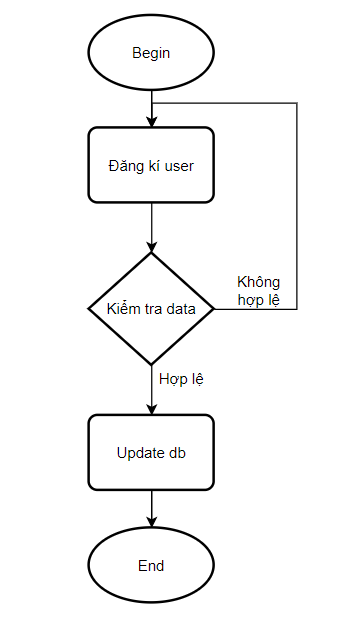
1. **Đăng kí user mới (admin): Mã MH: 001**

* Mô tả: Add/Register 1 user mới, tài khoản admin đã đăng nhập gởi các thông tin cần thiết, server kiểm tra, nếu hợp lệ thì tiến hành mã hóa password và lưu database, trả kết quả về client có thành công hay không (tài khoản admin thì tạo bằng cách ??)



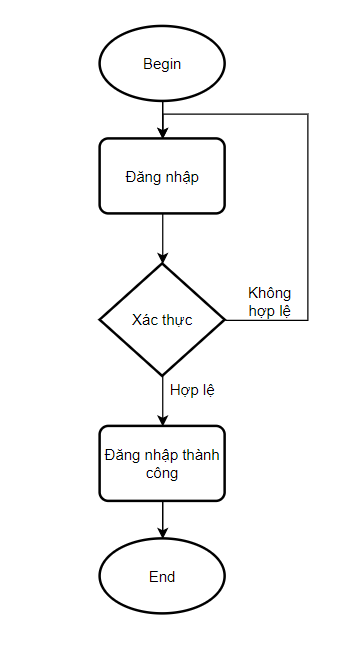
* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | email\_address | VARCHAR2(255) | Email của user |
| 2 | full\_name | VARCHAR2(255) | Full name của user |
| 3 | user\_name | VARCHAR2(64) | Username |
| 4 | password | VARCHAR2(32) | Password |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |

* Yêu cầu front end: tạo form để user input thông tin, gởi đến server qua giao thức post với dữ liệu body như input phía trên.
* Yêu cầu backend xây dựng API như sau: kiểm tra các thông tin của dữ liệu input, nếu hợp lệ, trùng user và email trong database thì tiến hành băm và lưu mật khẩu, thông tin khác vào database, sau đó phản hồi thành công về client hoặc phản hồi thất bại nếu thông tin không hợp lệ.

1. **Đăng nhập: Mã MH: 002 (pop up bất cứ đâu)**

* Mô tả: Người dùng truy cập màn hình đăng nhập, nhập thông tin username, password, hệ thống tiến hành băm và kiểm tra mật khẩu. Thông báo cho user nếu thất bại hoặc chuyển lại màn hình phù hợp nếu đăng nhập thành công.



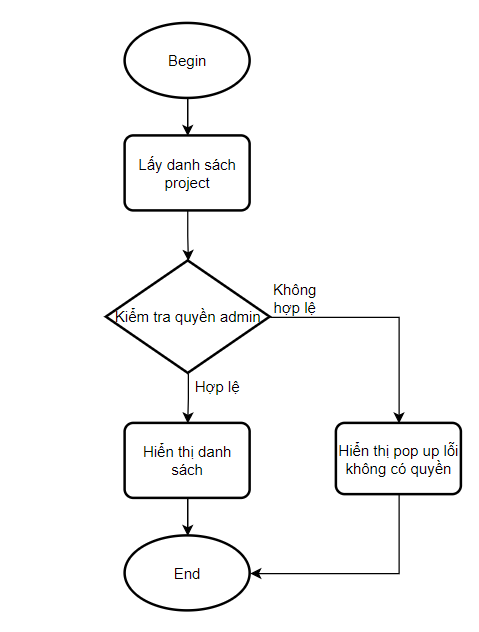
* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | user | VARCHAR2(64) | Email hoặc username |
| 2 | password | VARCHAR2(32) | Password của user đó |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |

* Yêu cầu front end: tạo form để user input thông tin, gởi đến server qua giao thức post với dữ liệu body như input phía trên, sau khi nhận kết quả phía server, client kiểm tra nếu đã nhận được thông tin đăng nhập cùng token thì tiến hành lưu token này lại cho những lần gọi request tiếp theo.
* Yêu cầu backend xây dựng API như sau: kiểm tra các thông tin của dữ liệu input, nếu tìm được thông tin đăng nhập của user phù hợp. Tiến hành kiểm tra và xóa những token đã hết hạn, tạo mới token cho phiên đăng nhập này và gởi thông tin token về cho client.

