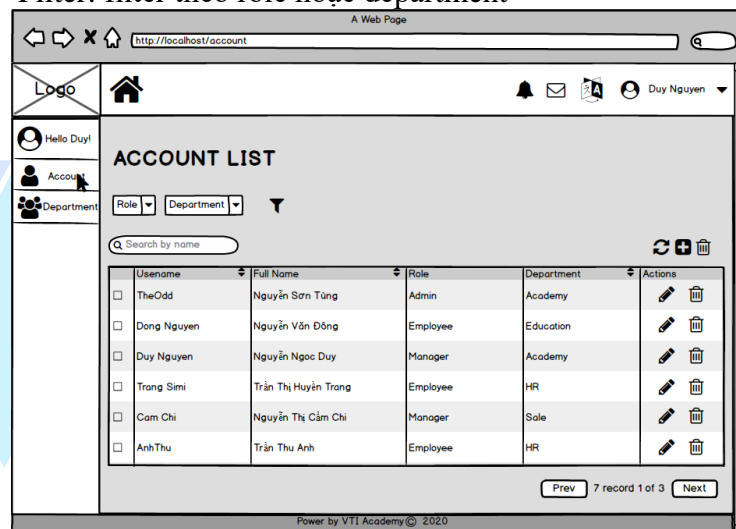


## Advanced Java – Final Exam

### Function

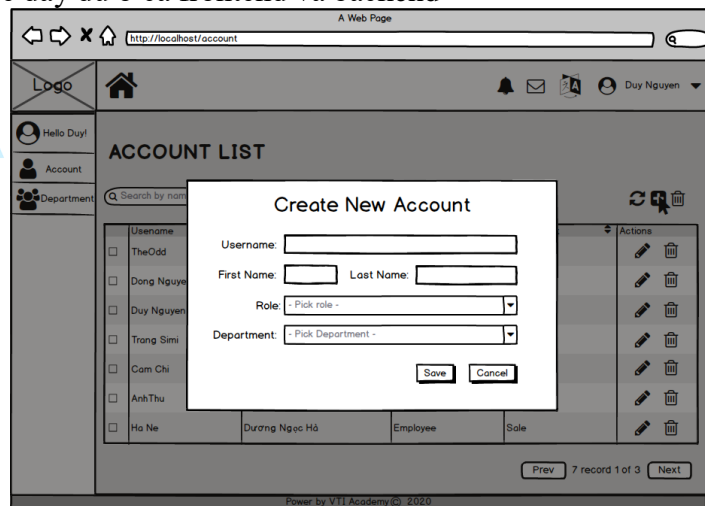
#### 1. View List Account

- Thiết kế paging, sort, search, filter cho View List như theo thiết kế
  - Paging: hiển thị 10 records/page
  - Sort: default sắp xếp theo createDate desc
  - Search: sẽ search theo username & firstname & lastname
  - Filter: filter theo role hoặc department



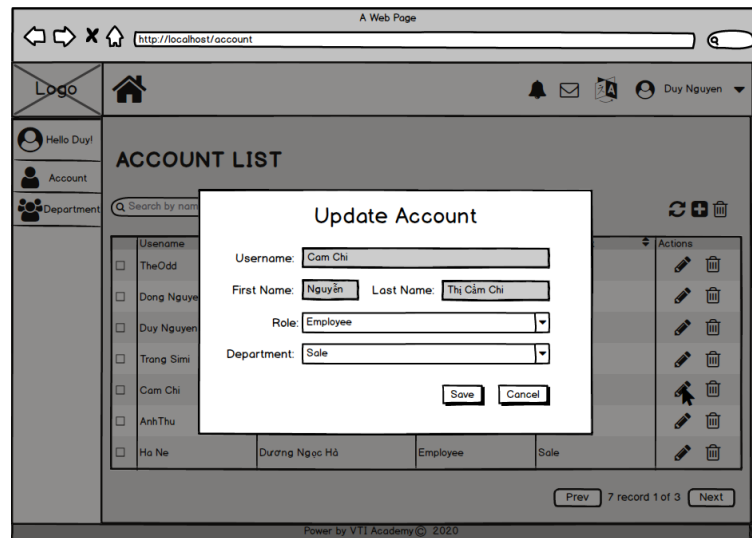
#### 2. Create Account (Admin sẽ tạo account)

