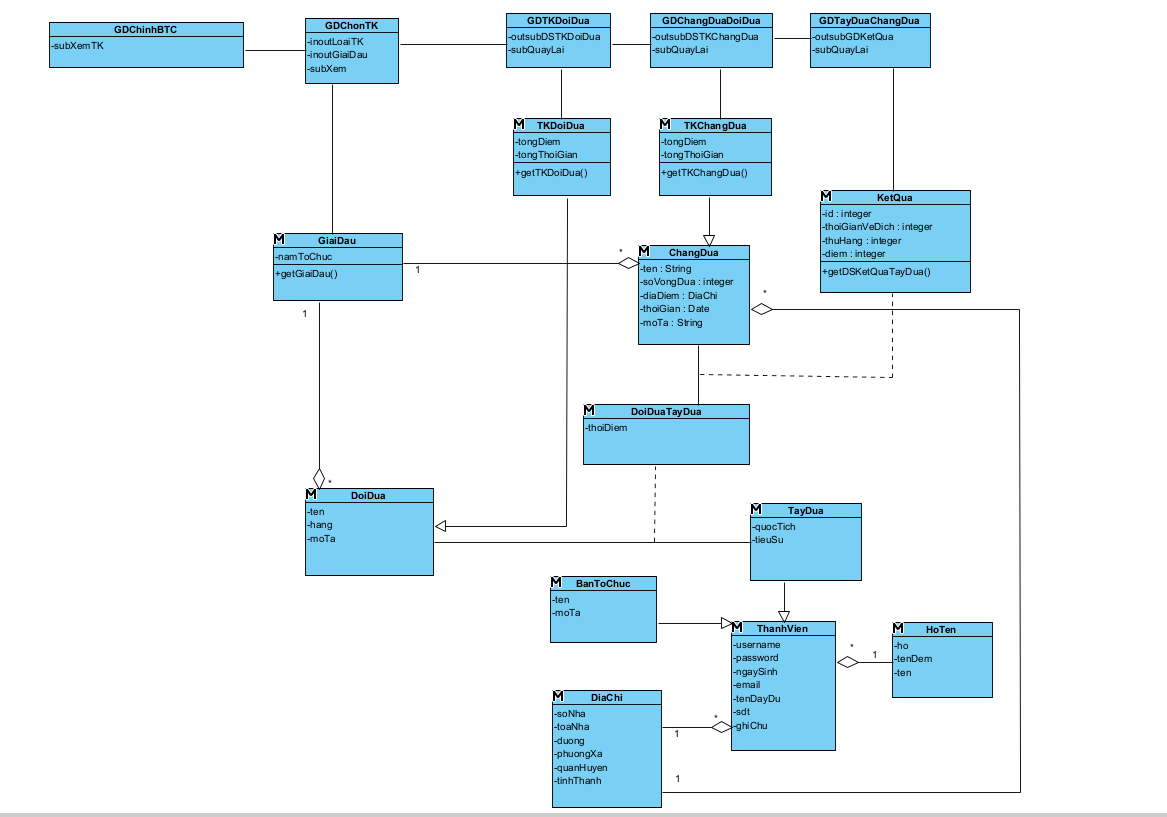
1. **Trích lớp thực thể và biểu đồ lớp thực thể liên quan đến modul xem bảng xếp hạng các đội đua**

* **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:



