[**Đăng ký User mới**](#_mmwzz8ro8b58) **2**

[**Đăng nhập User**](#_u6t6p3tyabk9) **4**

[**Xem thông tin User**](#_pcj9jr2chisc) **6**

[**Sửa thông tin User**](#_lrrttcj9vbfw) **9**

[**Tạo ISSUE (all users)**](#_dy4u4yra098o) **12**

[UI tạo Issue](#_rdayp1c4wq15) 12

[nút “Create” trên thanh topbar](#_5tfsbbjep4oa) 12

[screen “Create issue”](#_1u8wki2ipsgn) 12

[Popup thành công](#_h6os4duyguns) 13

[Đặc tả tạo Issue](#_x042hbg8ivn) 13

[**Xem danh sách ISSUES**](#_p4z4rsewtpnk) **16**

[UI xem danh sách Issues](#_gn4ugsjyw60v) 16

[nút “Issues” trên thanh sidebar](#_vcp56olqwp8l) 16

[screen “Issues”](#_mzq0chpgn00q) 16

[Đặc tả xem danh sách Issues](#_wpznr7d7ue0c) 17

[**Xem ISSUE**](#_icfbbjsrv8ni) **18**

[UI xem Issue](#_i7kf0yrv19sg) 18

[screen “Issues”](#_vm9eepx1xu23) 18

[screen “Issue”](#_27vkkvt022iw) 18

[Đặc tả xem Issue](#_ao11o4gq5jb5) 18

[**Chỉnh ISSUE**](#_4cs57cnih1lw) **20**

[UI chỉnh Issue](#_ije7xc2tkv29) 20

[screen “Issue”: tạo thêm nút Edit trên screen Issue](#_kd5bq25vwhiv) 20

[screen “Edit issue”](#_bvqqi02ny03v) 20

[Popup thành công](#_oh4jppylm615) 22

[Đặc tả chỉnh Issue](#_39w6qabpinja) 22

[**Xóa ISSUE**](#_fdk9fjyhdunf) **25**

[UI xóa Issue](#_fp5dp1bhit09) 25

[screen “Issue”: tạo thêm nút “Delete” trên screen Issue](#_9sn8in85zq2n) 25

[modal “Delete Issue”](#_c8nryhch7izf) 25

[Đặc tả xóa Issue](#_xkwko3okrcjl) 25

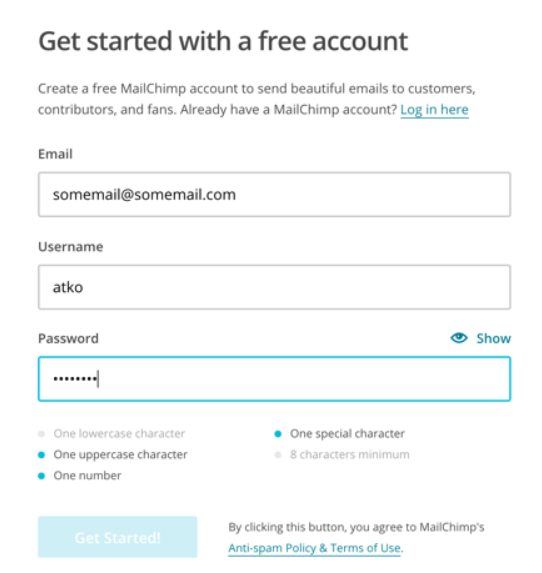
LUỒNG CƠ BẢN:

quản lí User -> quản lí Project -> Định nghĩa Workflow -> quản lí Issue

# Đăng ký User mới

1.1. Ui đăng ký user.

Form gồm các input: Email, UserName,Full-Name, Password, Re-Password



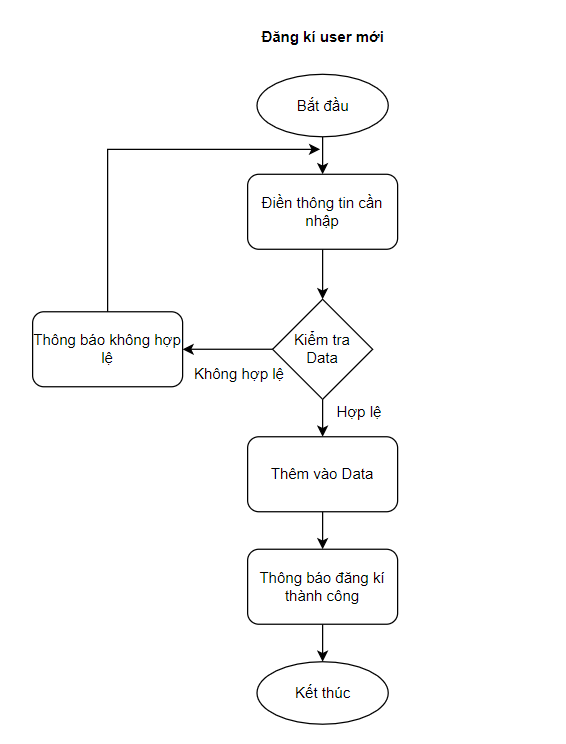
1.2) Đặc tả đăng ký User.

Mô tả: Khi User đăng kí 1 tài khoản mới thì nhập vào form đăng ký những thông tin cần thiết.

server kiểm tra, nếu username hoặc email đã tồn tại thì thông báo không hợp lệ. Còn

không thì tiến hành mã hóa password và lưu thông tin vào database trả kết trả cho

client là thành công và vào trang chủ đồng thời lưu token mà server gửi lên.



| **INPUT** | | | |
| --- | --- | --- | --- |
| 1 | email\_address | varchar(150) | email của user |
| 2 | full\_name | varchar(150) | tên của user |
| 3 | user\_name | varchar(150) | username của user |
| 4 | password | varchar(150) | password của user |
| **Output** | | | |
| 1 | code | varchar(5) | Mã lỗi |
| 2 | description | varchar(250) | Mô tả |
| 3 | token | varchar(150) | token của user |

Frontend:

* Tạo form để user nhập input thông tin.
* Kiểm tra email có hợp lệ không nếu không thông báo cho người dùng nhập lại.
* Kiểm tra re-Password với password có giống nhau không nếu không thông báo người dùng nhập lại.
* Gửi thông tin lên server qua phương thức post với dữ liệu body.

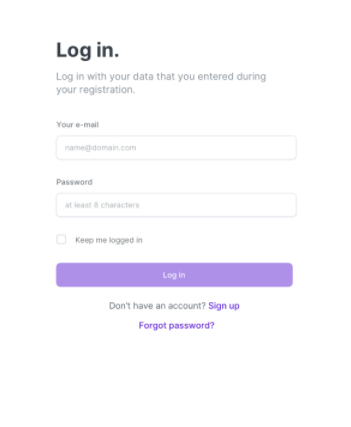
Backend:

* Backend kiểm tra các dữ liệu input, kiểm tra sự tồn tại của username và email. Nếu không hợp lệ gửi phản hồi về client.
* Nếu hợp lệ tiến hành băm mật khẩu sau đó lưu thông tin vào database.
* Sau đó tạo token gửi thông báo thành công đính kèm với token của người dùng

# Đăng nhập User

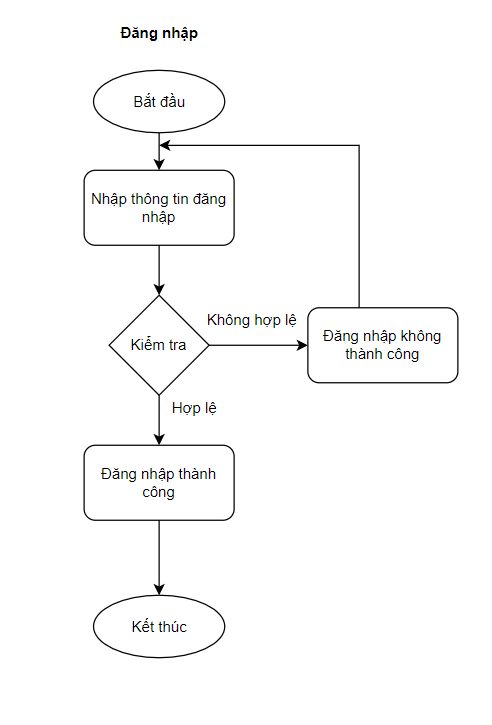
2.1) UI đăng nhập.

Form đăng nhập bao gồm username/email và password



2.2) Đặc tả đăng nhập.

Yêu cầu người dùng nhập username hoặc email và password. Hệ thống sẽ tiến hành băm mật khẩu và kiểm tra username/email của user đó. Nếu không phù hợp thông báo cho user và chuyển qua màn hình khác nếu phù hợp.



| **INPUT** | | | |
| --- | --- | --- | --- |
| 1 | email/username | varchar(150) | email hoặc username của user |
| 2 | password | varchar(150) | mật khẩu của user |
| **Output** | | | |
| 1 | code | varchar(5) | Mã lỗi |
| 2 | description | varchar(250) | Mô tả |
| 3 | token | varchar(150) | token của user |

Frontend:

* Tạo form để user nhập input thông tin.
* Gửi thông tin lên server qua phương thức post với dữ liệu body.
* Lưu token vào session nếu nhận được.

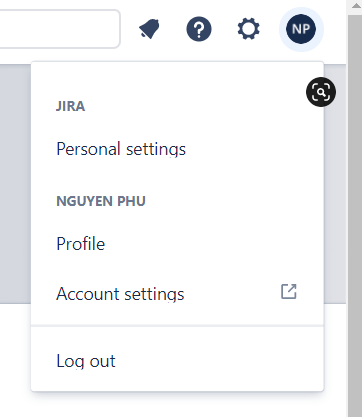
Backend:

* Backend kiểm tra các dữ liệu input, kiểm tra sự tồn tại của username hoặc email. Nếu không hợp lệ gửi phản hồi về client.
* Nếu hợp lệ tiến hành băm mật khẩu sau đó kiểm tra trùng nếu mật khẩu không chính xác phản hồi lên client. Nếu mật khẩu chính xác chuyển qua bước sau.
* Sau đó tạo token gửi thông báo thành công đính kèm với token của người dùng

