UI / UX KNOWLEGEND

BlueOC - Week 4

#### By [45. Trần Xuân Trường](mailto:tranxuantruong203hn@gmail.com) Mar 11, 2025

### UI (User Interface) - Giao diện người dùng

Tập trung vào: Hình thức và cảm quan của sản phẩm.

Nhiệm vụ: Thiết kế các yếu tố giao diện (nút bấm, biểu tượng, màu sắc, bố cục...) sao cho đẹp mắt, hấp dẫn và dễ sử dụng.

Mục tiêu: Tạo ra một giao diện trực quan, dễ nhìn, dễ tương tác, giúp người dùng dễ dàng thực hiện các thao tác mong muốn.

### UX (User Experience) - Trải nghiệm người dùng

Tập trung vào: Cảm xúc và trải nghiệm tổng thể của người dùng khi sử dụng sản phẩm.

Nhiệm vụ: Nghiên cứu người dùng, phân tích hành vi, thiết kế luồng tương tác, đảm bảo sản phẩm đáp ứng nhu cầu và mang lại sự hài lòng cho người dùng.

Mục tiêu: Tạo ra một trải nghiệm liền mạch, dễ chịu, hiệu quả và đáng nhớ cho người dùng.

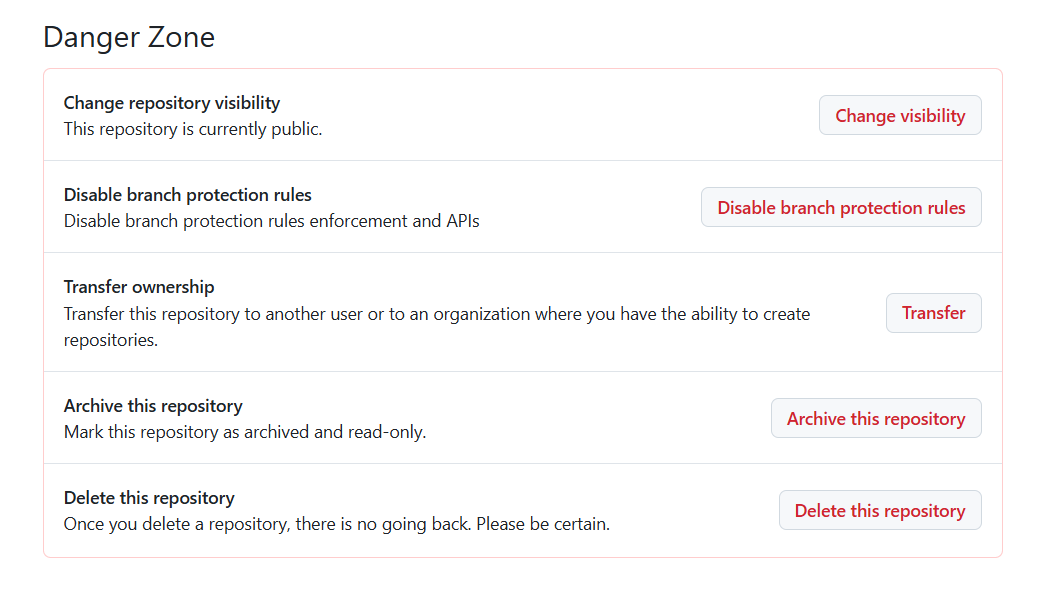
CASE STUDIES

### DANGER ZONE

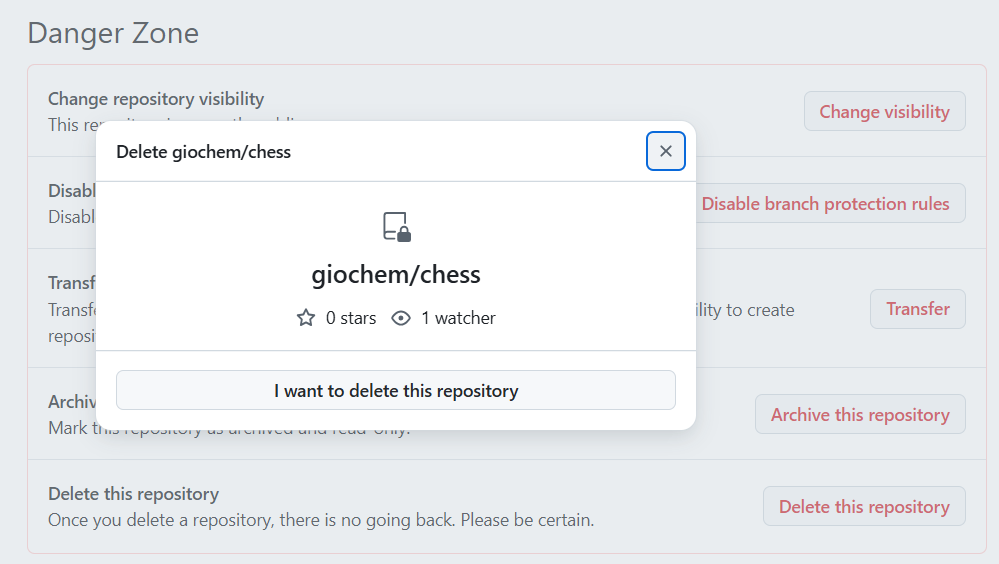
Việc cảnh báo người dùng về việc thực thi những hành động nguy hiểm trên ứng dụng.

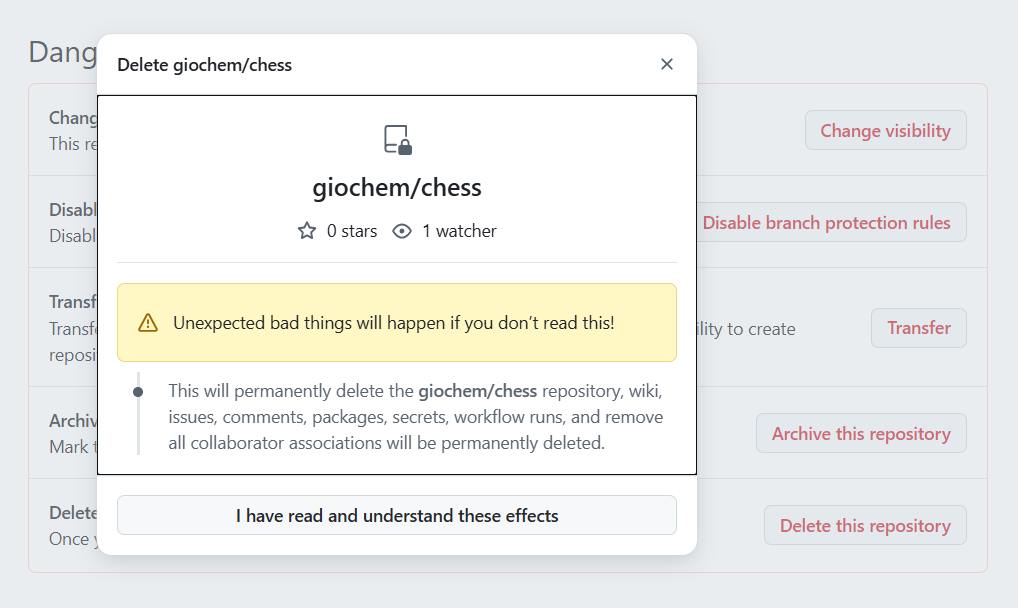
Ví dụ: việc xoá một repository trên github

Việc tìm kiếm vùng Danger Zone không dễ dàng, việc phải vào setting và kéo xuống cuối trang để tìm được vùng này xác nhận người dùng có ý thức về việc tìm kiếm việc những hành động quan trọng có chủ đích.