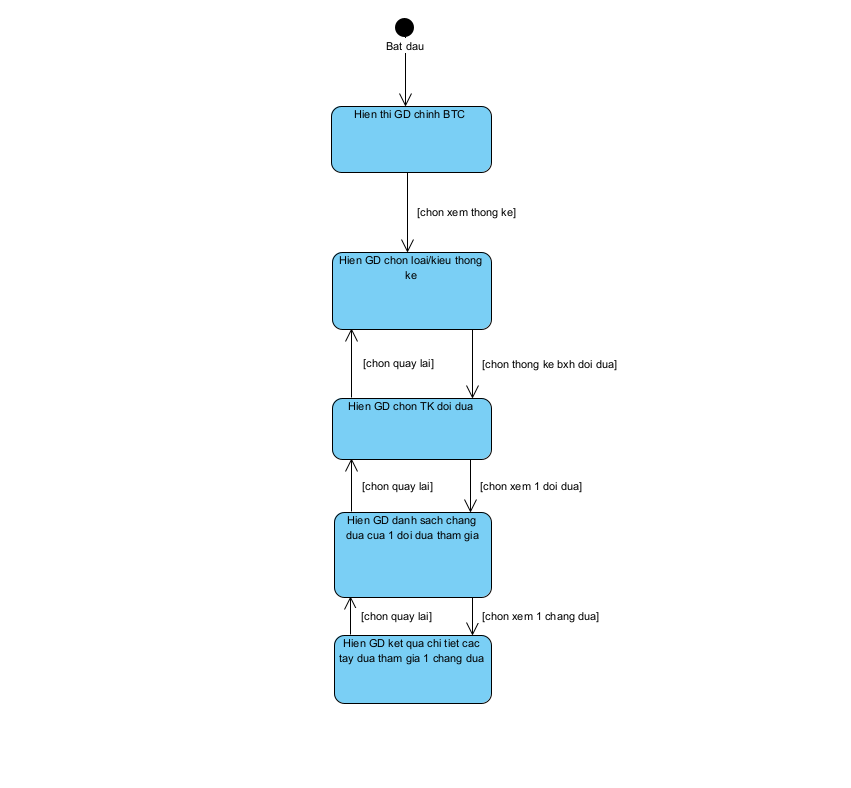
1. **Phân tích tĩnh cho modul xem BXH các đội đua**

* 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



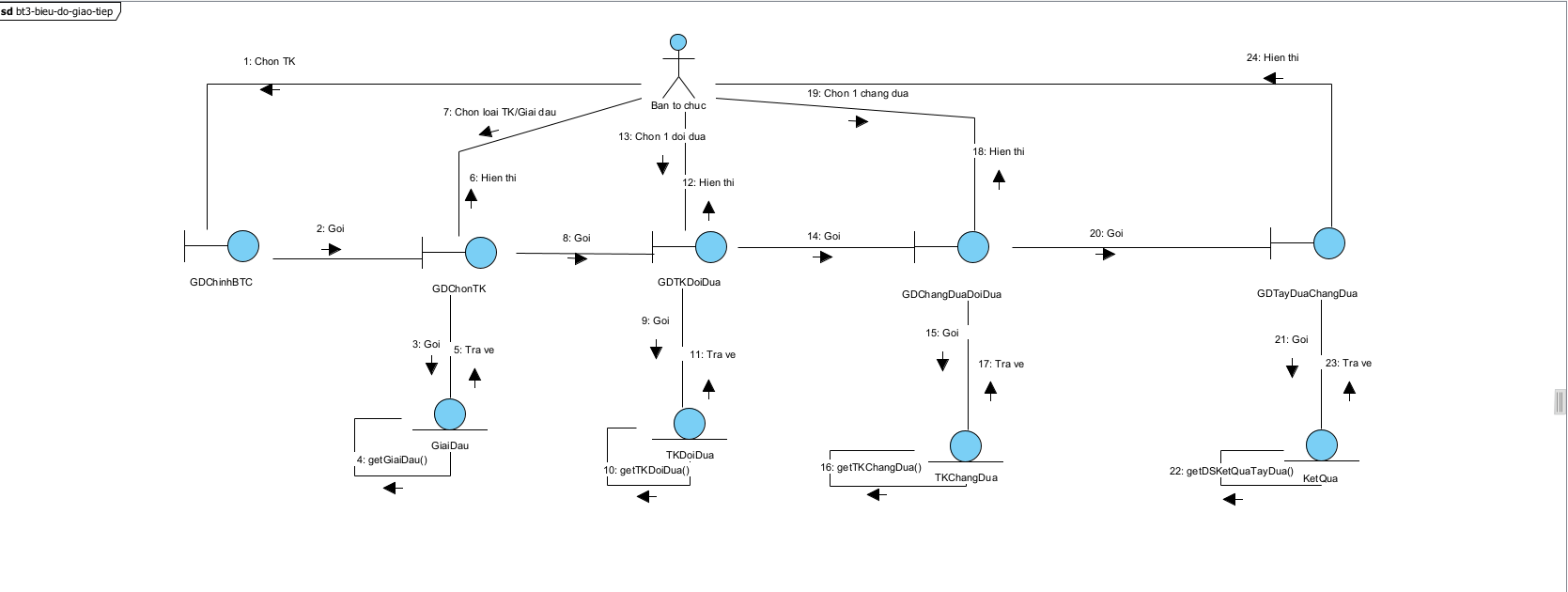
1. **Biểu đồ trạng thái module xem BXH các đội đua**