- Validate đầy đủ ở cả frontend và backend



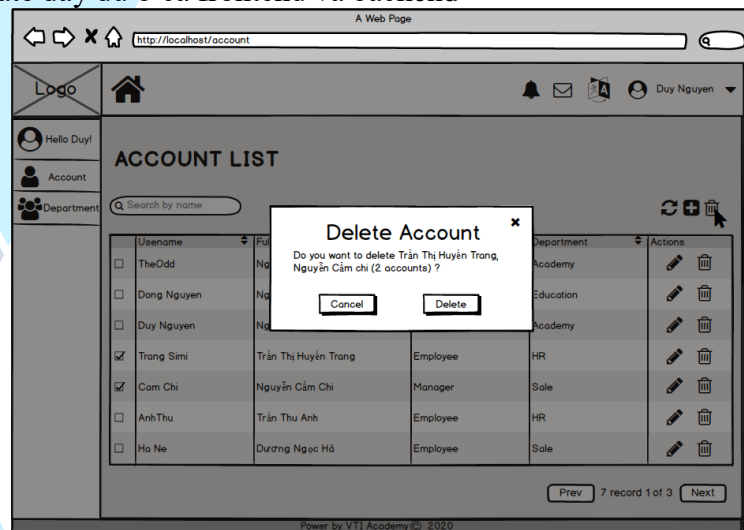
#### 3. Update Account

- Validate đầy đủ ở cả frontend và backend



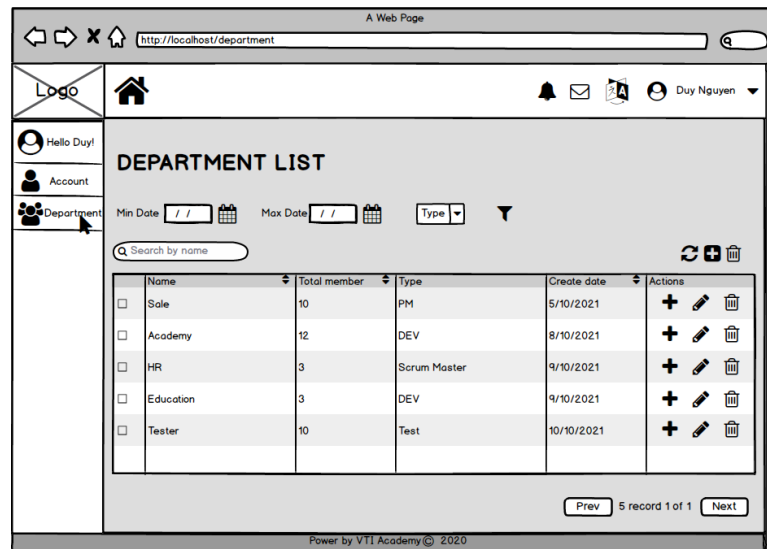
#### 4. Delete nhiều Account 1 lúc

- Validate đầy đủ ở cả frontend và backend



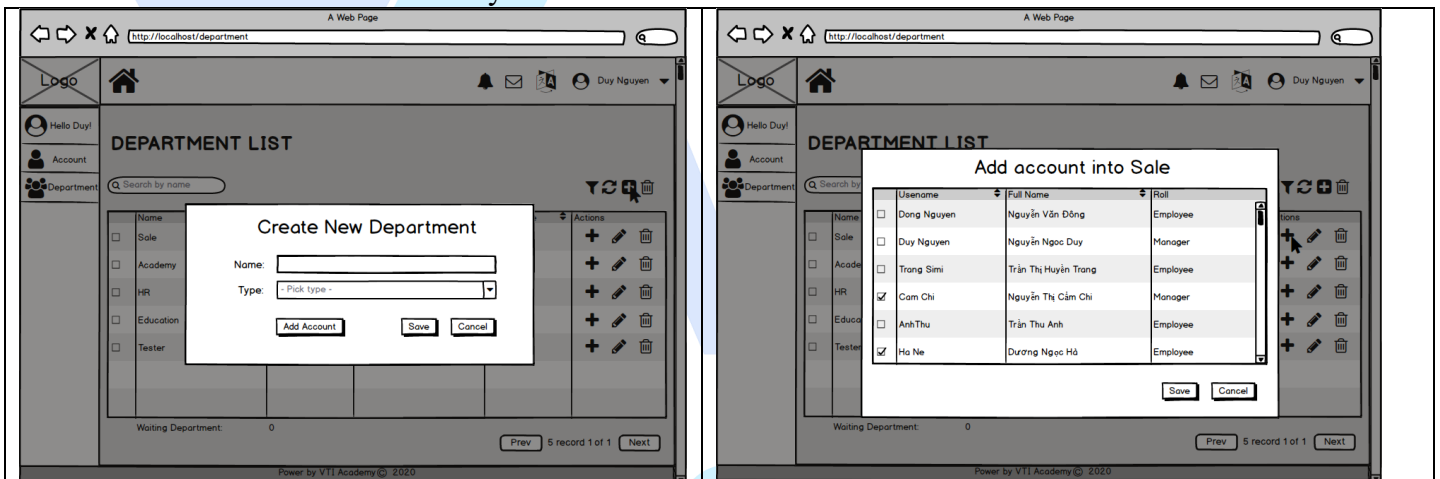