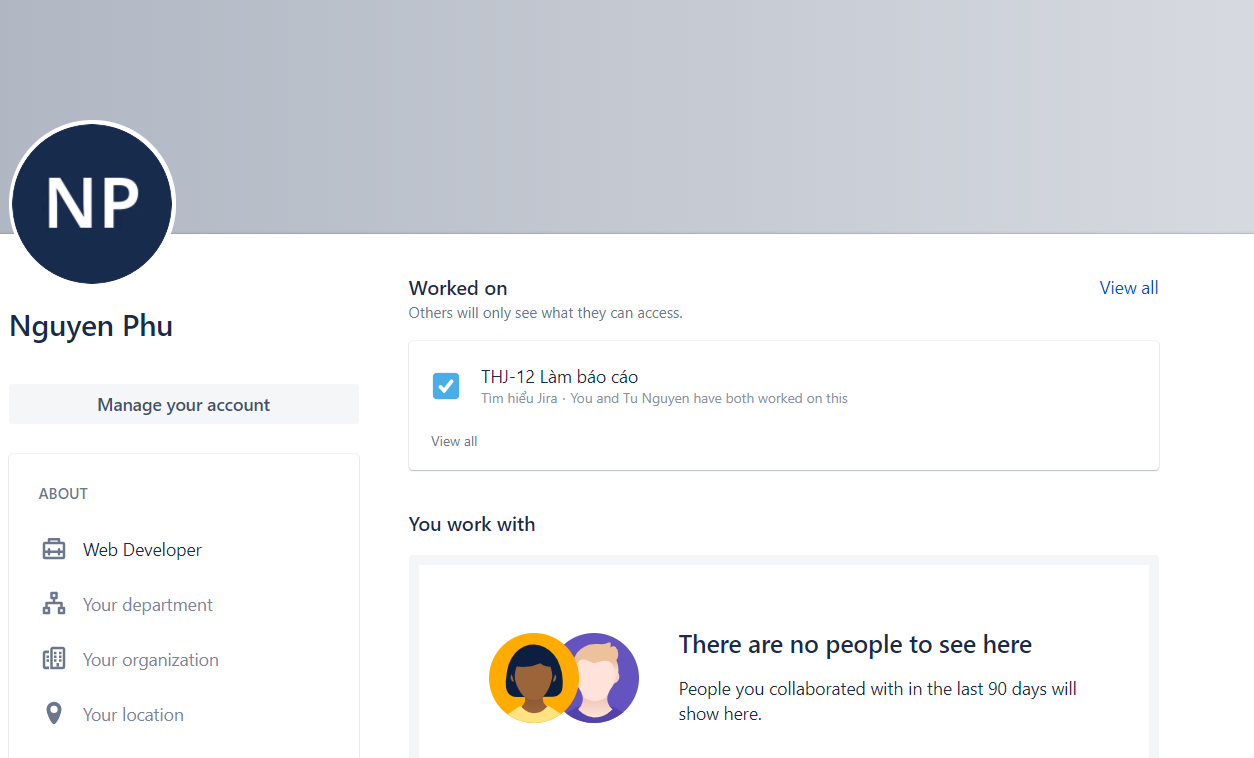
# Xem thông tin User

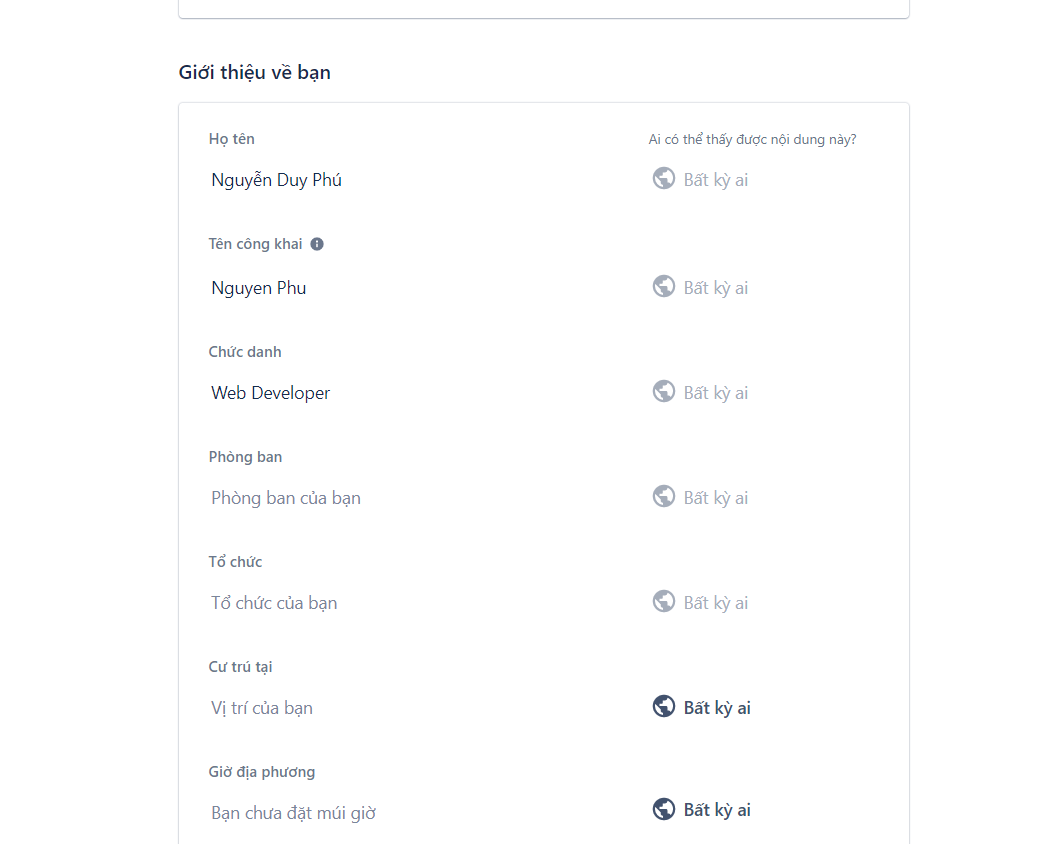
3.1: UI xem thông tin người dùng

UI: Nhấn vào Profile



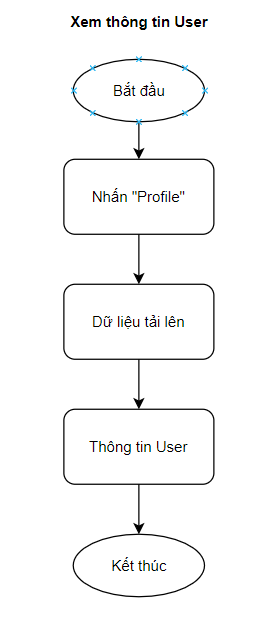
screen profile user





3.2) Đặc tả xem thông tin User.

Khi người dùng click “profile” thì sẽ chuyển sang trang thông tin người dùng. Người dùng sẽ xem được những thông tin của mình ở đó.



| **INPUT** | | | |
| --- | --- | --- | --- |
| 1 | token | varchar(150) | token của người dùng |
| **Output** | | | |
| 1 | code | varchar(5) | Mã lỗi |
| 2 | description | varchar(250) | Mô tả |
| 3 | data | varchar(150) | thông tin người dùng |

Frontend:

* Gửi token lên server.
* Xem thông tin người dùng

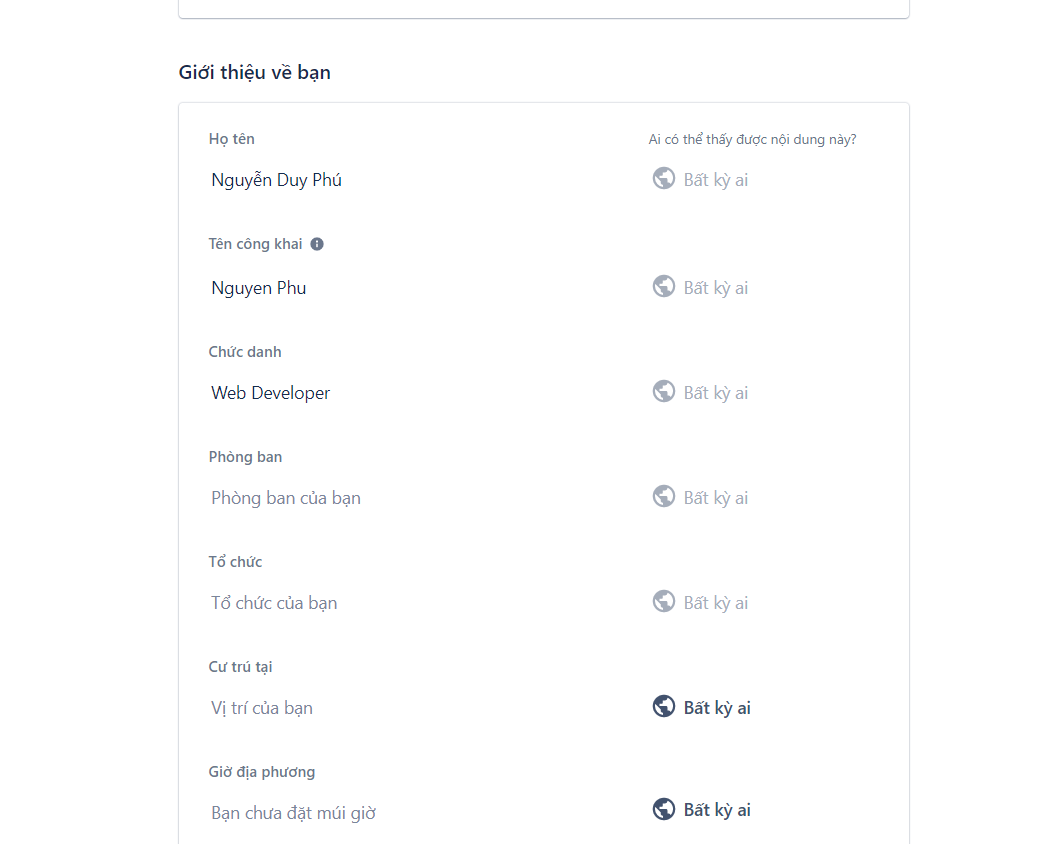
Backend:

* Kiểm tra token của người dùng tiến hành băm token đó nếu không hợp lệ gửi lên client ngược lại nếu token hợp lệ chuyển qua bước sau.
* Nếu hợp lệ tiến hành kiểm tra data và lấy thông tin người dùng gửi lên client.

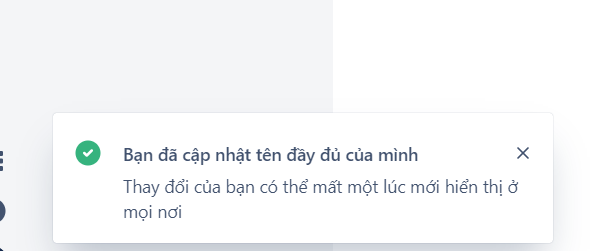
# Sửa thông tin User

4.1: UI xem thông tin người dùng

Nhấn vào “Manage your account”

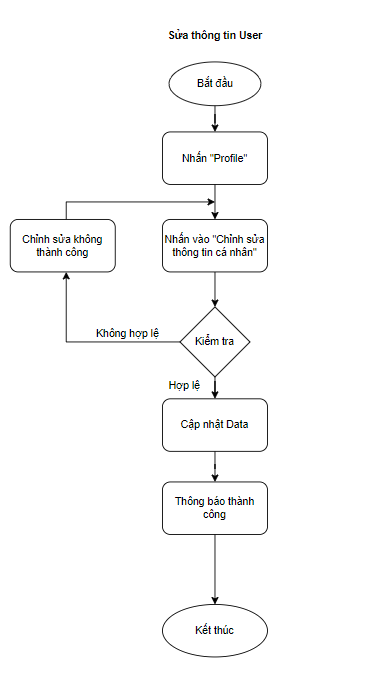


Chỉnh sửa thành công:



3.2) Đặc tả chỉnh sửa thông tin User.