* 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

****

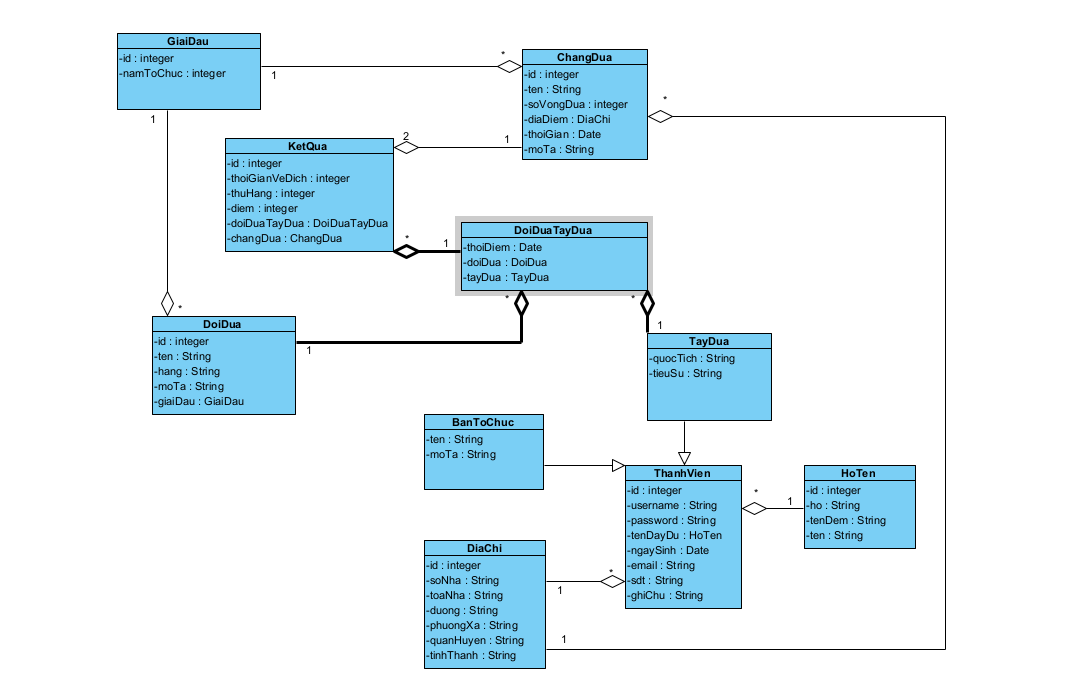
1. **Biểu đồ giao tiếp module xem BXH các đội đua**