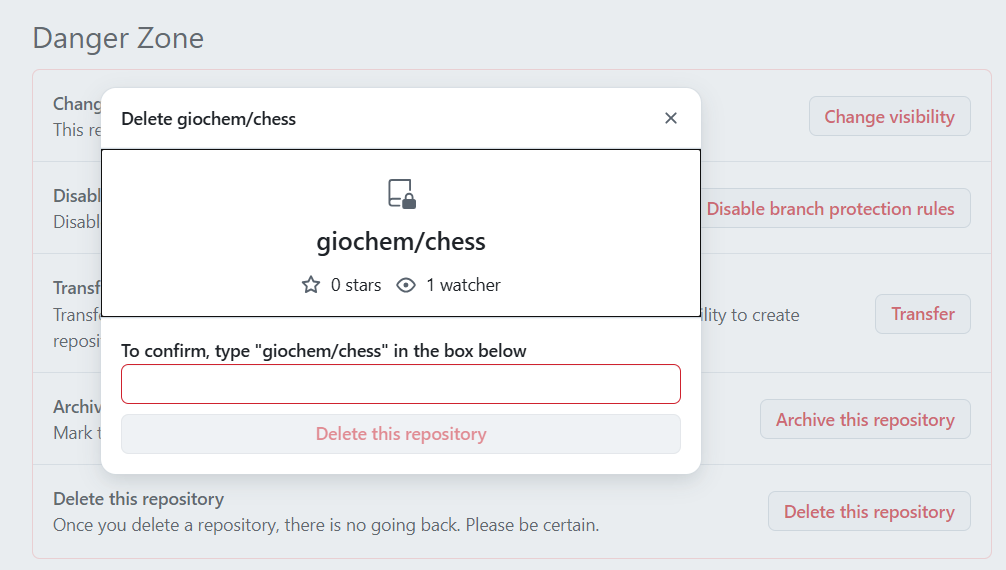


Không chỉ xác nhận xóa, github còn cảnh báo chi tiết về những tổn thất mà hành động này gây ra, đảm bảo người dùng hiểu rõ về nguy hiểm của hành động này



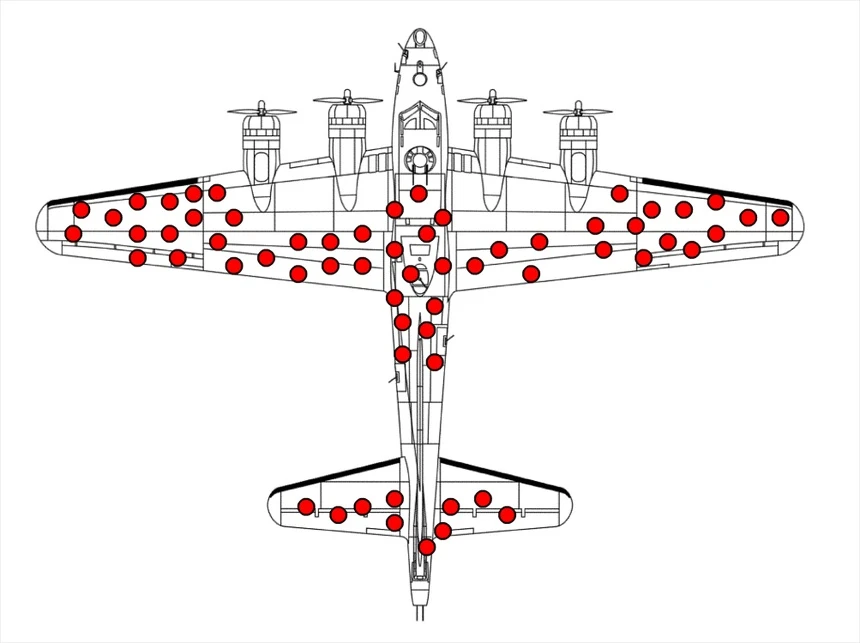


Mặc dù đã xác nhận rõ, github vẫn đưa ra 1 hành động mang tính phải dùng não 🙂, đảm bảo người dùng đã suy nghĩ kỹ trước khi xoá dự án.



Việc xoá dự án có thể cần thêm 1 bước xác nhận danh tính như nhập mật khẩu github, authentication code để xác minh danh tính, bước thêm vào chủ yếu để kéo dài thời gian xác nhận hành động trong Danger Zone.

### DESIGN THINKING



Wald nói, không cần dùng cho những nơi có nhiều lỗ đạn. Nó cần được bọc ở những nơi không có lỗ đạn: động cơ.

Lấy cảm hứng từ anh Bình về câu chuyện: Những lỗ đạn còn thiếu trên máy bay. Câu chuyện này diễn ra trong Thế chiến thứ hai, khi các nhà phân tích quân sự của quân Đồng Minh cố gắng tìm cách gia cố máy bay ném bom để tăng khả năng sống sót của chúng.

Quan sát:

* Các nhà phân tích đã kiểm tra những chiếc máy bay ném bom trở về sau các nhiệm vụ.
* Họ nhận thấy có nhiều lỗ đạn tập trung ở cánh, thân và đuôi máy bay.

Suy luận sai lầm:

* Họ cho rằng cần gia cố những khu vực bị trúng đạn nhiều nhất để tăng khả năng bảo vệ máy bay.