Khi người dùng muốn thay đổi những thông tin mà mình muốn nhấn vào những fields đó để thay đổi.



| **INPUT** | | | |
| --- | --- | --- | --- |
| 1 | token | varchar(150) | token của người dùng |
| 2 | full\_name | varchar(150) | tên người dùng |
| **Output** | | | |
| 1 | code | varchar(5) | Mã lỗi |
| 2 | description | varchar(250) | Mô tả |
| 3 | data | varchar(150) | thông tin người dùng |

Frontend:

* Gửi token lên server.
* Gửi thông tin người đó muốn thay đổi qua server qua phương thức post

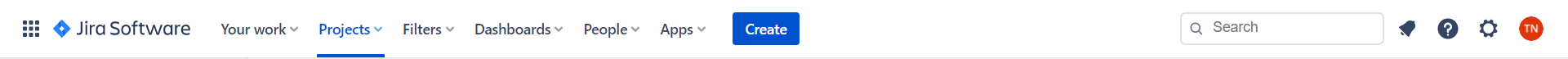
Backend:

* Kiểm tra token của người dùng tiến hành băm token đó nếu không hợp lệ gửi lên client ngược lại nếu token hợp lệ chuyển qua bước sau.
* Nếu hợp lệ tiến hành cập nhật lại data của người dùng đó.

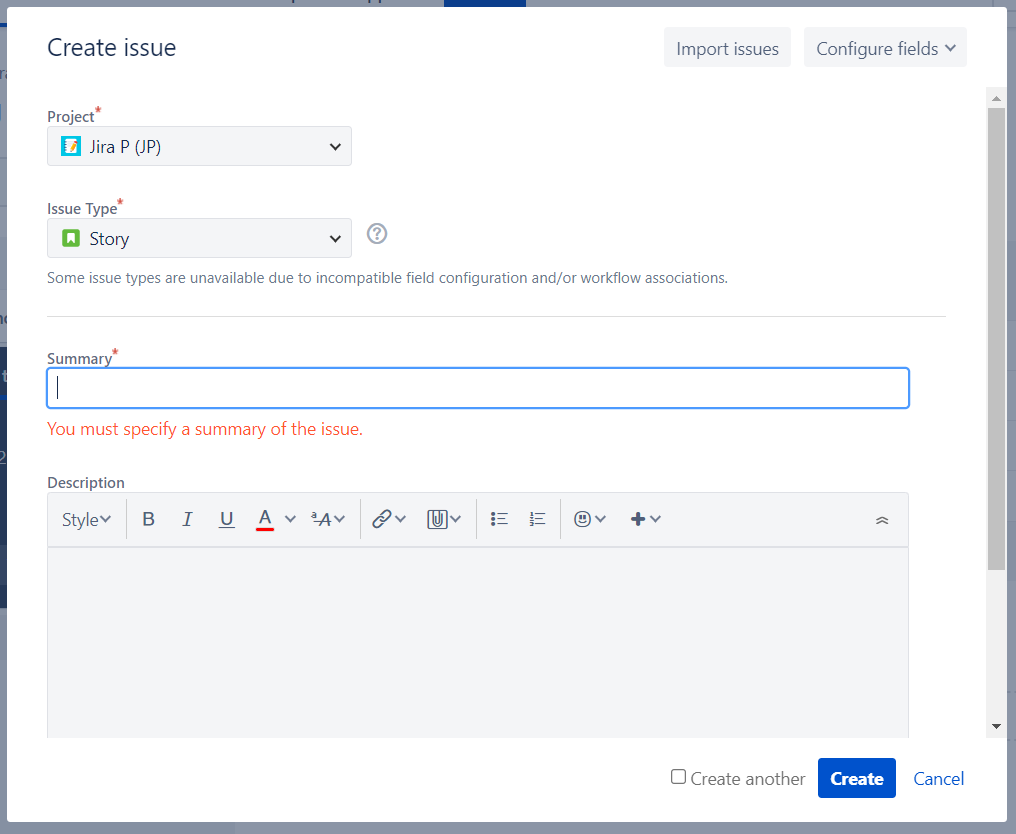
# Tạo ISSUE (all users)

## UI tạo Issue

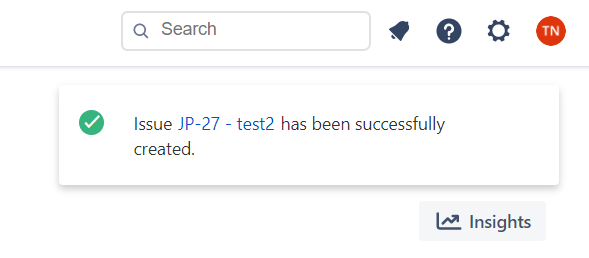
### nút “Create” trên thanh topbar



### screen “Create issue”



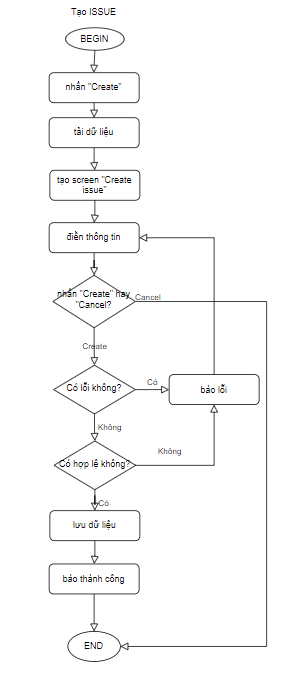
### Popup thành công



## Đặc tả tạo Issue

Mô tả: user muốn tạo một Issue mới

User nhấn vào nút “Create” trên thanh topbar. Chương trình sẽ tạo một screen modal cung cấp các fields cần thiết để tạo Issue. Sau đó, user nhập các fields cần thiết để tạo. Nếu hợp lệ, tạo một Issue mới và gửi popup thành công. Nếu không hợp lệ, báo lỗi và yêu cầu user nhập lại field bị lỗi.



Bước 1: Khi User click vào nút “Create”, client gửi request lấy data liên quan dựa trên các fields cần khởi tạo (danh sách Projects, danh sách Users, danh sách Issues …), sau đó client hiện modal screen tạo Issue chờ user nhập

| **INPUT** | | | |
| --- | --- | --- | --- |
| 1 | Token | varchar(150) | token của user đã đăng nhập thành công trước đó |
| **Output** | | | |
| 1 | code | varchar(5) | Mã lỗi |
| 2 | description | varchar(250) | Mô tả |
| 3 | data | Max | data liên quan đến các fields tạo Issue |

Frontend:

* Tạo nút create trên thanh topbar
* Gửi request lấy các field cần thiết để tạo modal screen “Create issue”
* Tạo modal screen “Create issue”

Backend:

* Kiểm tra token
* response data

Bước 2: Sau khi User nhập đầy đủ các fields bắt buộc và nhấn nút “Create”, thì Client sẽ kiểm tra thông tin nhập có hợp lệ không? Nếu không hợp lệ báo lỗi và chờ user nhập lại. Nếu hợp lệ, client sẽ gửi request tạo Issue tương ứng. Server kiểm tra thông tin và xử lý yêu cầu, lưu vào database và response về cho client. Client hiển thị popup thành công.

| **INPUT** | | | |
| --- | --- | --- | --- |
| 1 | Token | varchar(150) | token của user đã đăng nhập thành công trước đó |
| 2 | data | Max | các fields dùng để tạo Issue |
| **Output** | | | |
| 1 | code | varchar(5) | Mã lỗi |
| 2 | description | varchar(250) | Mô tả |

Frontend:

* Kiểm tra thông tin ở modal screen “Create issue” có hợp lệ?
* request tạo Issue
* Tạo popup thành công

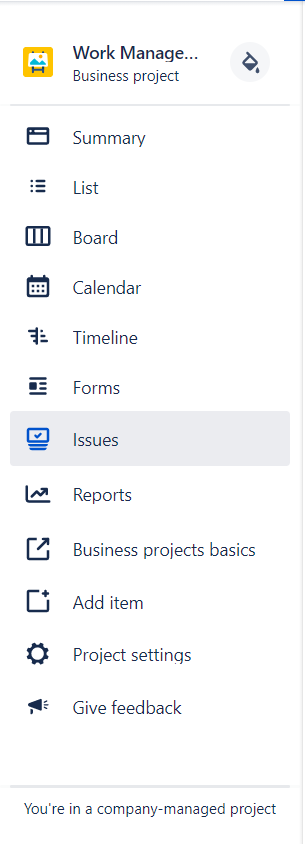
Backend:

* Kiểm tra token
* Kiểm tra data có hợp lệ và tạo Issue
* phản hồi client

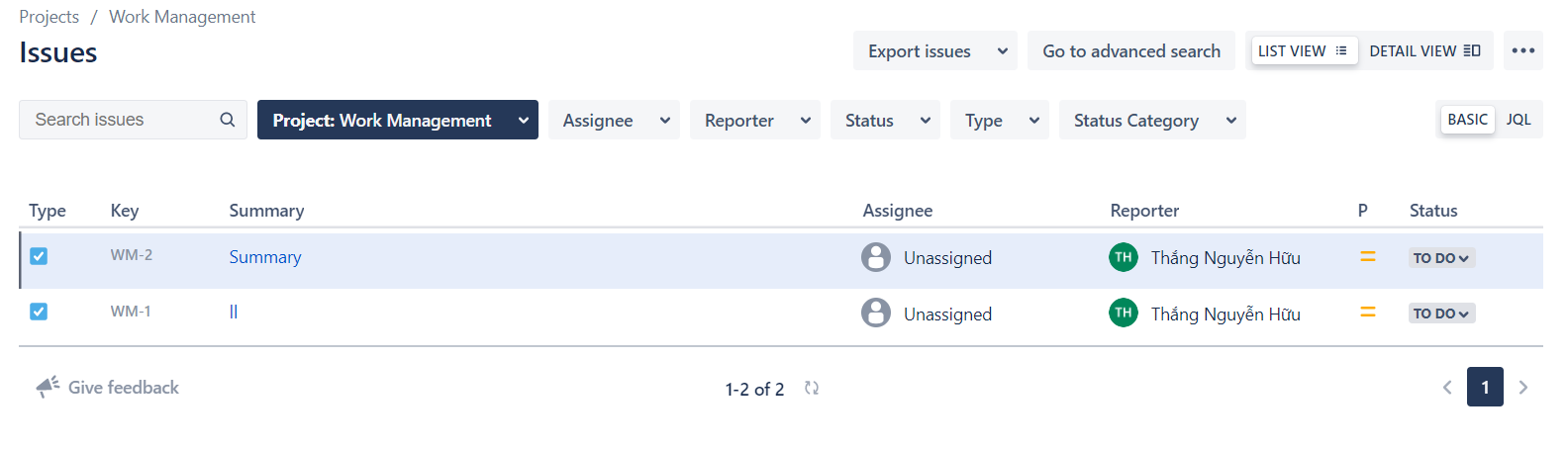
# Xem danh sách ISSUES

## UI xem danh sách Issues

### nút “Issues” trên thanh sidebar



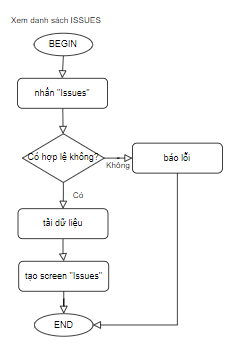
### screen “Issues”



## Đặc tả xem danh sách Issues

Mô tả: user muốn xem danh sách Issues của Project hiện tại

User nhấn vào nút “Issues” trên thanh sidebar. Chương trình sẽ tạo một screen chứa danh sách các Issues cho user.



