9:00am: gặp khách hàng giải thích thêm về entity

# Soạn tin

+ Ở cái duyệt tin thì sẽ có 1 cái check box tự đăng (ở phần detail design)

+ Người đang và hạ tin **HTML** (thường là chủ thể của bạn tin đó), còn hay hơn thì sẽ trên cấp thì có thể hạ tin của người dưới cấp. Còn tin chuyên đề thì chỉ có tổng biên tập mới được hạ thôi, muốn ai hạ thì ủy quyền, còn ngang cấp thì chỉ được hạ tin của mình thôi, 🡪 Rule để xuống detail design

**<Công cụ soạn tin, công cụ hiển thị bản tin**

Làm thế nào công cụ soạn tin nó làm thế nào gửi dữ liệu cho công cụ hiển thị bản tin 🡪 Xem lại require, provide cho công cụ hiển thị bản tin. Nhưng mà trong hệ thống soạn tin vẫn chưa hình dung được, tạo tin r 🡪 thì làm sao hiển thị bản tin lên.> 🡪 phân tích

Bên hiển thị soạn tin và quản trị tin, db có phải là một system không?

Từ hệ thống soạn tin đưa tin vào db, r hiển thị lấy từ db 🡺 bây h khác, sẽ sử dụng một cái **service<để tăng tính bảo mật>**, công cụ hiển thị soạn tin yêu cầu bộ quản trị tin một cái service để đăng tin

Hệ thống hiển thị thì đòi rất rõ rệt, ra bên **intranet** và **internet** *<2 hệ thống độc lập>e*ntity??

2 bên nó cung cấp cho mình 2 cái service khác nhau. 🡪 phân tích.

Chuyển tin có thể chuyển nhiều tin ( khi chuyển có sự cố gì sao )..

Ủy quyền thì phải có hủy quyền

# Hỏi đáp

Xóa trên android là không cho xóa

Trên android thì giống trên destop ( không cho xóa trên android). *<xuống phần detail phân tích tiếp>*

+ Công cụ quản trị bộ từ điển trên android nó lấy dữ liệu từ hệ thống nào, lấy ntn??

+ Hệ thống bên android lấy dữ liệu từ cái quản trị bên desktop, không bao giờ nó lấy qua db được 🡪 nó chỉ lấy trên service mà thôi. Gửi lên **service (gì đó)**, thì service tự động đưa dữ liệu vào bên android. 🡪 **Rule**

**Vậy thì trong cái provide bên từ điển trên desktop cung cấp service cho bên android.**

# Ý chung

**+ Database có phải là entity hay không?**

***+ <bỏ phần use case diagram>***

***+ <Mật kết nối và mất điện là khác nhau thì nó lưu trữ như thế nào?>***

+ Thiếu entity trong usecase description

+ xem lại precondition (vd: công cụ soạn tin đã sẵn sàng>

+ **Alternate flows** nó không độc lập mà là đường rẽ nhành trên cái main flow nào.

+ Exception viết theo cái cũ nhưng phải tại chỗ nào nữa, thời điểm nào nữa

+ Post condition: vd: là một bản tin đã được tạo và sẵn sàng cho những thằng nào dùng dùng..

Phải mô tả các sự cố trong các use case diagram ( có cả quality attribute <tính toàn vẹn, tính chỉnh sửa..)

*Không kết nối được service thì sẽ được lưu xuống local*

*Khi mất điện thì ntn??? 🡪 không giải quyết vấn đề mất điện*

*Nếu như alternate flows có post condition 🡪 thì có thêm alternate flows cho post condition(vd : như file lưu tạm thì lưu ở đâu vd local thì mô tả ntn?..)*

# Quality Attributes

Luxen search no áp dụng cho những dự án nhỏ mà vừa 🡪 xem lại,

Công cụ soạn tin phải

Khi mà gửi thì tạm thời phải lưu xuống db của mình, ( mặc dịnh là 3 cái db – 3 hệ thống)

2 hệ thống internal và internet, thì hiển trị ở intranet thì dùng service, còn internal thì lưu xuống database

Công cụ soạn tin cho phép tạo index ( luxen tạo các chỉ mục cho các nội dung cần tìm kiếm )

Đáp ứng 1000 người thì nó thuộc về performance ( tộc độ truy cập và dữ liệu là ntn? )

Kham khảo thêm cái tính toàn vẹn **lưu tạm** thì nó là avaibility hay là tính toàn vẹn?

Cấu hình 🡪 usibility: cũng tùy, xem có làm kịp không ( người dùng làm được )

Khả năng mở rộng qua iOS 🡪 các chức năng này có thể chạy bên iOS

**Scalbility**: có khả năng mở rộng **độc lập,** tập trung được với các chức năng ( quản trị bộ từ điển, hiển thị bộ từ điển và công cụ soạn tin)

-

-

Tìm hiểu xem về usability xem còn gì không 🡪 gọi ý cho khách hàng

Index đúng ra phải nhanh, nhưng nó còn phụ thuộc vào cái luxen

Security: mã hóa (mã hóa mật khẩu) và chứng thực (là mình đăng nhập pass qua hệ thống khác thì phải đăng nhập lại ) chứng thực người dùng, xác nhận người dùng.

Từ điển là những text trả lời 🡪 khỏi quality attribute, mặc định là nhanh. Có cái là nó truyền xuống android thì phải nhanh.

**Xem xét tất cả những người dùng làm đều phải lưu vết lại**

**Tìm hiểu xem thuật toán nào hình lớn mà đẩy lên internet cho nhanh**