**Important Notes:**

* For every call to the API, should be a check what is the position of the user and if the user can call the specific API.

**API:**

* /user
  + /

request: POST with user’s info

response: status of success / reason of failure

usage: adding a new user to the system. Should be used only for managing the users by a manager.

* + /

request: DELETE with username

response: status of success / reason of failure

usage: can only be used by a manger user. Do not delete the user – change the active field to false;

* + /

request: PUT with user info

response: status of success / reason of failure

usage: could be used by the manager to change details about a user. Cannot change username or password. Could be used by the user itself to update its profile data (only when the user is logged in).

* + /position

request: POST with username

response: position of the user

usage:

* /users
  + /

request: GET

response: all **active** users in the system / reason of failure

usage: can only be used by a manger user to view all the users

* /register
  + /

request: POST with user’s info (full customer info / username & password)

response: status of success / reason of failure

usage: signing up or signing in to the system. If the username exists – use passport-local to authenticate the user. If the username does not exist – create a Customer user with the provided data and sign the user in.

* /signout
  + /

request: GET

response: status of success / reason of failure

usage: if there is no user logged in return error, else logout the current user.

* /shoes
  + /

request: GET (with category)

response: returns a list of all shoes in a specific category

usage: shopping

* + /categories

request: GET

response: returns a list of possible filters for the shoes with the possible values for each filter.

usage: filtering the shoes

* + /filters

request: GET

response: returns a list of possible categories for the shoes

usage: links for the shoes in the menu

* + /colors

request: GET

response: returns a list of possible colors for the shoes

usage: filtering the shoes

* + /designer

request: GET

response: returns a list of possible designers for the shoes

usage: filtering the shoes

**Models:**

* User
  + Username
  + Password
  + Name
  + Email
  + Position (default – customer)
  + Required for purchasing:
    - Credit card
    - Phone
    - Address
    - Full name
* Employee extends User
  + ID
  + Work percentage
  + Home Branch
  + Manager Evaluation
  + Start work date
  + Birthday
  + Position set to Employee
* Manager extends Employee
  + Position set to Manager
* Item
  + Label
  + Serial number
  + Price
  + Quantity
  + Description
  + Categories (list of Category)
  + Filters (list)
* Category
  + ID
  + Name