Bước 1: Khi User click vào nút “Issues”, client gửi request lấy danh sách Issues liên quan đến Project hiện tại mà User có quyền xem, Server kiểm tra request có hợp lệ không? Nếu hợp lệ, sẽ trả data cho client, sau đó client hiện screen chứa danh sách Issues . Nếu không hợp lệ Server sẽ trả lỗi và client hiện lỗi cho user.

| **INPUT** | | | |
| --- | --- | --- | --- |
| 1 | Token | varchar(150) | token của user đã đăng nhập thành công trước đó |
| 2 | issue\_project\_id | int | project id hiện tại cần lấy danh sách Issues |
| **Output** | | | |
| 1 | code | varchar(5) | Mã lỗi |
| 2 | description | varchar(250) | Mô tả |
| 3 | data | Max | danh sách Issues |

Frontend:

* Tạo nút “Issues” trên thanh sidebar
* Gửi request lấy danh sách Issues
* Hiện screen danh sách Issues
* Popup lỗi nếu có

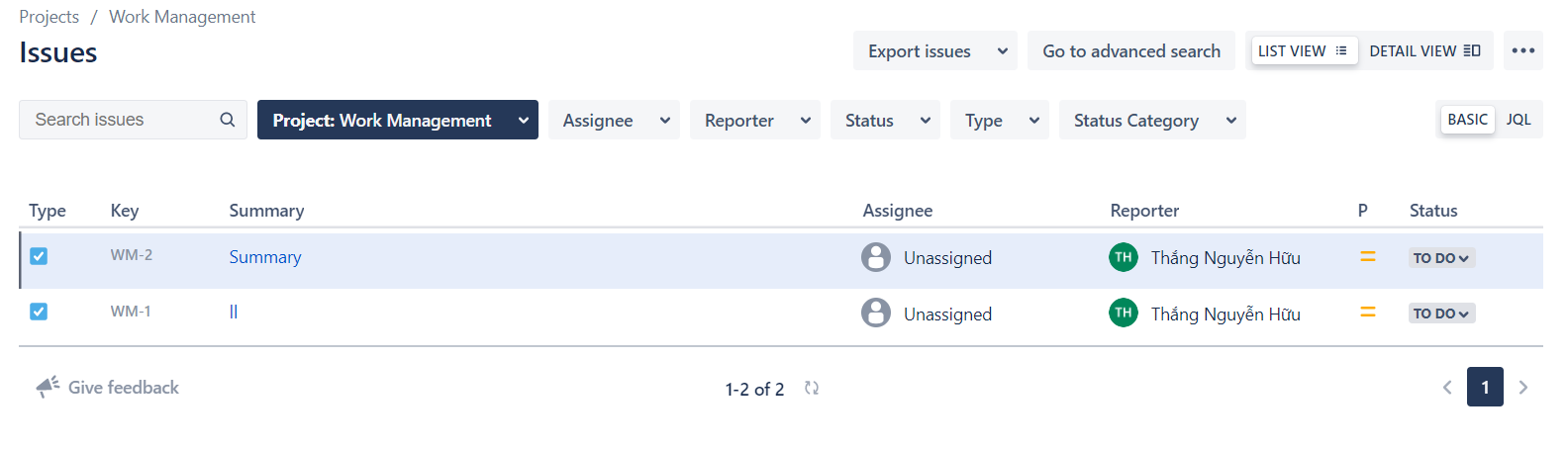
Backend:

* Kiểm tra token
* Kiểm tra data có hợp lệ và lấy danh sách Issues
* trả danh sách Issues cho Client

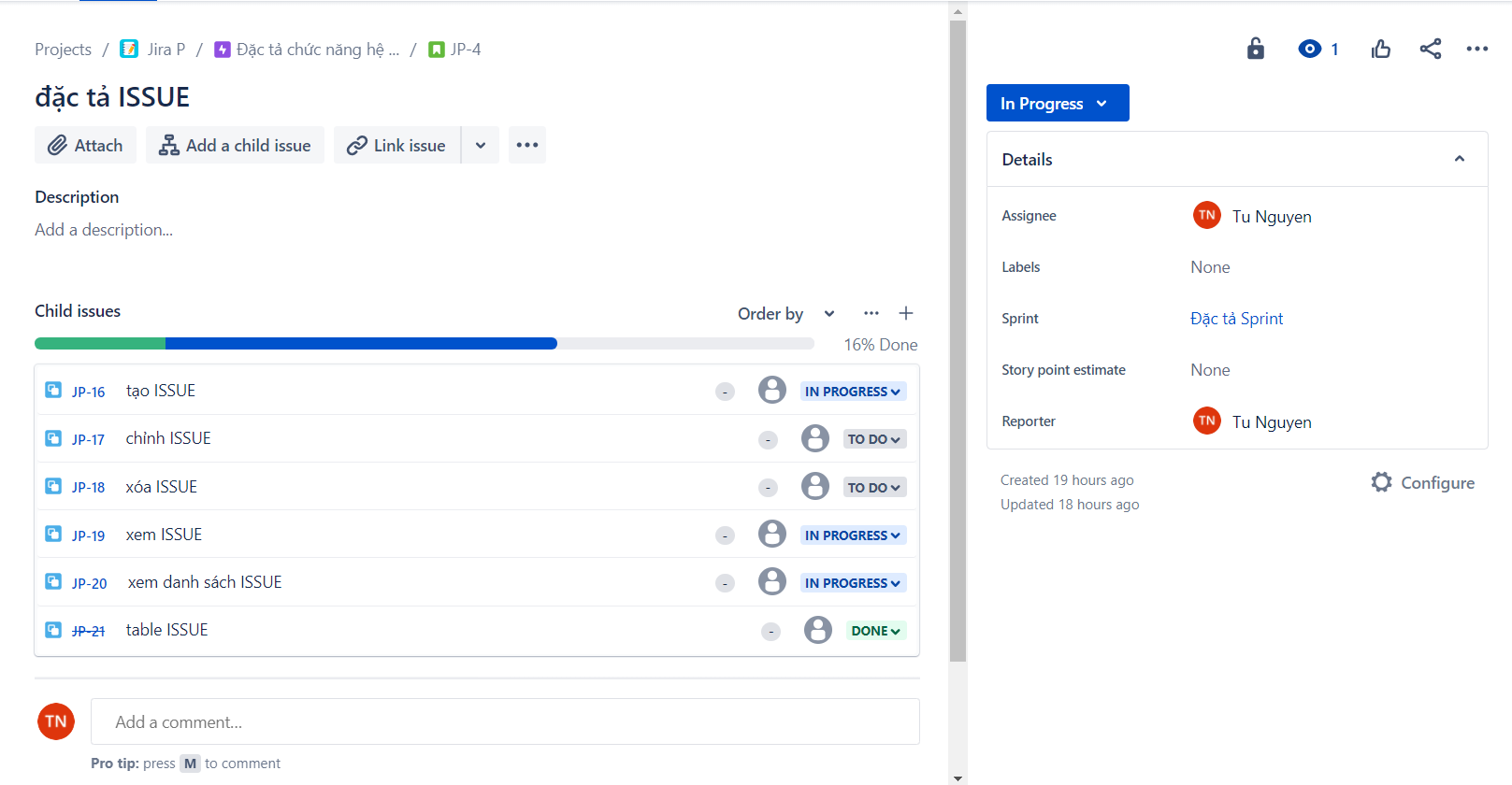
# Xem ISSUE

## UI xem Issue

### screen “Issues”



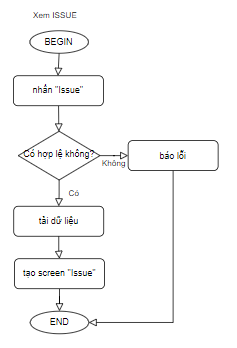
### screen “Issue”



## Đặc tả xem Issue

Mô tả: User muốn xem chi tiết một Issue

User nhấn vào một Issue bất kỳ trên screen danh sách Issues. Chương trình sẽ tạo một screen Issue chứa các thông tin cho user.



Bước 1: Khi User click vào một Issue bất kỳ, client gửi request lấy data của Issue đó, Server sẽ kiểm tra request có hợp lệ không? Nếu hợp lệ, Server sẽ trả Issue về cho client, sau đó client tạo screen Issue cho user. Nếu không hợp lệ, Server sẽ trả lỗi, và client báo lỗi cho user.

| **INPUT** | | | |
| --- | --- | --- | --- |
| 1 | Token | varchar(150) | token của user đã đăng nhập thành công trước đó |
| 2 | issue\_id | int | issue id của Issue cần xem |
| **Output** | | | |
| 1 | code | varchar(5) | Mã lỗi |
| 2 | description | varchar(250) | Mô tả |
| 3 | data | Max | data của Issue |

Frontend:

* Gửi request lấy Issue
* Tạo screen Issue
* Báo lỗi

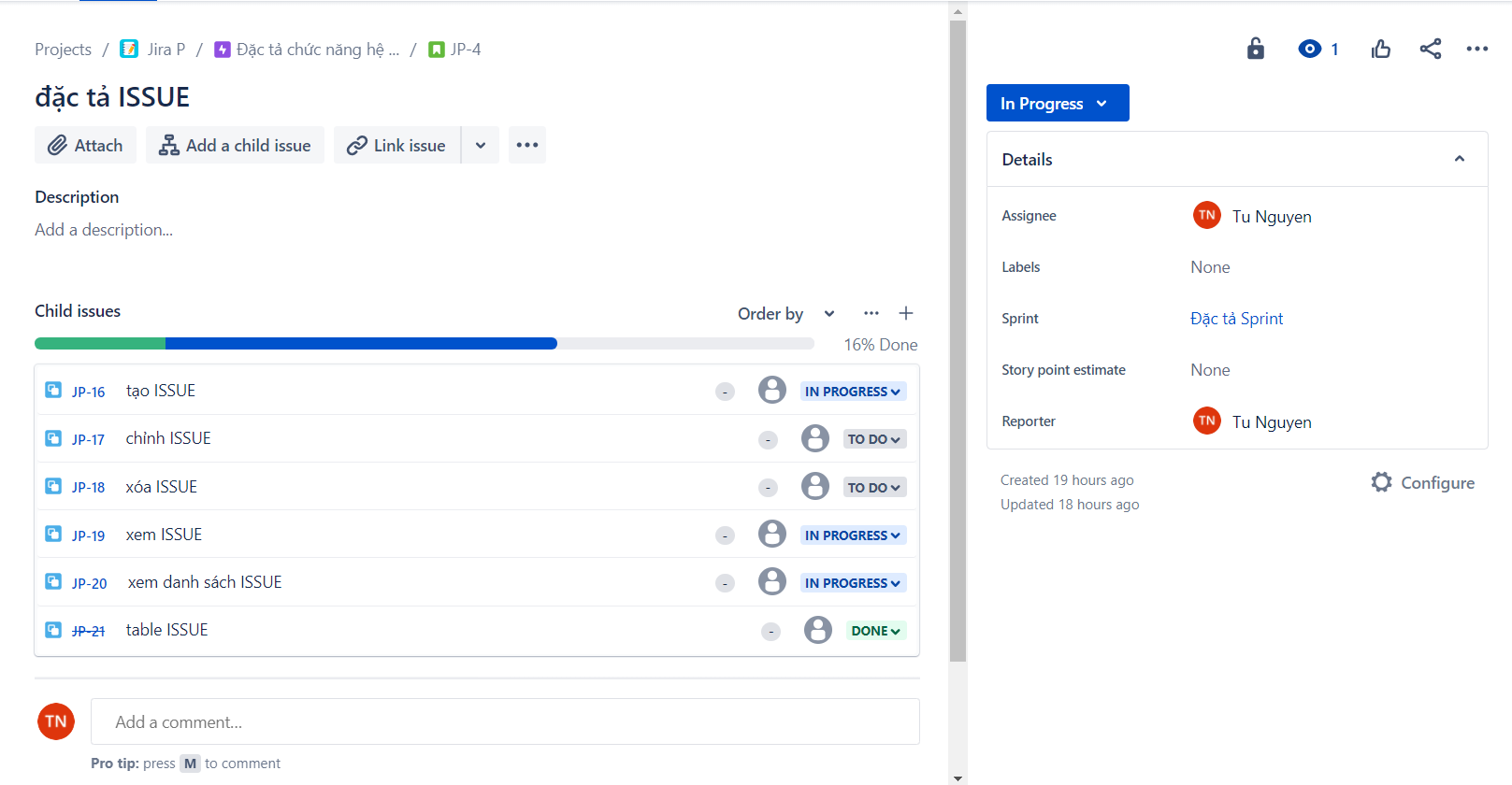
Backend:

* Kiểm tra token
* Kiểm tra request hợp lệ và lấy data Issue
* trả data về client

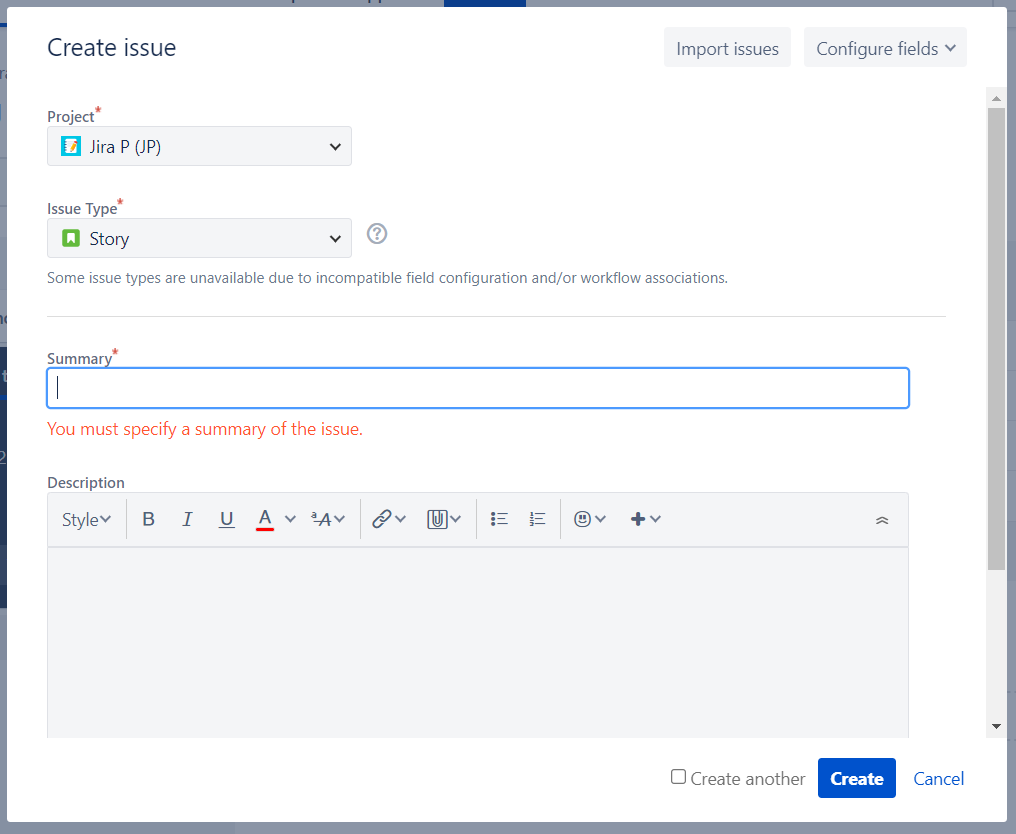
# Chỉnh ISSUE

## UI chỉnh Issue

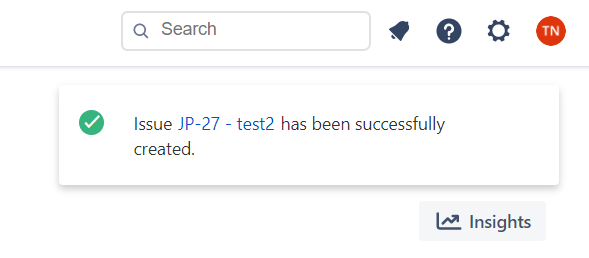
### screen “Issue”: tạo thêm nút Edit trên screen Issue



### screen “Edit issue”



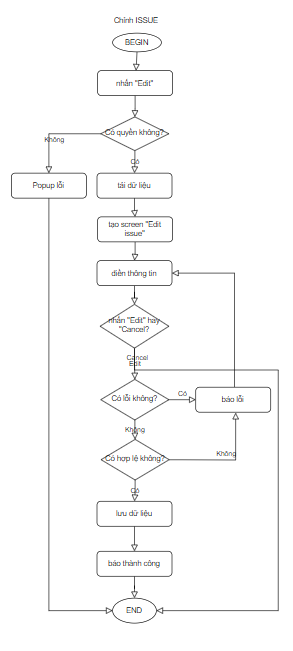
### Popup thành công



## Đặc tả chỉnh Issue

Mô tả: user muốn cập nhật một Issue

User nhấn vào nút “Edit” trong screen của một Issue. Chương trình sẽ tạo một screen modal cung cấp các fields cần thiết để update Issue. Sau đó, user nhập các fields cần thiết để tạo. Nếu hợp lệ, cập nhật Issue mới và gửi popup thành công. Nếu không hợp lệ, báo lỗi và yêu cầu user nhập lại field bị lỗi.



Bước 1: Khi User click vào nút “Edit”, client gửi request lấy data liên quan dựa trên các fields cần update (danh sách Projects, danh sách Users, danh sách Issues …), Server kiểm tra quyền chỉnh sửa Issue của User. Nếu user có quyền, Server gửi các field Issue và sau đó client hiện modal screen edit Issue chờ user nhập. Nếu user không có quyền, thì Server sẽ trả mã lỗi và client hiện popup thông báo user không có quyền.

| **INPUT** | | | |
| --- | --- | --- | --- |
| 1 | Token | varchar(150) | token của user đã đăng nhập thành công trước đó |
| 2 | issue\_id | int | issue id của Issue cần cập nhật |
| 3 | user\_id | int | user muốn chỉnh Issue |
| **Output** | | | |
| 1 | code | varchar(5) | Mã lỗi |
| 2 | description | varchar(250) | Mô tả |
| 3 | data | Max | data liên quan đến các fields cập nhật Issue |

Frontend:

* Tạo nút “Edit”
* Gửi request lấy Issue
* Tạo screen modal edit Issue
* Popup lỗi

Backend:

* Kiểm tra token
* Kiểm tra quyền chỉnh sửa của User và lấy Issue
* trả về Issue hoặc lỗi

Bước 2: Sau khi User nhập đầy đủ các fields bắt buộc và nhấn nút “Edit”, thì Client sẽ kiểm tra thông tin nhập có hợp lệ không? Nếu không hợp lệ báo lỗi và chờ user nhập lại. Nếu hợp lệ, client sẽ gửi request update Issue tương ứng. Server kiểm tra data có hợp lệ không? Nếu hợp lệ xử lý yêu cầu, lưu vào database và response về cho client. Client hiển thị popup thành công. Nếu không hợp lệ, backend trả mã lỗi và client báo lỗi cho user.

| **INPUT** | | | |
| --- | --- | --- | --- |
| 1 | Token | varchar(150) | token của user đã đăng nhập thành công trước đó |
| 2 | data | Max | các fields dùng để cập nhật Issue |
| **Output** | | | |
| 1 | code | varchar(5) | Mã lỗi |
| 2 | description | varchar(250) | Mô tả |

Frontend:

* Kiểm tra input có hợp lệ?
* gửi request cập nhật Issue
* Tạo popup thành công
* Báo lỗi

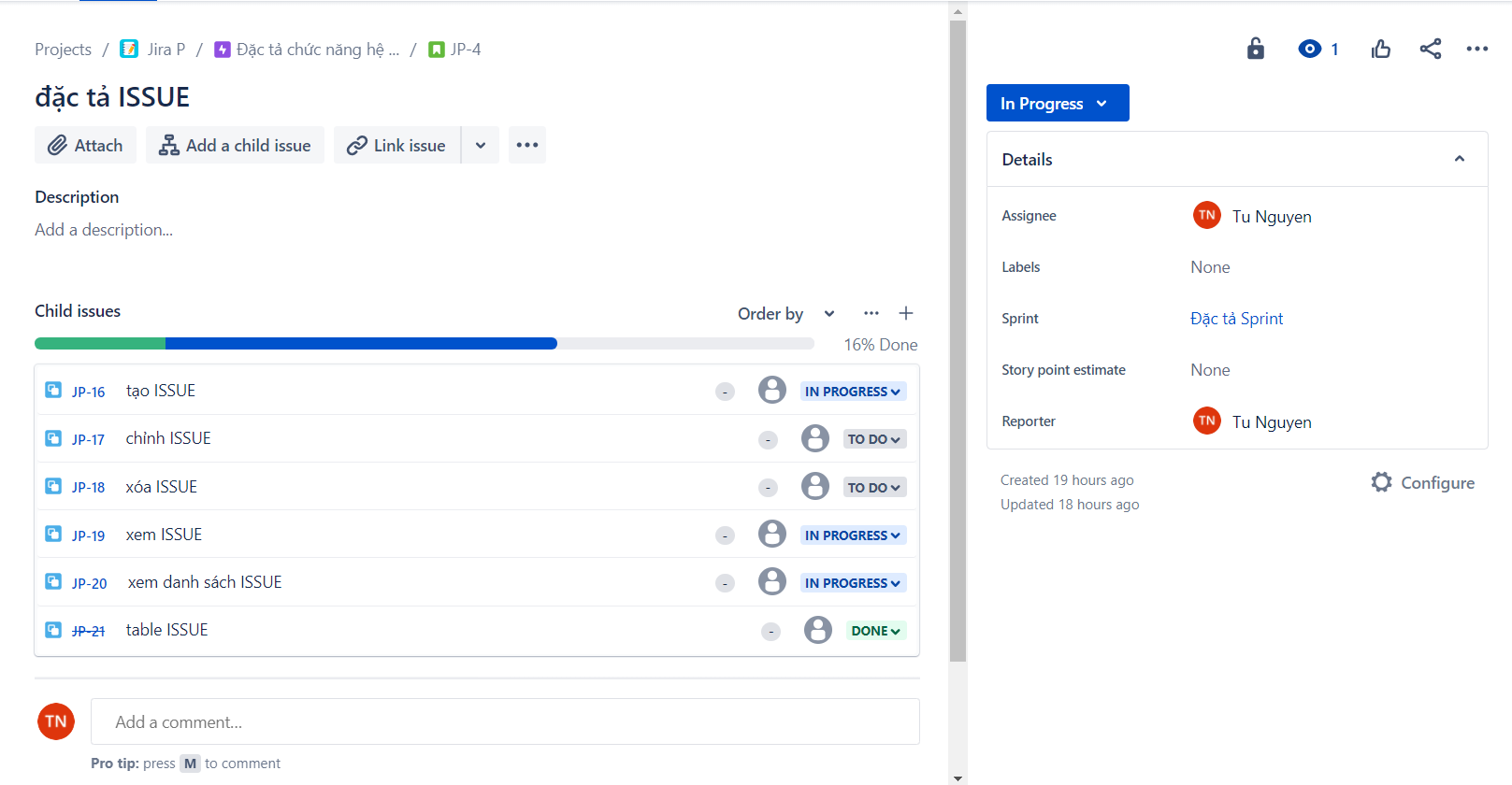
Backend:

* Kiểm tra token
* Kiểm tra data Issue có hợp lệ và cập nhật Issue
* phản hồi client

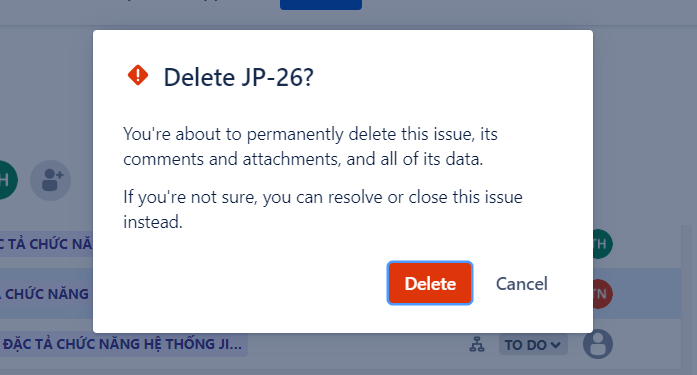
# Xóa ISSUE

## UI xóa Issue

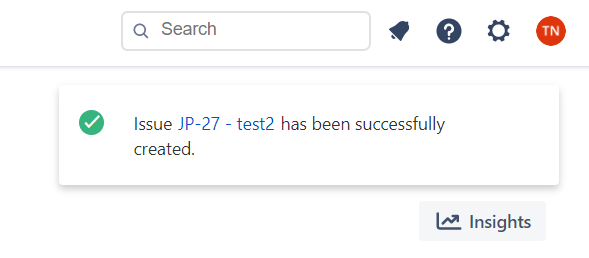
### screen “Issue”: tạo thêm nút “Delete” trên screen Issue



### modal “Delete Issue”



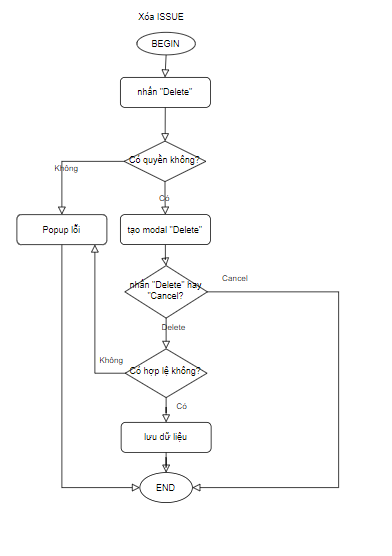
### Popup thành công



## Đặc tả xóa Issue

Mô tả: user muốn xóa một Issue

User nhấn vào nút “Delete” trong screen của một Issue. Nếu user có quyền xóa Issue đó thì chương trình sẽ tạo một screen modal chờ user xác nhận. Sau đó, user nhấn vào nút mong muốn. Nếu là nút Delete, xóa Issue. Nếu là nút Cancel, quay lại screen trước đó. Nếu user không có quyền thì client sẽ popup lỗi user không có quyền xóa Issue.



Bước 1: Khi User click vào nút “Delete”, client gửi request kiểm tra quyền xóa Issue của user. Nếu user có quyền, Server gửi mã code thành công và sau đó client hiện modal screen delete Issue chờ user nhập. Nếu user không có quyền, thì Server sẽ trả mã lỗi và client hiện popup thông báo user không có quyền.

| **INPUT** | | | |
| --- | --- | --- | --- |
| 1 | Token | varchar(150) | token của user đã đăng nhập thành công trước đó |
| 2 | issue\_id | int | issue id của Issue cần delete |
| **Output** | | | |
| 1 | code | varchar(5) | Mã lỗi |
| 2 | description | varchar(250) | Mô tả |

Frontend:

* Tạo nút “Delete”
* Gửi request kiểm tra quyền user
* Tạo screen modal delete Issue
* Popup lỗi

Backend:

* Kiểm tra token
* Kiểm tra quyền của User
* phản hồi client

Bước 2: Nếu user chọn delete, client gửi request delete Issue. Server kiểm tra thông tin có hợp lệ không? Nếu hợp lệ, Backend xóa issue và cập nhật database sau đó phản hồi client, client tạo popup thành công. Nếu không hợp lệ, backend trả mã lỗi, client popup thất bại.

| **INPUT** | | | |
| --- | --- | --- | --- |
| 1 | Token | varchar(150) | token của user đã đăng nhập thành công trước đó |
| 2 | issue\_id | int | id của issue muốn xóa |
| **Output** | | | |
| 1 | code | varchar(5) | Mã lỗi |
| 2 | description | varchar(250) | Mô tả |

Frontend:

* gửi request xóa Issue
* Tạo popup thành công
* tạo popup thất bại

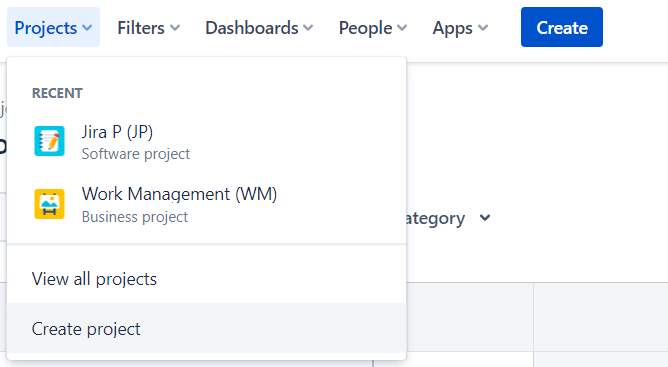
Backend:

* Kiểm tra token
* Kiểm tra data có hợp lệ và xóa Issue
* phản hồi client

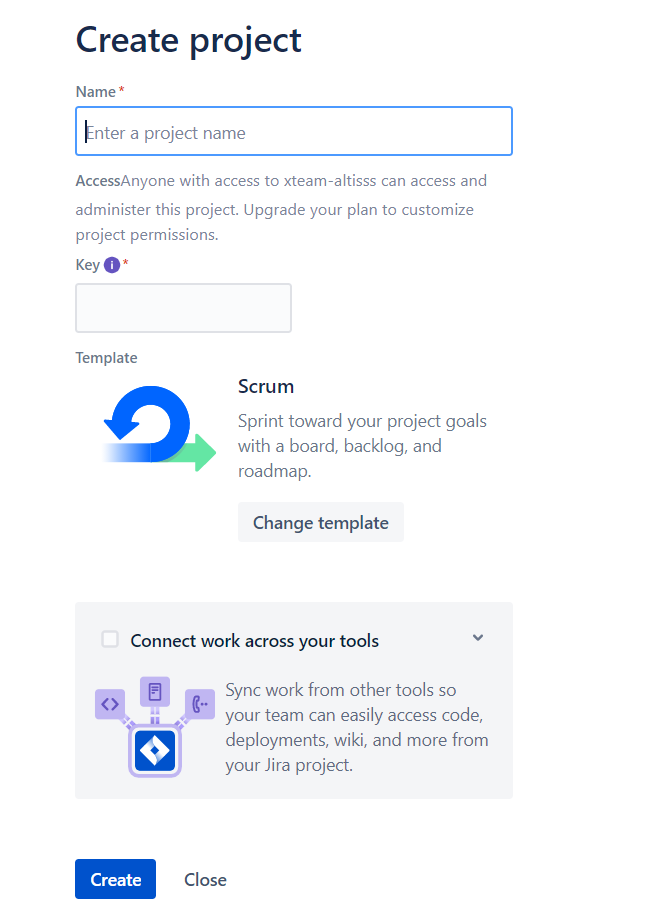
**9) TẠO PROJECT:**

9.1 UI tạo project:

UI : Nhấn vào “Project” trên thanh Topbar -> CLICK vào “Create project”

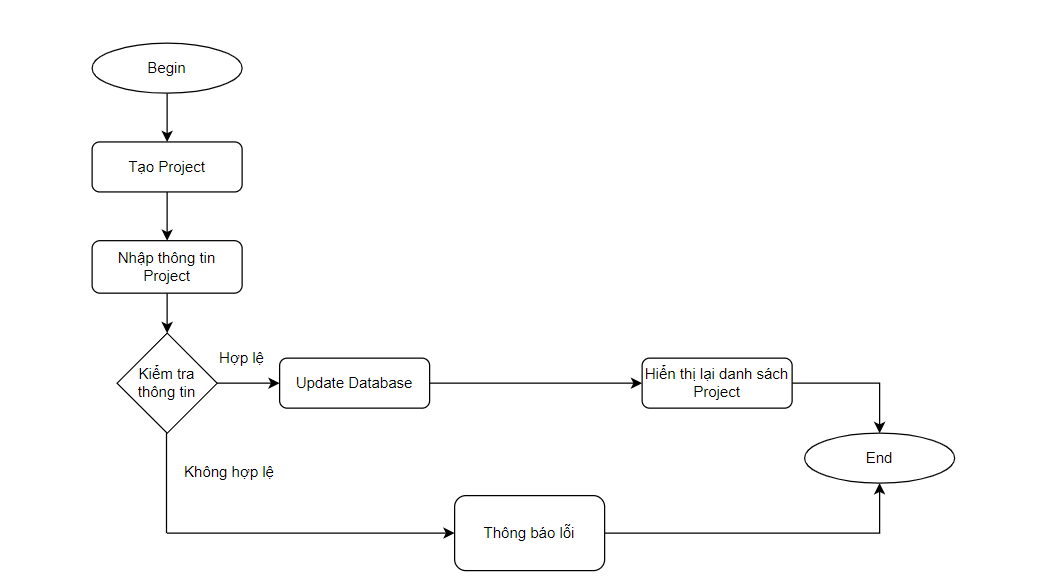


Screen Create Project



9.2 Đặc tả tạo Project:

**Mô tả**: Người dùng User truy cập vào thanh “Project” trên topbar, chọn tạo Project, sau đó nhập các thông tin như: Tên project, mã project. Thông báo cho user nếu thất bại hoặc tạo thành công.



| Input | | | |
| --- | --- | --- | --- |
| 1 | project\_name | varchar(250) | Tên project |
| 2 | project\_key | varchar(3) | Mã của project |
| 3 | user\_name | varchar(64) | Tên người quản lí project |
| 4 | project\_description | varchar(500) | Mô tả thêm về project |
| 5 | token | varchar(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 |

Frontend:

* Tạo nút “Project” trên thanh topbar.
* Tạo form để user input thông tin(bao gồm thông tin input như 1,2,3,4,5) để gửi đến server.
* Tạo popup thành công, thất bại .