1. **Quản lí project (admin):** 
   1. Hiển thị danh sách project: Mã MH: 003

Mô tả: Người dùng admin chọn chức năng quản lý project, chương trình hiển thị toàn bộ thông tin danh sách project hiện có. Cùng các nút chức năng thêm project. Sửa/Xóa cho từng project riêng biệt

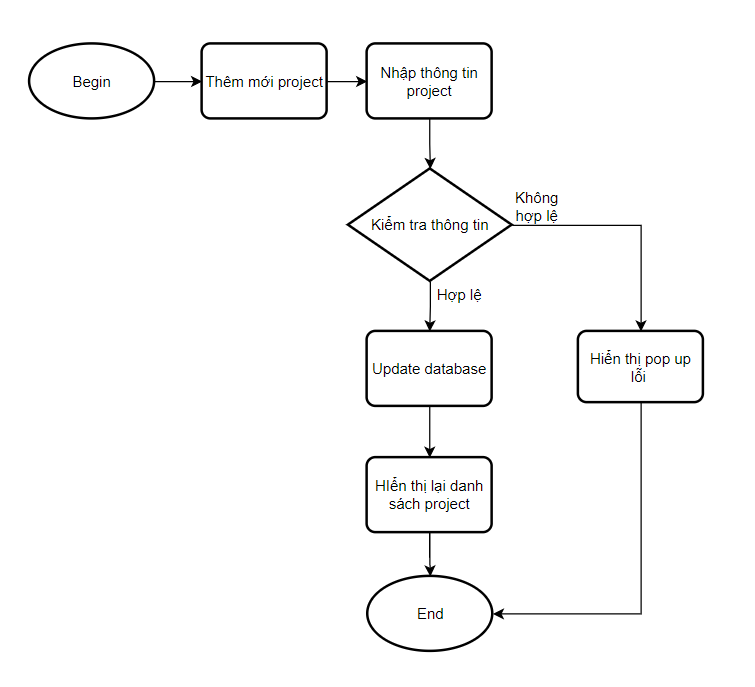


* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | token | VARCHAR2(150) | Mã token đã được đăng nhập thành công trước đó và còn hạn sử dụng |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |
| 3 | data | MAX | Dữ liệu danh sách toàn bộ các project |

* Yêu cầu front end: sau khi user chuyển đến màn hình quản lý project, fontend gởi request để nhận tạo được một bảng liệt kê toàn bộ các dánh sách, sau đó hiển thị toàn bộ danh sách này, gồm các cột Name, Key, Project lead và 2 button edit, delete cho từng dòng.
* Yêu cầu backend xây dựng API như sau: kiểm tra các thông tin của dữ liệu input, nếu token là của người dùng admin và còn hạn thì tiến hành query và trả kết quả cho client. Không thì báo kết quả thất bại
  1. Thêm mới project. Mã MH 004 (pop up trên màn hình 003)

Mô tả: Người dùng admin truy cập màn hình quản lí project -> chọn tạo mới project, nhập các thông tin như: Tên project, mã project, project lead (dưới dạng drop box để người dùng lựa chọn những user đang có trong hệ thống). Thông báo cho user nếu thất bại hoặc tạo project thành công.