#### 5. View List Department

- Thiết kế paging, sort, search, filter cho View List như theo thiết kế
  - Paging: hiển thị 10 records/page
  - Sort: default sắp xếp theo Total Member desc
  - Search: sẽ search theo name
  - Filter: filter theo min created date, max created date, hoặc type



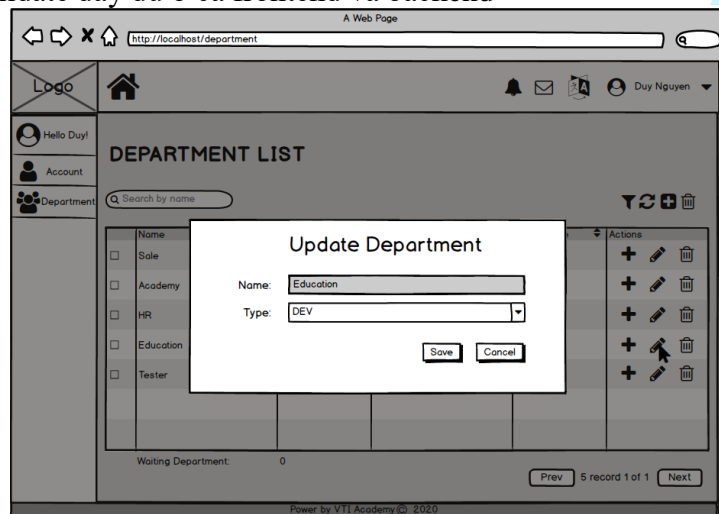
## 6. Create Department & Add account to Department