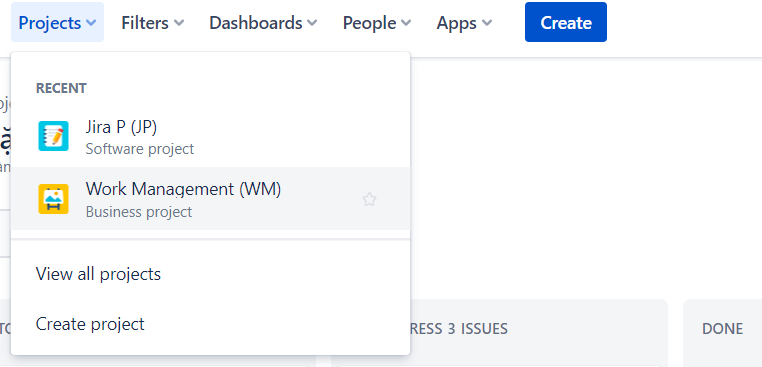
Backend:

* Xây dựng API : kiểm tra thông tin của dữ liệu input(xem đã tồn tại chưa). Sau đó trả về thành công hay thất bại

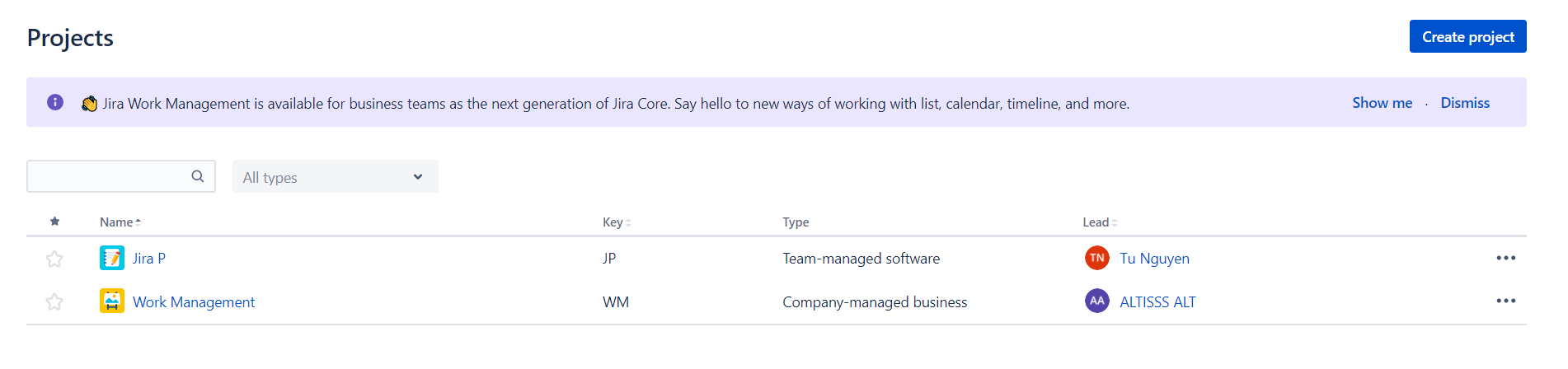
**10) Xem danh sách Project:**

10.1 UI xem danh sách project:

UI : Nhấn vào “Project” trên thanh Topbar -> CLICK vào “View all projects”



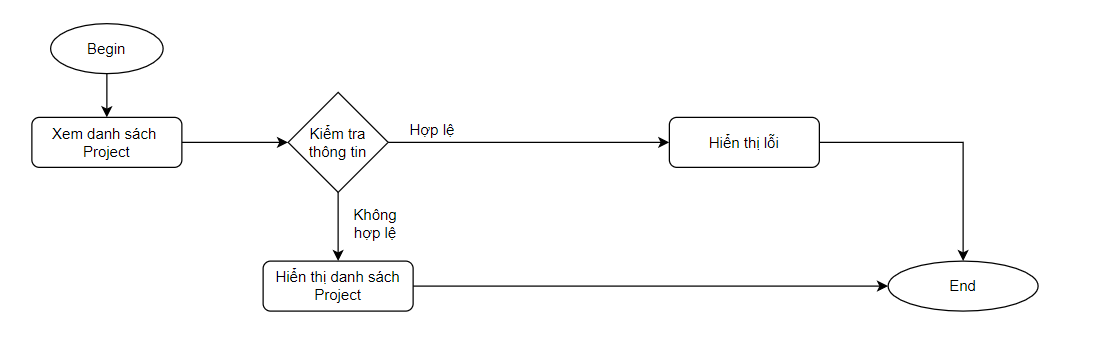
Screen View Project



Thêm Tiêu đề (Định dạng > Kiểu đoạn) và chúng sẽ xuất hiện trong phần mục lục.

10.2 Đặc tả xem danh sách Project:

**Mô tả**: User nhấn vào mục xem toàn bộ danh sách project, screen sẽ hiện ra tất cả project đã có.



Sau khi CLICK xem danh sách, client sẽ gửi request lên server để kiểm tra, nếu thành công sẽ hiện ra danh sách project, thất bại sẽ thông báo lỗi.

Dữ liệu gửi lên và nhận lại:

| **INPUT** | | | |
| --- | --- | --- | --- |
| 1 | token | varchar(150) | token của user đã đăng nhập thành công trước đó |
| 2 | user\_id | int | id của user |
| **Output** | | | |
| 1 | code | varchar(5) | Mã lỗi |
| 2 | description | varchar(250) | Mô tả |
| 3 | data | Max | danh sách Project |

Frontend:

* Gửi request lấy data liên quan
* Hiện screen chi tiết danh sách Project
* Tạo 2 button sửa/xóa cho từng project
* Popup lỗi

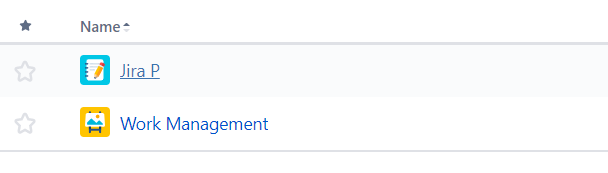
Backend:

* Kiểm tra token
* Kiểm tra thông tin và lấy data danh sách project
* trả data danh sách về client

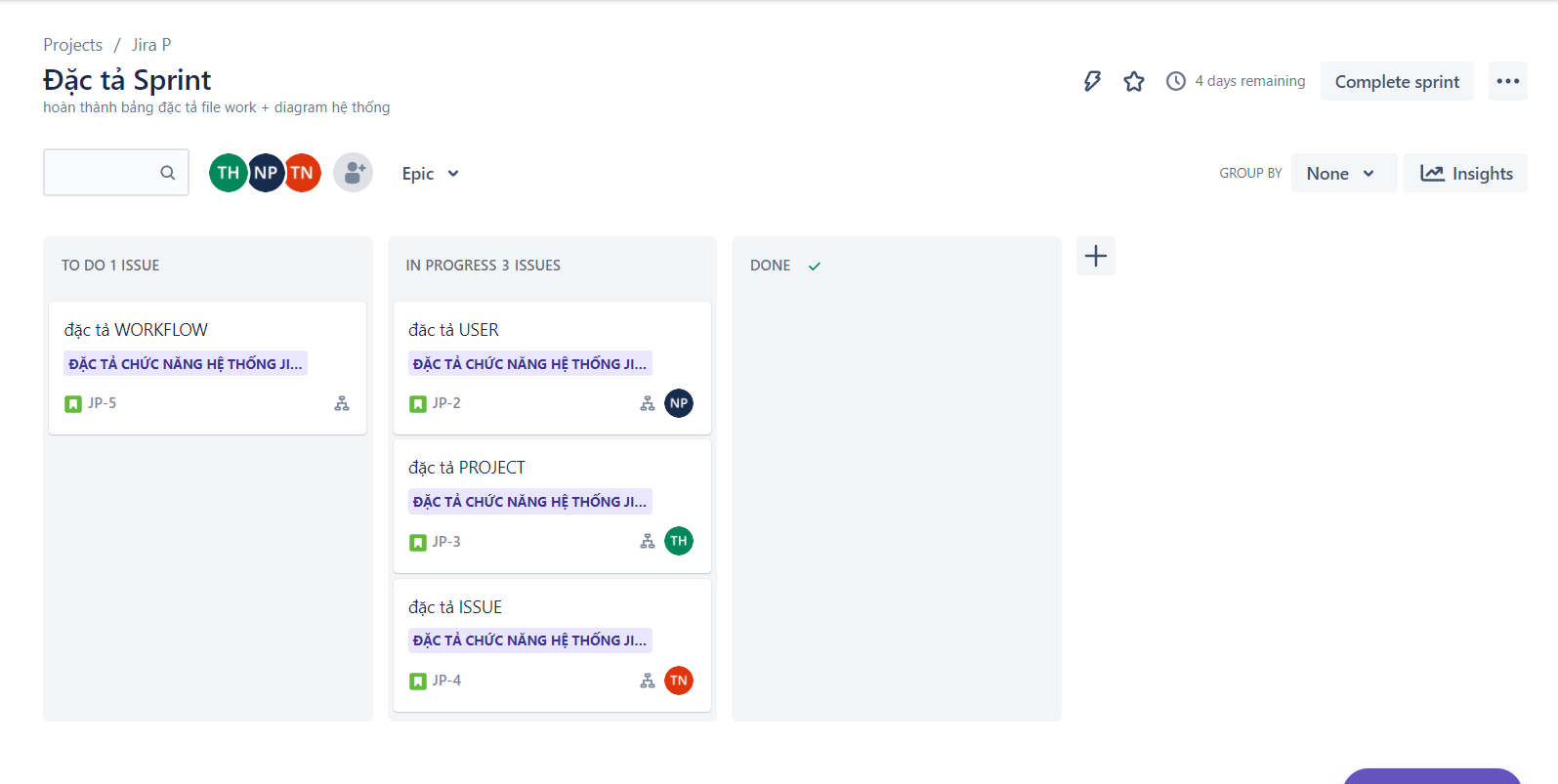
**11) XEM 1 PROJECT:**

11.1 UI xem 1 project

UI : Nhấn vào “Project” trên thanh Topbar -> CLICK vào “View all projects” -> Click chọn Project muốn xem

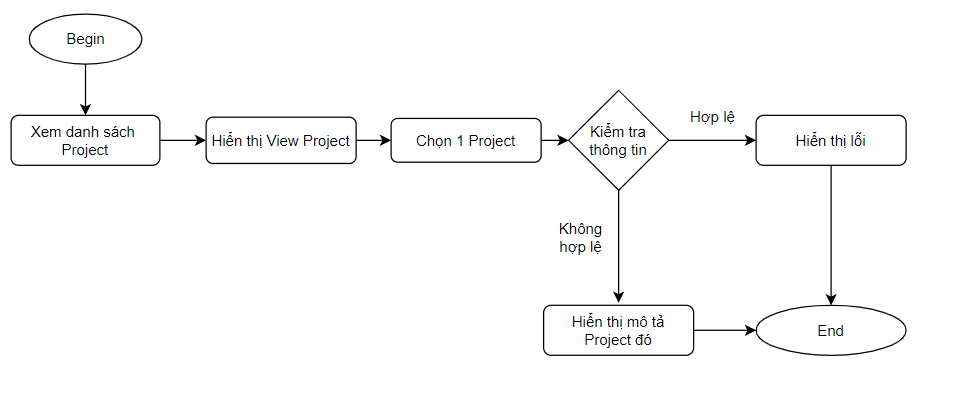


Sau khi CLICK:



11.2 Đặc tả xem 1 project:

Mô tả: User chọn vào mục danh sách project, sau đó chọn vào project muốn xem, screen sẽ hiển thị ra mô tả project đó.