* 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



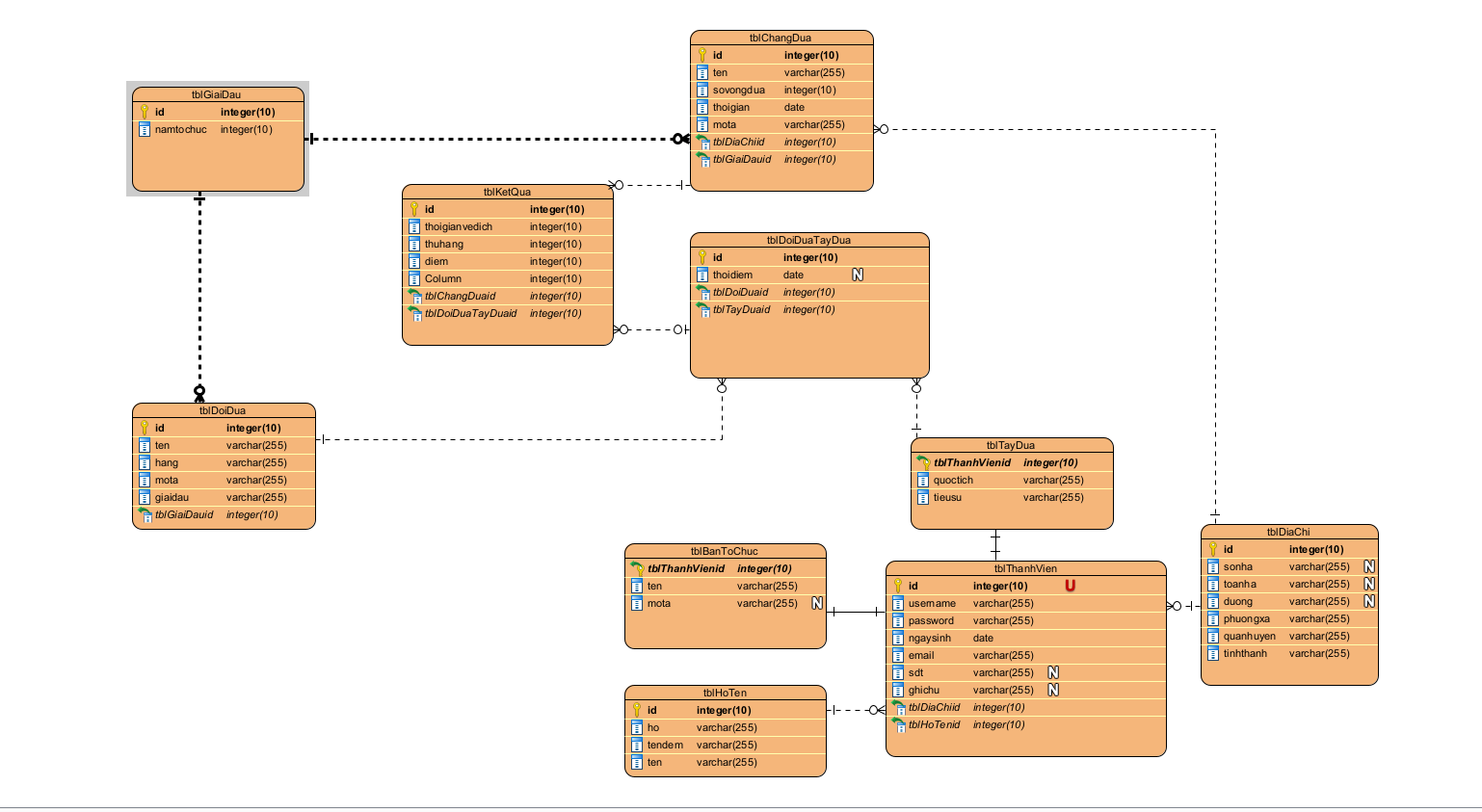
1. **Thiết kế lớp thực thể**

* 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:



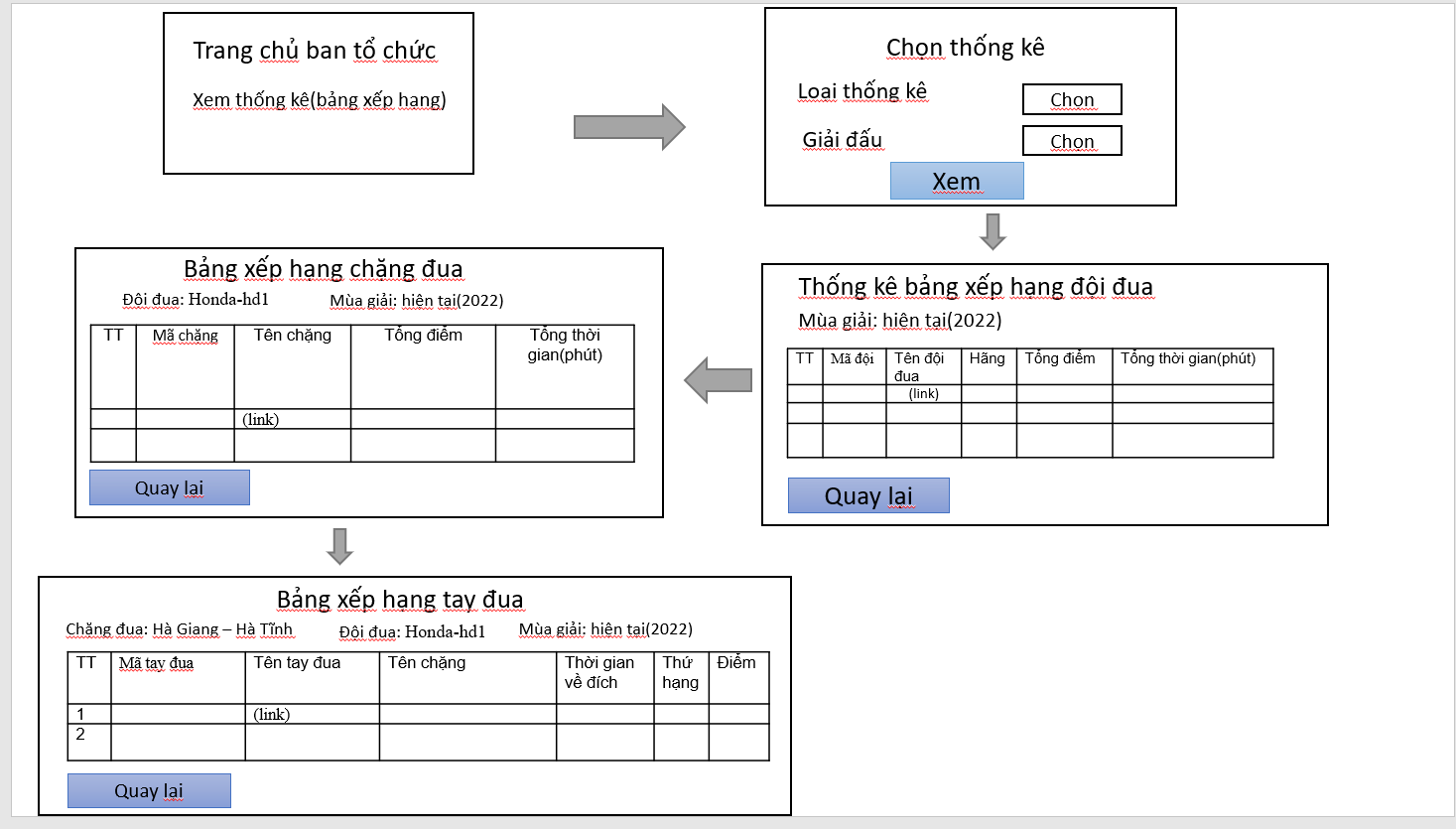
1. **Thiết kế cơ sở dữ liệu**

* 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**

* 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: gdChinhQL, gdChonTK, gdTKBXHDoiDua, gdBXHChangDua, gdBXHTayDua.
  + Tầng nhận request từ client và định tuyến router gồm các lớp HomeController.java, ThongKeController.java
  + Tầng xử lý logic có các lớp GiaiDauService.java, DoiDuaService.java, ChangDuaService.java, TayDuaService.java
  + Tầng truy cập dữ liệu có các interface GiaiDauRepository.java, DoiDuaRepository.java, ChangDuaRepository.java, TayDuaRepository.java kế thừa từ CrudRepository
  + 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)

****

1. **Biểu đồ thiết kế lớp thực thể**

**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)