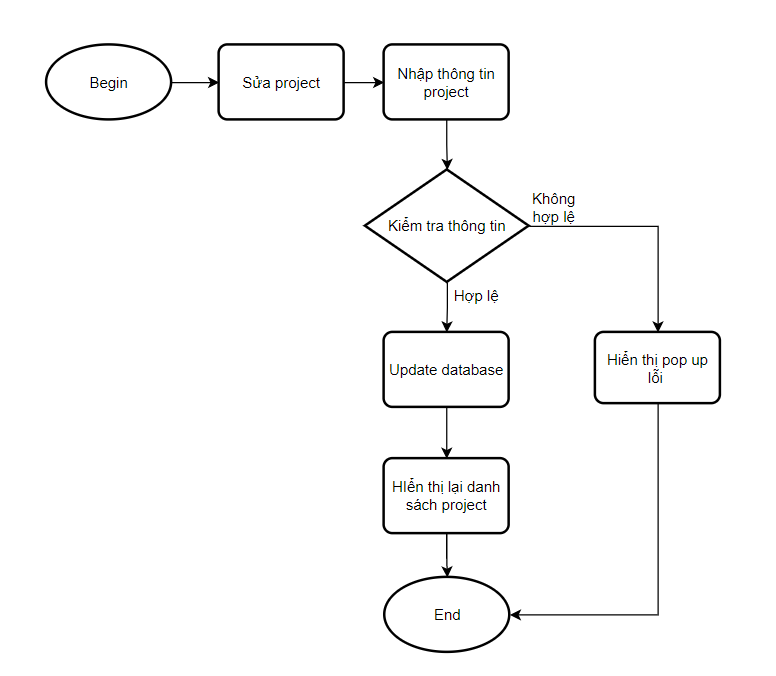


* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | project\_name | VARCHAR2(255) | Tên project |
| 2 | project\_key | VARCHAR2(3) | Mã của project |
| 3 | user\_name | VARCHAR2(64) | Tên người quản lí project |
| 4 | project\_description | VARCHAR(500) | Mô tả thêm về project |
| 5 | token | VARCHAR2(150) | Mã token đã được đăng nhập thành công trước đó và còn hạn sử dụng |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |

* Yêu cầu front end: tạo form để user input thông tin, gởi đến server qua giao thức post với dữ liệu body như input phía trên (1, 2, 3, 4) + mã token (5) đã được đăng nhập thành công bởi user admin trước đó, sau khi có kết quả thành công, hiển thị pop up thông báo và tiến hành gọi api để làm mới danh sách projects.
* Yêu cầu backend xây dựng API như sau: kiểm tra các thông tin của dữ liệu input, project\_key và project name phải là duy nhất, user\_name phải là một user đã tồn tại trong database. Sau đó trả kết quả về thành công hay thất bại.
  1. Sửa project: MH 005 (pop up)

Mô tả: Người dùng admin truy cập màn hình quản lí project -> chọn button sửa project cho project cần sửa, hiển thị pop up để user nhập các thông tin như: Tên project, mã project, project lead (dưới dạng drop box để người dùng lựa chọn những user đang có trong hệ thống), mô tả. Thông báo cho user nếu thất bại hoặc sửa project thành công.

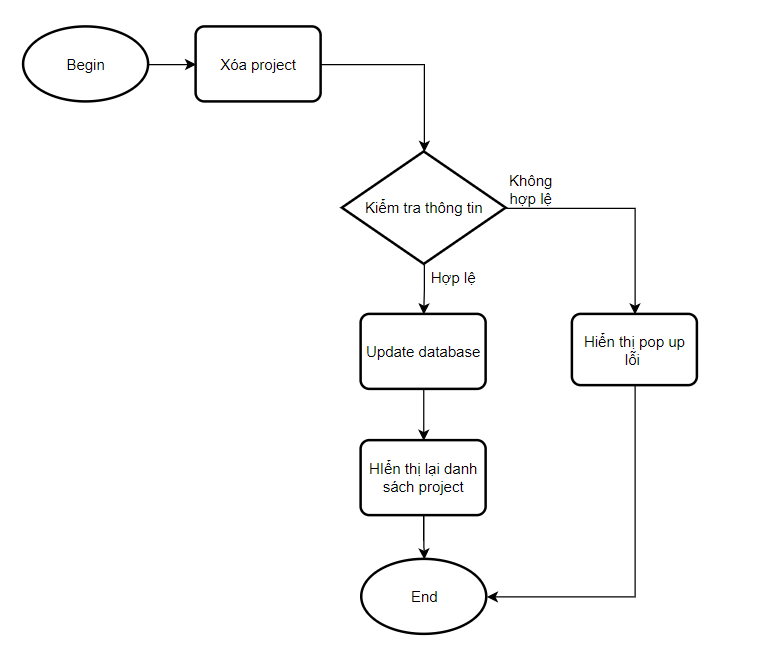


* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | old\_project\_key | VARCHAR(3) | Mã project hiện tại |
| 2 | project\_name | VARCHAR2(255) | Tên mới project |
| 3 | project\_key | VARCHAR2(3) | Mã mới của project |
| 4 | user\_name | VARCHAR2(64) | Tên người quản lí project |
| 5 | project\_description | VARCHAR(500) | Mô tả thêm về project |
| 6 | token | VARCHAR2(150) | Mã token đã được đăng nhập thành công trước đó và còn hạn sử dụng |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |

* Yêu cầu front end: tạo form để user input thông tin, gởi đến server qua giao thức post với dữ liệu body như input phía trên (1, 2, 3, 4, 6) + mã token (6) đã được đăng nhập thành công bởi user admin trước đó, sau khi có kết quả thành công, hiển thị pop up thành công và tiến hành gọi api để làm mới danh sách projects.
* Yêu cầu backend xây dựng API như sau: kiểm tra các thông tin của dữ liệu input, kiểm tra có mã project cũ đang tồn tại hay không, mã mới có trùng với các mã khác,… nếu thông tin hợp lệ thì update toàn bộ các bảng có liên quan và trả về kết quả cho client
  1. Xóa project: MH006 (pop up confirm)

Mô tả: Người dùng admin truy cập màn hình quản lí project -> chọn button xóa project cho project cần xóa, hiển thị pop up để user xác nhận đồng ý xóa hay không, nếu user đồng ý thì tiến hành xóa và làm mới danh sách projects.



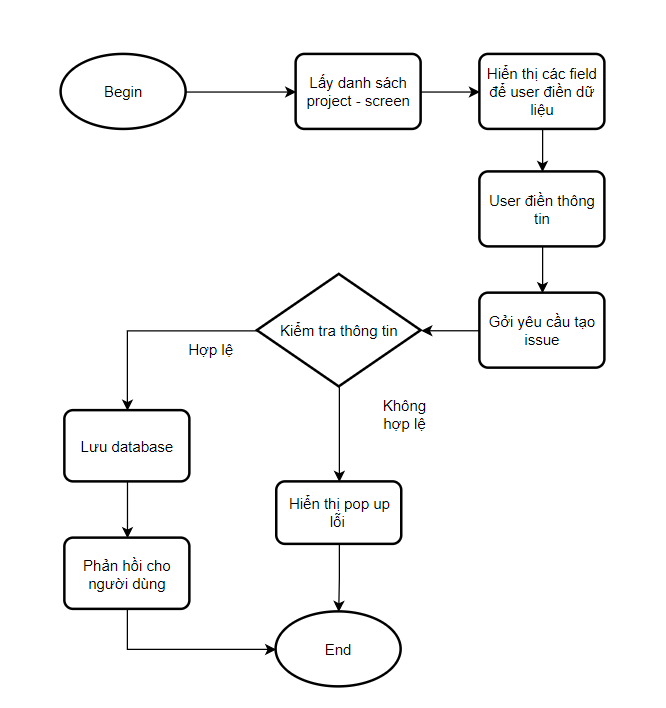
* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | project\_key | VARCHAR(3) | Mã project hiện tại |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |

* Yêu cầu front end: tạo form để user confirm có đồng ý xác nhận xóa project không.
* Yêu cầu backend xây dựng API như sau: kiểm tra mã project mà user muốn xóa, tiến hành xóa toàn bộ dữ liệu liên quan của project ở tất cả các bảng và phản hồi về client.

1. **Tạo mới issue (admin + common user)**

* Mô tả: User click nút create (issue) trên thanh top bar. Người dùng nhập vào các thông tin như: Project, Issue type, Summary, Description và custom field khác dựa trên loại . Thông báo nếu thất bại và để form cho user nhập lại, nếu thành công thì chuyển sang màn hình chi tiết của issue đó.



STEP 1 Khi user click nút create, client gởi yêu cầu lấy danh sách toàn bộ project và create\_sreen\_view tương mà user đó có quyền truy cập, sau đó tùy vào việc user chọn project nào thì sẽ hiện view tương ứng để user điền thông tin vào các field của issue đó:

* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | Token | VARCHAR(150) | Token đã được user đăng nhập thành công trước đó |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |
| 3 | data | MAX | Dữ liệu danh sách toàn bộ các project |

* Yêu cầu front end: gởi reuqest yêu cầu danh sách các project và create\_sreen\_view, sau đó hiển thị form dựa trên project (drop down list) và field của project đó.
* Yêu cầu backend xây dựng API như sau: kiểm tra token và trả về toàn bộ danh sách các project + screen tương ứng mà user đó có quyền.