- Validate đầy đủ ở cả frontend và backend



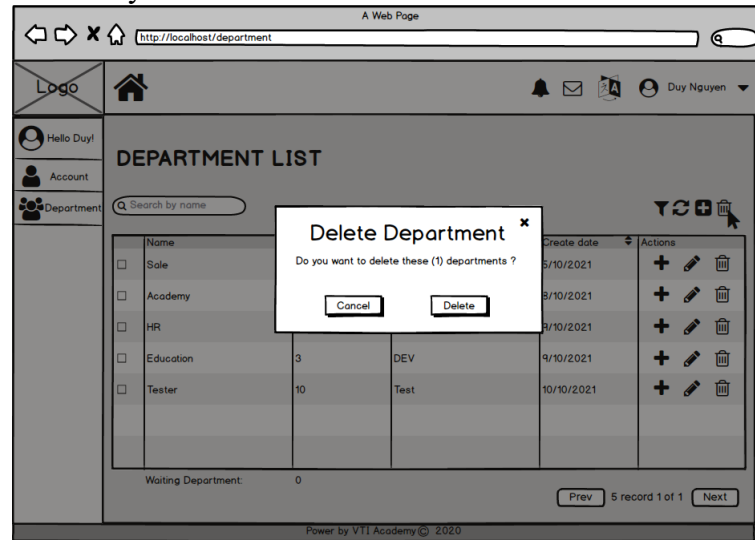
## 7. Update Department

- Validate đầy đủ ở cả frontend và backend



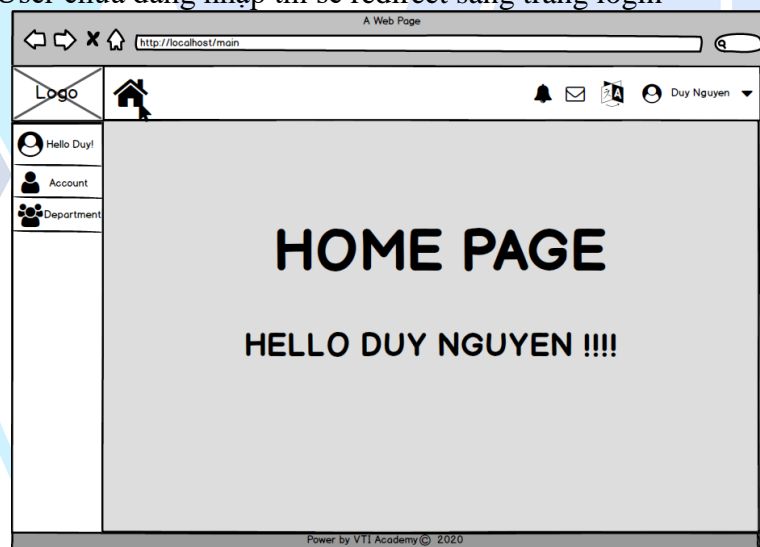
## 8. Delete Department

- Validate đầy đủ ở cả frontend và backend



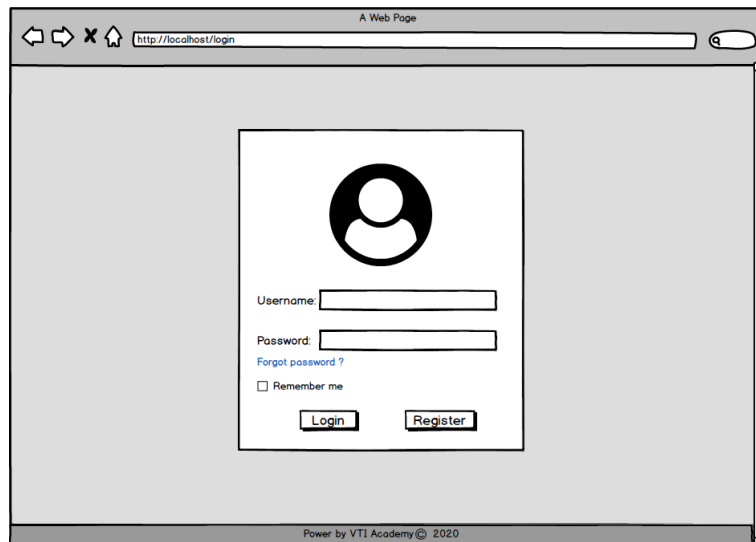
## 9. Home page

- Thiết kế home page ở frontend như thiết kế
- Nếu User chưa đăng nhập thì sẽ redirect sang trang login



## 10. Login & Remember me

- Login vào thông qua Username & Password
- Sử dụng JWT hoặc Basic Authentication
- Có thông báo Account chưa active (nếu có)