Phát hiện của Abraham Wald:

* Abraham Wald, một nhà toán học, đã chỉ ra sai lầm trong suy luận này.
* Ông lập luận rằng những chiếc máy bay trở về là những chiếc đã sống sót, còn những chiếc bị bắn hạ thì không có mặt để được kiểm tra.
* Do đó, những khu vực không có lỗ đạn trên những chiếc máy bay trở về có thể là những khu vực quan trọng, nơi nếu bị trúng đạn sẽ khiến máy bay bị rơi.

Kết luận:

* Wald đề xuất gia cố những khu vực không có lỗ đạn, vì đó mới là những điểm yếu thực sự của máy bay.

Câu chuyện cá nhân:

Khi đi thanh toán dịch vụ hiện nay, đa phần chúng ta sẽ quét mã thanh toán trực tuyến, nhưng có 1 vấn đề (trong rất nhiều vấn đề khác 😂 ). Nếu có ai đó giả mạo đơn thanh toán, bill vẫn đúng số tiền, ngân hàng, thậm chí cả ngày giờ và tên người nhận –> đâu sẽ là giải pháp chống lại việc này.

Uh, giải pháp để xử lý đã có rồi, nhưng e sẽ đưa ra 1 số cách giải quyết trước đó e đã từng nghĩ tới:

* xác nhận 1:1, việc này dễ, nhưng chỉ dành cho dịch vụ có lượng khách ko đông, và máy nhận tiền phải có ở quán,
* chụp bill của khách: bill này sẽ tạo ra 1 mã đặc biệt để đảm bảo việc chuyển khoản là chính xác, 👏 . Nếu suy nghĩ kỹ thêm về giải pháp này, sẽ có rất nhiều vấn đề phát sinh như time để scan, hoặc đúng bill nhưng là bill hôm trước,...
* nhìn chung việc cố gắng phát hiện gian lận ở đây e chỉ thấy tạo ra thêm nhiều vấn đề hơn,

Vậy giải pháp là: máy đọc đơn thanh toán, ta có 1 cái mã QR + máy nói, vậy là ta ko tìm kẻ gian nữa, kẻ gian phải tìm cách hack cái máy đọc đơn của ta, 😂.

So, thật sự việc làm ra cái máy ko khó, việc tìm ra cách ứng dụng của cái máy mới thực sự đáng nể, thề cái máy nó hay thực sự.

Nói về việc thiết kế giao diện, việc thiết kế 1 giao diện phải wow hay animation siêu mướt là điều dev, designer e nghĩ ai cũng đều muốn, nó thể hiện cho tài năng và công sức của họ. Tuy vậy, giao diện nhìn thì bình thường, nhưng ai cũng dễ dàng làm quen, hiểu được chức năng của từng nút bấm, ko phải dùng não này, những người bị thiếu khuyến cũng có thể trải nghiệm được là 1 điều mà phải có cái tâm, cái tầm, và chắc chắn phải trải qua nhiều trải nghiệm thực tế mới có được.

Em thì được cái làm thì xấu nhưng logic nhìn cũng ngu nốt nên cũng ko cái trải nghiệm gì lắm, méo có mẫu sẵn thì tìm cái thư viện nào đẹp làm theo cho nó theo format thoi, nhìn ko đấm vào mắt mình là đc 🙂.

### SHOW WHAT NEEDS TO BE SHOWN

Cái này liên quan về những gì giao diện được hiển thị cho người dùng. Nếu người dùng đăng nhập 1 tài khoản vào BlueOC gồm email, password.

Việc hiển thị chi tiết thông báo tất cả mọi thứ cho người dùng khi đăng nhập sai có là cần thiết ?

Vd:

Nếu nhập đúng email, sai password: Email người dùng đã chính xác, mật khẩu người dùng ko chính xác, mời nhập lại.

Nếu nhập đúng email, password không đúng định dạng (giả định rule cho password có ít nhất 1 chữ cái in Hoa): Email người dùng đã chính xác, mật khẩu người dùng chưa có chữ cái in Hoa, mời nhập lại.

Nếu nhập sai email, đúng password 🙂 : Email người dùng ko chính xác, mật khẩu người dùng đã chính xác, mời nhập lại.

Đọc được từ 1 bài viết từ đâu đấy, đăng nhập sai thì auto: Email hoặc mật khẩu ko chính xác.(méo cần bik có cái lỗi gì hết)