STEP 2:

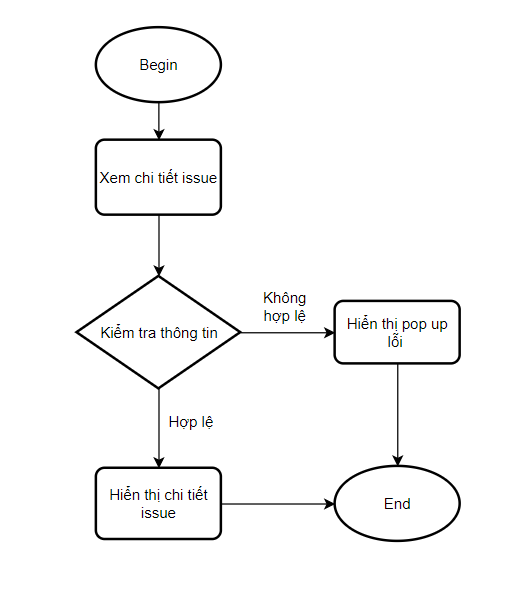
* Sau khi user đã fill toàn bộ thông tin vào các field, nhấn nút create, client gởi yêu cầu tạo issue về backend để xử lí

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | project\_key | VARCHAR2(3) | Mã của project |
| 2 | token | VARCHAR2(150) | Token của user đã đăng nhập trước đó |
| 3 | field\_name1 | VARCHAR2(n) | Giá trị của field 1 |
| 4 | field\_name2 | VARCHAR2(n) | Giá trị của field 2 |
| 5 | field\_namen | VARCHAR2(n) | Giá trị của field n |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |

* Yêu cầu front end: gởi reuqest yêu cầu tạo issue đến máy chủ, chứa tất cả các thông tin mà user đã fill ở trên. Nếu thành công thì chuyển sang màn hình view của issue đó
* Yêu cầu backend xây dựng API như sau: kiểm tra các thông tin của dữ liệu input, nếu hợp lệ, tiến hành tạo issue và phản hồi về client.

1. **Xem chi tiết issue**

* Mô tả: Người dùng truy cập màn hình xem chi tiết issue, hệ thống hiển thị các thông tin của issue đó



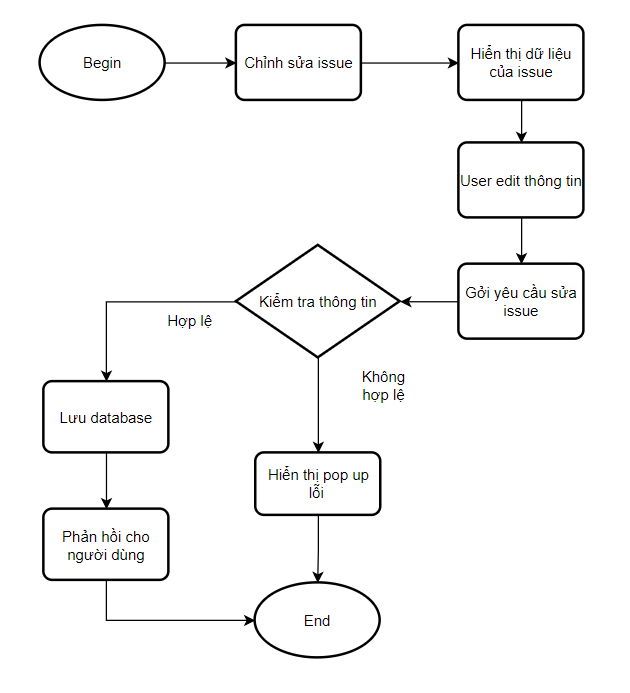
* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | issue\_id | VARCHAR2(15) | id của issue |
| 2 | project\_key | VARCHAR2(3) | key của project |
| 3 | token | VARCHAR2(150) | token của user |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |
| 3 | data | MAX | Dữ liệu các trường và giá trị của issue |

* Yêu cầu front end: sau khi người dùng click vào 1 issue cụ thể, tiến hành gọi request để lấy thông tin issue đó, hiển thị thông tin issue dựa trên data trả về.
* Yêu cầu backend xây dựng API như sau: Kiểm tra thông tin từ client, lấy dữ liệu từ các bảng phù hợp để trả về cho client.