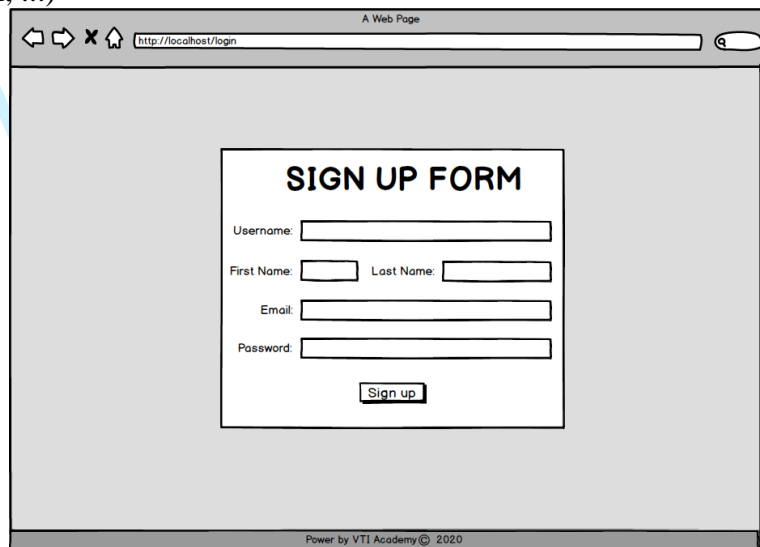


## 11. Authorization

- Admin có thể vào được các API
  - CRUD Account
  - CRUD Department
- Employee và Manager chỉ vào được Home Page
- Ai cũng vào được
  - Login, Remember me, Register Account, Forgot password

## 12. Register Account

- User phải nhập vào các field: Username, first name, last name, email, password
- Có validate cả ở frontend và backend (VD: trùng username, trùng email, ...)



## 13. Forgot password

- User nhập địa chỉ email vào và link thay đổi password sẽ được gửi qua email



#### 14. Change password

- Tự thiết kế

#### 15. Thiết kế đa ngôn ngữ trên cả frontend và backend

- Bao gồm: Việt nam, Tiếng Anh và Tiếng pháp (Tự Google Translate)

#### 16. Error page

- Tự thiết kế error page ở frontend khi gặp lỗi bất ngờ từ backend

#### 17. Deploy

- Sau khi làm xong sẽ deploy code backend & frontend lên heroku

#### Database Requirement:

1. Tự thiết kế Database
2. Thêm nhiều record vào từng table (để phục vụ filter, sort, paging, ...) (khuyến khích sử dụng tool để import)

#### Backend Requirement:

1. Sử dụng mô hình 3-layers & Spring boot để xây dựng chương trình
2. Validate đầy đủ API
3. Xử lý các Exception hay gặp phải như thiếu param, sai datatype, sai url, validate sai, ... và in ra API theo 1 định dạng nhất định
4. Lưu thông tin User Request + các Exception vào 1 file Application.log
5. Thêm Document cho API, sử dụng tools: Swagger & HATEOAS
6. Chia file config application.properties thành các môi trường Dev, Test, Production
7. Cross Origin: chỉ cho phép frontend của mình truy cập được
8. Tích hợp các Tools
  - a. Lombok
  - b. Spring Boot DevTools
  - c. ModelMapper
9. Lưu lại các API & Link Demo vào Postman Collection

### Frontend Requirement:

1. Thiết kế web theo single-page
2. Validate và thông báo đầy đủ giống như backend & database
3. Có page error khi backend xảy ra lỗi bất ngờ
4. Áp dụng bootstrap hoặc 1 admin template nào đó (file đính kèm)

### Exam Requirement:

1. Mỗi function 1 điểm → 6 function tối thiểu sẽ qua
2. Có thể làm theo nhóm (tối đa 3 người/1 nhóm, mỗi người làm cả frontend và backend)

### Nộp bài (về nhà làm và đẩy lên drive sau)

1. Up bài thực hành lên git và đặt tên "FinalExam"
2. Sau đó quay video 10 phút để giới thiệu về bài làm theo các bước sau  
B1: Giới thiệu họ và tên  
B2: Nói lại đề bài xem đọc hiểu tới đâu  
B3: Nói qua ý tưởng làm từng câu và giới thiệu Code  
B4: Demo code

### Chú ý:

- Viết đúng coding convention
- Tuân thủ các best practice
- Không chép bài người khác (sẽ có hình thức phạt nếu bị phát hiện)