Mind The Code JAVA Project.

Σύστημα διαχείρισης προσωπικού.

# Εισαγωγή

Καταγραφή των λειτουργικών απαιτήσεων για την υλοποίηση εφαρμογής «Μητρώο Εργαζομένων» της UniSystems. Ο υπάλληλος της UniSystems αποτελεί την βασική οντότητα για μια σειρά από εφαρμογές. Ο βασικός στόχος του Μητρώου είναι να αποτελέσει το single point of entry για τα βασικά στοιχεία των εργαζόμενων και παράλληλα να αποτελέσει την βασική πηγή άντλησης στοιχείων εργαζομένων από όλες τις εφαρμογές της εταιρείας.

# Οργανωτική Δομή

Η οργανωτική δομή της UniSystems υποστηρίζεται από μια σειρά από ιεραρχίες τριών επιπέδων οι οποίες πάντα καταλήγουν στον υπάλληλο.

**Επίπεδο #1:** Στο πρώτο επίπεδο βρίσκεται η **εταιρεία** (UniSystems)

**Επίπεδο #2:** Στο δεύτερο επίπεδο βρίσκονται τα **Business Units** ανεξάρτητα από το αντικείμενο στο οποίο δραστηριοποιούνται. Ο τύπος του Business Unit καθορίζει τα επόμενα επίπεδα της ιεραρχίας.

**Επίπεδο #3:** Στο τρίτο επίπεδο βρίσκονται τα **Departments** που απαρτίζουν το κάθε Business Unit του προηγούμενου επιπέδου.

**Επίπεδο #3:** Στο τέταρτο επίπεδο βρίσκονται τα **Units** για κάθε Department του προηγούμενου επιπέδου.

**Επίπεδο #4:** Στο τέταρτο και τελευταίο επίπεδο βρίσκονται οι Υπάλληλοι που ανήκουν σε κάθε Unit του προηγούμενου επιπέδου.

Ενδεικτικά οι πληροφορίες που χρειάζονται για τους υπάλληλους :

* Αριθμός Μητρώου
* Επώνυμο
* Όνομα
* Διεύθυνση Κατοικίας
* Τηλέφωνο
* Ημερομηνία Πρόσληψης
* Ημερομηνία Αποχώρησης
* Status (active, inactive)
* Contract Type (UniSystems, External)
* Company
* Business Unit
* Department
* Unit
* Position (e.g. Senior Analyst, Department Manager etc.)

# Απαιτήσεις.

1. ~~Υλοποιήστε ενα CRUD Application σε Spring Boot το οποίο να κάνει expose ένα RESTful API της πληροφορίες της παραπάνω οργανωτικής δομής, κάνοντας χρήση της αρχιτεκτονικής Controller-Service-Repository.~~
2. ~~Προστατέψτε το σύστημα έτσι ώστε η οργανωτική δομή της εταιρείας να μην μπορεί να αλλαχθεί κατα λάθως. Μόνο οι υπάλληλοι (employees) θα επιστρέπεται να μπορούν να αλλαχθούν ή να προστεθούν καινούργιοι.~~
3. ~~Στα πλαίσια του DEMO θα πρεπει το application να περιέχει Mock Data.~~
4. ~~Ενα παραμετρικό endpoint το οποίο θα δεχεται δυο παραμετρους : 1) searchCriteria ∈ [unit, businessUnit, company, department] και 2) id Long και θα επιστρέφει όλους τους υπάλληλους που ανήκουν στο <searchCriteria> eg unit με το συγκεκριμένο id.~~
5. Δημιουργήστε ένα νέο Entity task (id, title, desc , estimationA : int (οι μέρες που χρειάζεται για να υλοποηθεί), estimationB: int, estimationC: int, status[NEW, STARTED, DONE], updates List<String>), ένας υπάλληλος θα μπορεί να έχει assigned από κανένα μέχρι Χ tasks, ένα task θα μπορούν να το δουλεύουν από 0 μέχρι N υπάλληλοι. Υλοποιήστε όλα τα CRUD operations για αυτό το Entity που θα επιστρέφουν την πληροφορία (id, title, desc, difficulty, status). Αν ο μέσος όρος των 3 estimations είναι <2 diffculty = EASY, <=4 MEDIUM, >5 HARD.
6. Δημιουργήστε endpoint όπου θα δέχεται taskId και θα επιστρέφει “όλη” την πληροφόρία του Task(id, title, desc, difficulty, status, **assignedEmployees**, **updates**).
7. Δημιουργήστε extra έλεγχο στα POST/PATCH/PUT endpoints του παραπάνω entity να μην μπορούν να αναλάβουν ένα task employees που να ανήκουν σε διαφορετικά units.
8. Δημιουργήστε search endpoint(s) που θα μπορούν να αναζητηθούν tasks βάσει του difficulty, και/ή του αριθμού των εργαζομένων που τα δουλεύουν.
9. Error handling και input validation για όλα τα endpoints του συστήματος.
10. Unit & Acceptance tests σε τα endpoints.
11. Κάντε deploy το σύστημα σε έναν public server.

# Extra Απαιτήσεις.

1. Αντικαταστήστε την H2 in memory database με MySQL.
2. Εισάγετε στο σύστημα role based authentication, οι ρόλοι θα είναι admin, employee. Ο employee θα μπορεί μόνο να κάνει GET πληροφορίες και θα παίρνει ένα error αν προσπαθήσει να αλλάξει κάτι, ο admin θα έχει πρόσβαση σε όλα τα endpoints.

# Style Points.

1. Χρήση OO design patterns.
2. Κάντε refactor το role based authenticaton με ρόλους **admin**, **companyManager**, **businessUnitManager**, **departmentManager**, **unitManager**, **employee**.

O employee θα μπορεί μόνο να αναζητήσει πληροφορίες. Οι managers θα έχουν πλήρη πρόσβαση αλλαγής σε ότι βρίσκεται από κάτω τους.

Πχ admin = companyOwner = πλήρης πρόσβαση, departmentManager = πρόσβαση αλλαγής σε departments, units, employees, tasks. etc..