1. **Chỉnh sửa issue:**

* Mô tả: Người dùng truy cập màn hình xem chi tiết issue, hệ thống hiển thị các thông tin của issue đó -> người dùng click nút edit, màn hình show ra pop up để user chỉnh sửa thông tin.



**STEP 1: Lấy dữ liệu của issue hiện tại**

* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | issue\_id | VARCHAR2(15) | id của issue |
| 2 | project\_key | VARCHAR2(3) | key của project |
| 3 | token | VARCHAR2(150) | token của user |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |
| 3 | data | MAX | Dữ liệu các trường và giá trị của issue |

* Yêu cầu front end: sau khi người dùng click vào edit issue, tiến hành gọi tới back end để lấy dữ liệu edit screen của issue hiện tại. Sau đó show pop up để người dùng edit.
* Yêu cầu backend xây dựng API như sau: lấy screen\_view\_edit: dựa vào dữ liệu project\_key, issue\_id để show field name và field data của issue đó.

STEP 2: Update dữ liệu:

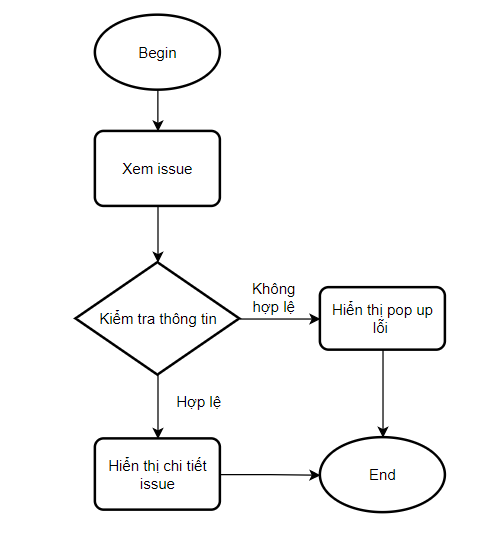
* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | issue\_id | VARCHAR2(15) | id của issue |
| 2 | project\_key | VARCHAR2(3) | key của project |
| 3 | token | VARCHAR2(150) | token của user |
| 4 | field\_name1 | VARCHAR2(n) | Giá trị của field 1 |
| 5 | field\_name2 | VARCHAR2(n) | Giá trị của field 2 |
| 6 | field\_namen | VARCHAR2(n) | Giá trị của field n |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |

* Yêu cầu front end: sau khi người dùng đã input xong thông tin mới, gọi request để update dữ liệu
* Yêu cầu backend xây dựng API như sau update field dựa trên dữ liệu client gởi lên.

1. **Xem chi tiết issue**

* Mô tả: Người dùng truy cập màn hình xem chi tiết issue, hệ thống hiển thị các thông tin của issue đó



* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | issue\_id | VARCHAR2(15) | id của issue |
| 2 | project\_key | VARCHAR2(3) | key của project |
| 3 | token | VARCHAR2(150) | token của user |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |
| 3 | data | MAX | Dữ các trường và giá trị |

* Yêu cầu front end: sau khi người dùng click vào 1 issue cụ thể, tiến hành gọi request để lấy thông tin issue đó, hiển thị thông tin issue dựa trên data trả về.
* Yêu cầu backend xây dựng API như sau: Kiểm tra thông tin từ client, lấy dữ liệu từ các bảng phù hợp để trả về cho client.