Khi CLICK vào 1 project để xem, client sẽ gửi request lên server để kiểm tra thông tin sau đó sẽ trả về screen mô tả project đó (nếu thành công), popup thông báo lỗi (nếu thất bại)

Dữ liêu gửi lên và trả về:

| **INPUT** | | | |
| --- | --- | --- | --- |
| 1 | Token | varchar(150) | token của user đã đăng nhập thành công trước đó |
| 2 | project\_key | varchar(3) | project key của project cần xem |
| 3 | project\_user\_id | int | id của user muốn xem project đó |
| **Output** | | | |
| 1 | code | varchar(5) | Mã lỗi |
| 2 | description | varchar(250) | Mô tả |
| 3 | data | Max | data của project |

Frontend:

* Gửi request lấy data liên quan
* Hiện screen chi tiết danh sách Project
* Hiện chi tiết Project đc chọn

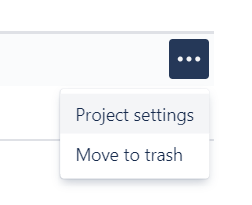
Backend:

* Kiểm tra token
* Kiểm tra thông tin và lấy data danh sách project
* trả data danh sách về client

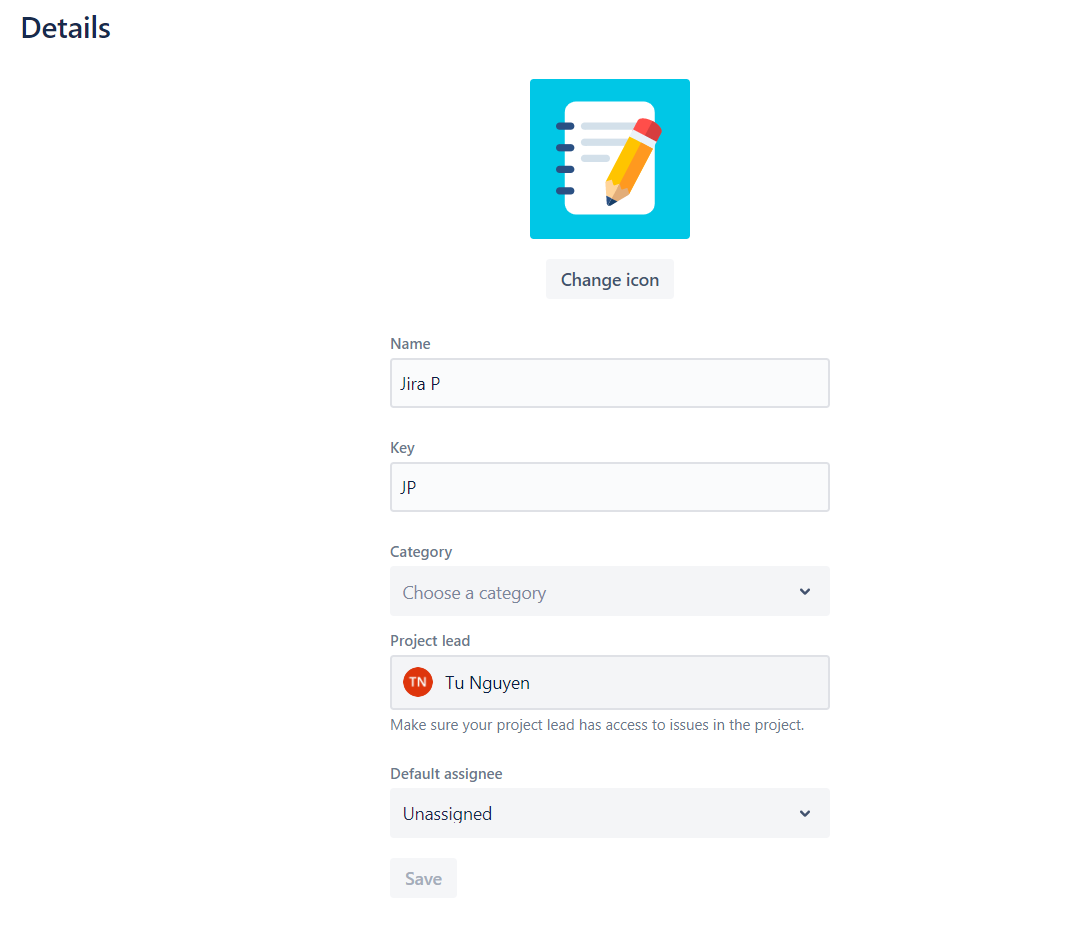
**12) SỬA PROJECT**

12.1 UI sửa project

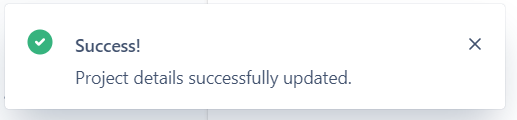
UI : Nhấn vào “Project” trên thanh Topbar -> CLICK vào “View all projects” -> Click vào biểu tượng -> Click vào “Project settings”



Screen:

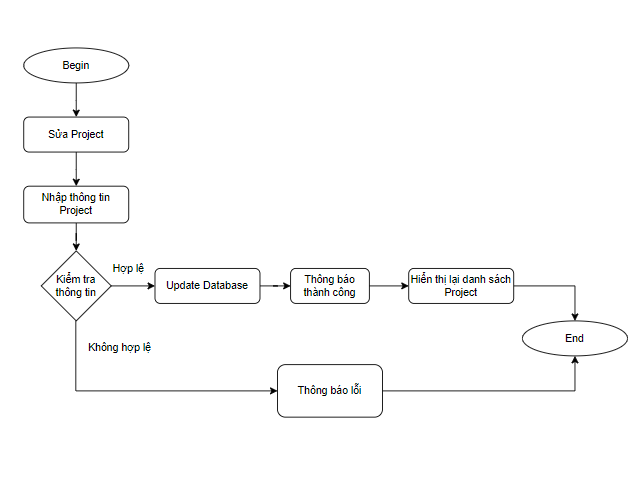


Sửa thành công Screen sẽ thông báo:



12.2 Đặc tả sửa Project:

**Mô tả:** User truy cập vào danh sách Project, chọn Button sửa project cho project cần sửa, screen hiển thị popup để user chỉnh sửa lại thông tin(tên project, mã project). Hiện popup thông báo thành công hay thất bại.



Dữ liệu gửi lên và trả về

| Input | | | |
| --- | --- | --- | --- |
| 1 | old\_project\_key | varchar(3) | Mã project hiện tại |
| 2 | project\_name | varchar(255) | Tên mới project |
| 3 | project\_key | varchar(3) | Mã mới của project |
| 4 | user\_name | varchar(64) | Tên người quản lí project |
| 5 | project\_description | varchar(500) | Mô tả thêm về project |
| 6 | token | varchar(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 |

Frontend:

* Tạo form để user chỉnh sửa lại project
* Gửi thông tin từ client đến server để kiểm tra thông tin có phù hợp
* Tạo popup thành công, thất bại
* Gọi api để làm mới danh sách project

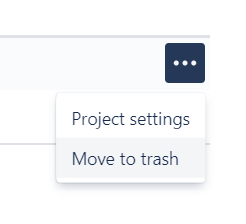
Backend:

* Kiểm tra dữ liệu thông tin từ input gửi vào, xem có tồn tại chưa,.... Nếu hợp lệ thì update lại data project và trả về kết quả client

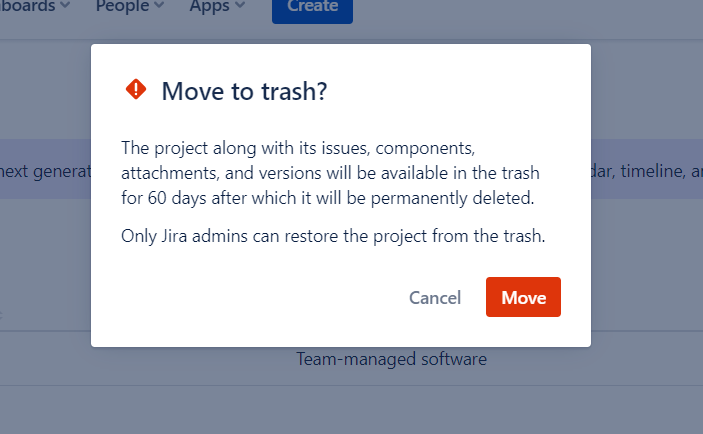
**13) XÓA PROJECT:**

13.1 UI xóa project

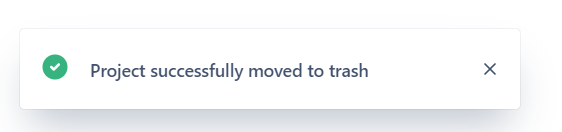
UI : Nhấn vào “Project” trên thanh Topbar -> CLICK vào “View all projects” -> Click vào biểu tượng -> Click vào “Project settings”



Screen : Hiện ra thông báo

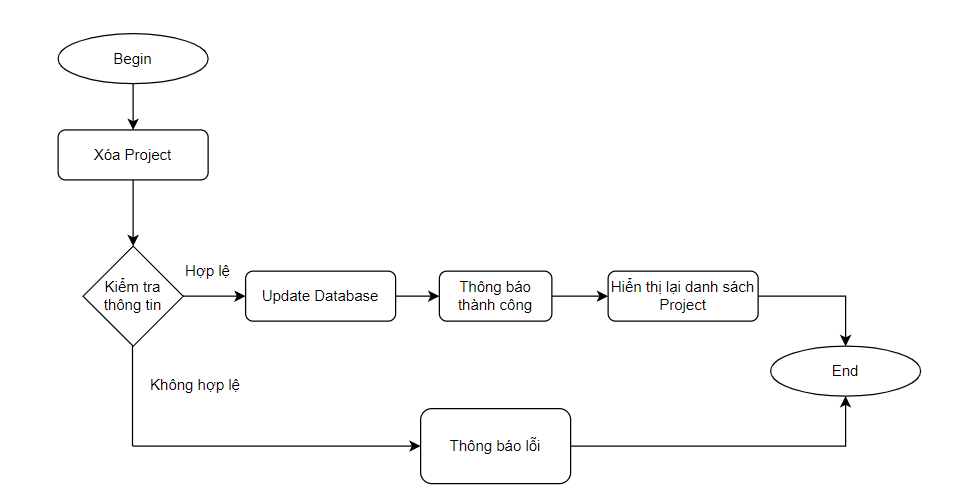


Nếu xóa thành công screen sẽ hiện ra thông báo:



13.2 Đặc tả xóa project

**Mô tả** : User truy cập vào danh sách tất cả các project, chọn project cần xóa, Click vào button xóa, sau đó screen sẽ hiển thị popup chắc chắc xóa hay hay không. Nếu đồng ý project sẽ bị xóa và screen trả lại các project còn lại. Nếu không đồng ý, screen sẽ quay lại hiện thị danh sách tất cả project.



Dữ liệu gửi lên và nhận về:

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

Frontend:

* Tạo form để user confirm có đồng ý xác nhận xóa hay không

Backend:

* Kiểm tra mã project mà user muốn xóa, tiến hành xóa hết data của project đó ở tất cả các bảng và phản hồi về client.