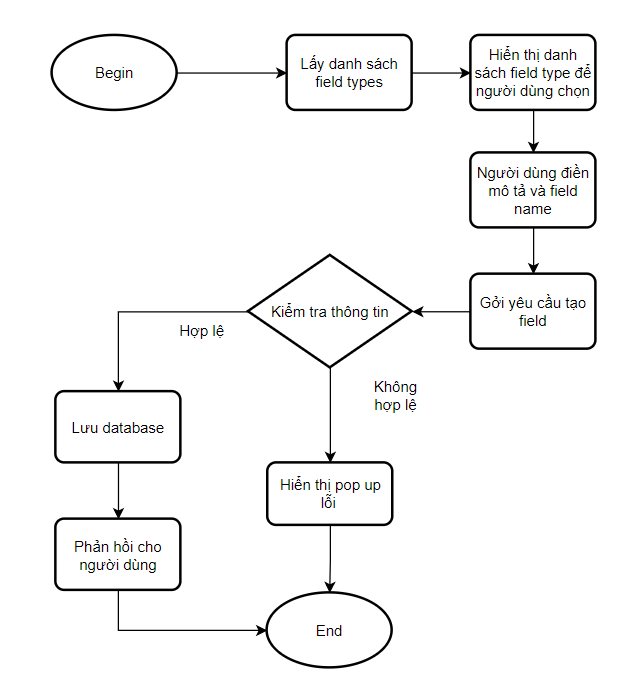
1. **Xem danh sách custom field**

* Mô tả: Người dùng truy cập màn hình xem danh sách custom field, hệ thống hiển thị toàn bộ các custom field hiện có và các screen mà custom field đang được áp dụng
* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | token | VARCHAR2(150) | token của user |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |
| 3 | data | MAX | Dữ các trường và giá trị |

* Yêu cầu front end: sau khi người dùng click vào chức năng xem danh sách custom field, client gởi request đến server để lấy danh sách custom field. Hiển thị dữ liệu dưới dạng danh sách gồm các cột: Field name, Field type, Screens. Hiển thị thêm nút chức năng tạo mới field trên màn hình
* Yêu cầu backend xây dựng API như sau: Kiểm tra thông tin từ client, trả dữ liệu danh sách gồm tất cả các field và các screen đang được áp dụng field đó.

1. **Tạo mới một field (admin)**

* Mô tả: Người dùng click nút tạo field ở màn hình xem danh sách custom field, hệ thống hiển thị pop up cùng các loại field đang được hỗ trợ, người dùng chọn loại field mình muốn thêm, sau đó điền các thông tin như tên field, mô tả,… sau đó chọn nút create để tạo field, sau khi tạo thành công thì đóng pop up và refresh lại danh sách các field, nếu tạo không thành công thì show lỗi ra pop up để người dùng input thông tin lại. 

STEP 1: lấy danh sách field types:

* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | token | VARCHAR2(150) | token của user |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |
| 3 | data | MAX | Danh sách các field type mà hệ thống hỗ trợ |

* Yêu cầu front end: Sau khi người dùng click vào chức năng tạo mới một field, gọi để lấy danh sách toàn bộ các field tpye mà hệ thống đang hỗ trợ, hiển thị để người dùng chọn field phù hợp. Hiển thị các hộp thoại để người dùng thêm field name, description
* Yêu cầu backend xây dựng API như sau: Kiểm tra thông tin từ client, trả dữ liệu danh sách gồm tất cả các loại field đang được hỗ trợ

STEP 2: Tạo field dựa trên field type:

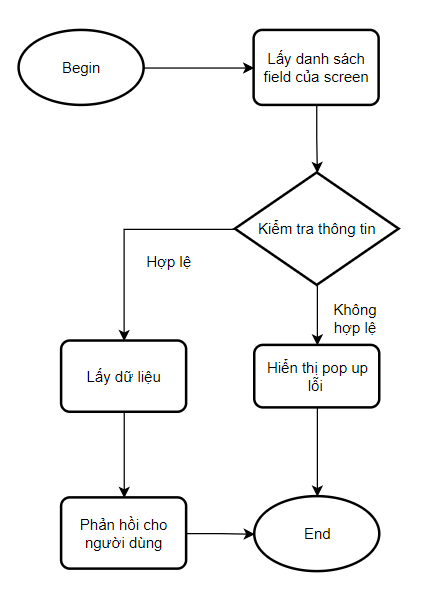
* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | token | VARCHAR2(150) | token của user |
| 2 | field\_type | VARCHAR2(10) | loại field mà người dùng muốn tạo |
| 3 | field\_name | VARCHAR2(50) | tên field |
| 4 | description | VARCHAR2(255) | mô tả field |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |

* Yêu cầu front end: Sau khi người dùng điền đầy đủ thông tin, gọi đên backend để lưu thông tin field
* Yêu cầu backend xây dựng API như sau: Kiểm tra thông tin từ client, lưu thông tin field và phản hồi về cho client

1. **Xem danh sách screen (admin)**

* Mô tả: Người dùng truy cập màn hình xem danh sách screen, hệ thống hiển thị toàn bộ các screen hiện có và các project có sử dụng screen đó.



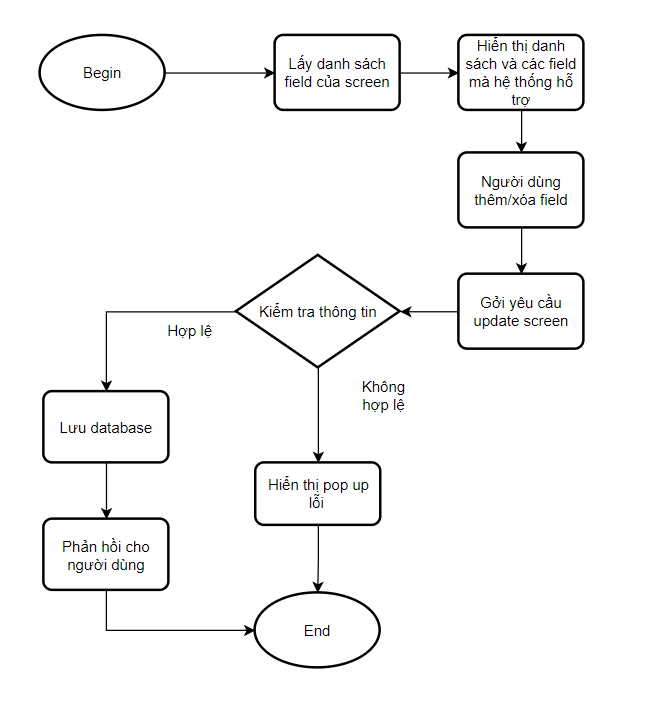
* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | token | VARCHAR2(150) | token của user |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |
| 3 | data | MAX | Dữ các trường và giá trị |

* Yêu cầu front end: sau khi người dùng click vào chức năng xem danh sách screen, client gởi request đến server để lấy danh sách screen. Hiển thị dữ liệu dưới dạng danh sách gồm các cột: screen name, project, các button edit, delete.
* Yêu cầu backend xây dựng API như sau: Kiểm tra thông tin từ client, trả dữ liệu danh sách gồm tất cả các screen và các project đang được áp dụng screen đó.

1. **Liên kết field và screen (admin)**

* Mô tả: Người dùng click nút edit screen ở màn hình xem danh sách screen. Hệ thống hiện thị mành hình liệt kệ tất cả các field đang có của screen, màn hình cho phép xóa các field đang tồn tại và thêm mới các field khác. Sau khi người dùng thêm/xóa field và đồng ý lưu lại. Hệ thống sẽ lưu thông tin và refresh lại trang hiện thị của screen đó



STEP 1: lấy danh sách field của screen:

* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | token | VARCHAR2(150) | token của user |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |
| 3 | data | MAX | Danh sách các field type mà hệ thống hỗ trợ |

* Yêu cầu front end: Gọi đến API để lấy thông tin các field hiện có của screen, các field mà hệ thống hỗ trợ. Sau đó liệt kê các field dưới dạng danh sách, kèm nút delete cho mỗi field đó, hiện một drop down list chứa danh sách các field mà hệ thống đã có (loại trừ những field mà screen đã có rồi) để người dùng có thể chọn và add vào screen
* Yêu cầu backend xây dựng API như sau: Kiểm tra thông tin từ client, trả dữ liệu danh sách gồm tất cả các field của screen được chỉ định và full field của hệ thống đang có.

STEP 2: Update dữ liệu: Sau khi người dùng đã thêm/xóa (cập nhật các field của screen đó xong) và nhấn save.

* Dữ liệu gởi lên và nhận về:

|  |  |  |  |
| --- | --- | --- | --- |
| Input |  |  |  |
| 1 | token | VARCHAR2(150) | token của user |
| 2 | field\_id1 | VARCHAR2(10) | field id 1 |
| 3 | field\_id2 | VARCHAR2(10) | field id 2 |
| 4 | field\_idn | VARCHAR2(10)) | field id n |
| Output |  |  |  |
| 1 | result\_code | VARCHAR(5) | Mã lỗi |
| 2 | description | VARCHAR(255) | Mô tả |

* Yêu cầu front end: Sau khi người dùng thêm/xóa field xong, tiến hành gởi lại danh sách các field của screen đó để backend update. Khi thành công thì làm mới danh sách field của screen đó
* Yêu cầu backend xây dựng API như sau: Kiểm tra thông tin từ client, lưu thông tin của screen đó và phản hồi về cho client.