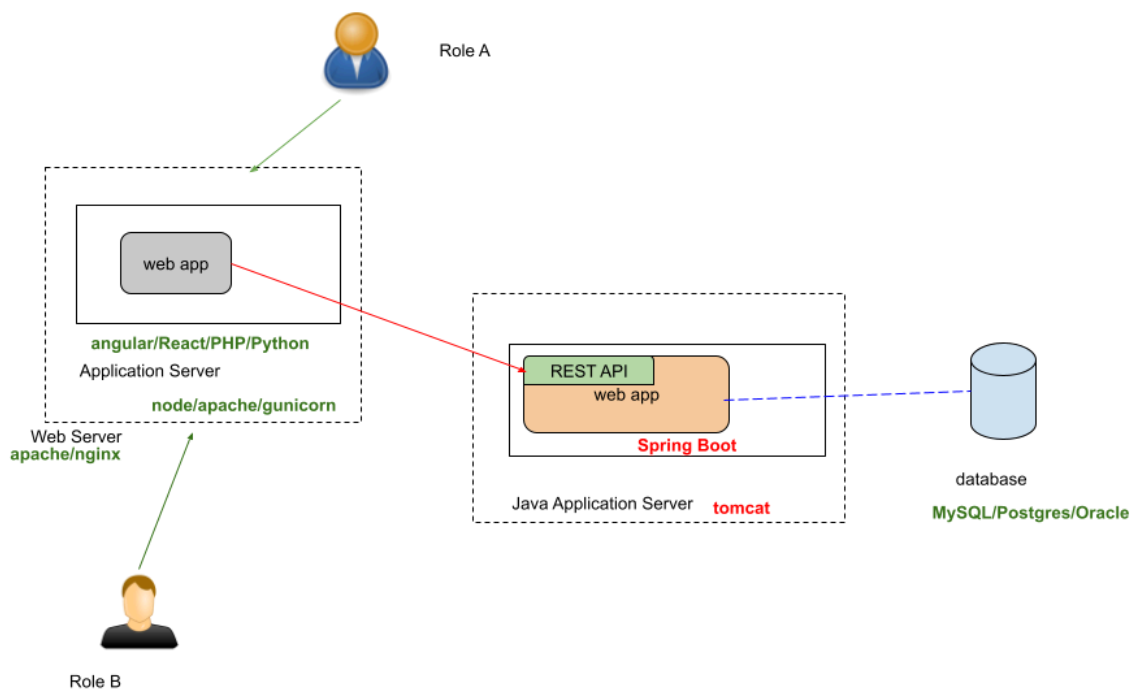


Γενικά

Κάθε εργασία στοχεύει στη δημιουργία ενός πληροφοριακού συστήματος που αποτελείται από δύο μέρη: την backend εφαρμογή με το υποστηριζόμενο REST API και την frontend εφαρμογή. Η πρώτη πρακτικά αντιστοιχεί στο σύστημα που αποθηκεύει στη βάση τα στοιχεία, ελέγχει την πρόσβαση μέσω αυθεντικοποίησης και υποστηρίζει όλες τις λειτουργίες και τις προσφέρει μέσα από ένα REST API, ενώ η δεύτερη παρουσιάζει τη λειτουργικότητα της εφαρμογής στους τελικούς χρήστες. Και οι δύο εφαρμογές θα υλοποιηθούν χρησιμοποιώντας προγραμματιστικά εργαλεία και περιβάλλοντα για την ανάπτυξη web-based εφαρμογών. Πρακτικά μπορούν να υλοποιηθούν ανεξάρτητα (στο πραγματικό κόσμο συνήθως και από διαφορετικές ομάδες προγραμματιστών). Η αρχιτεκτονική του συστήματος παρουσιάζεται στην εικόνα 1.



Εικόνα 1: Μια προτεινόμενη αρχιτεκτονική για το σύστημα που θα αναπτύξετε. Με **πράσινο** χρώμα σημειώνονται ενδεικτικές τεχνολογίες, με **κόκκινο** οι υποχρεωτικές

Συνοπτικά

Παραδοτέο		Τύπος	Προθεσμία
Σχεδίαση	<ul style="list-style-type: none">• παραδοχές• use case diagram• class diagram• component/deployment diagram• προαιρετικά activity, sequence diagrams	αρχείο pdf που θα περιλαμβάνει τα UML διαγράμματα σε εικόνες	6/12/2023
Υλοποίηση backend	<ul style="list-style-type: none">• git repo και REST documentation (swagger)• επικαιροποίηση προηγούμενου παραδοτέου	το προηγούμενο αρχείο επικαιροποιημένο μαζί με τα git repo links	10/1/2024
Υλοποίηση frontend	<ul style="list-style-type: none">• git repo• επικαιροποίηση προηγούμενων παραδοτέων	το προηγούμενο αρχείο επικαιροποιημένο μαζί με τα git repo links	31/1/2024

Για τη σχεδίαση της εφαρμογής, ξεκινήστε με ένα έγγραφο (pdf) το οποίο θα επικαιροποιείτε κάθε φορά

Πρώτο μέρος: backend εφαρμογή και REST API

Το περιβάλλον που θα υλοποιήσετε θα πρέπει υποστηρίζει την λειτουργικότητα που περιγράφεται στο αντίστοιχο μέρος της [περιγραφής](#) της εργασίας. Σε ότι αφορά την παρεχόμενη λειτουργικότητα θα πρέπει να παρέχονται διαφορετικές συναρτήσεις με τη μορφή REST API (μέσω REST Controllers), που θα μπορούν να κληθούν σε επόμενο χρόνο από το interface του μενού της εφαρμογής (frontend) από διαφορετικούς χρήστες της εφαρμογής μέσα από ένα Web Browser.

Στο backend θα πρέπει να υποστηρίζεται επίσης η **διαχείριση των χρηστών** της εφαρμογής από το διαχειριστή του συστήματος. Τη δημιουργία νέων χρηστών, όπως επίσης και την τροποποίηση των στοιχείων τους ή τη διαγραφή τους μπορεί να πραγματοποιήσει μόνο ο διαχειριστής του συστήματος, που σε καμία περίπτωση δεν είναι ένας από τους χρήστες του συστήματος.

Ο διαχειριστής του συστήματος μπορεί να συνδέεται σε αυτό και να το χρησιμοποιεί είτε μέσα από την frontend εφαρμογή (που θα φτιάξετε στο δεύτερο μέρος), είτε απευθείας στο backend σύστημα μέσα από ένα δικό του interface (χωρίς REST API). Στο πρώτο παραδοτέο σας που αφορά το σχεδιασμό της εφαρμογής θα πρέπει να περιγράψετε και να αιτιολογήσετε την επιλογή σας.

Εκτός από τους χρήστες, θα πρέπει να διατηρούνται πληροφορίες και για τους **ρόλους** που ορίζονται. Η διαχείριση χρηστών θα υλοποιηθεί ανεξάρτητα και θα πρέπει να υποστηρίζει την καταχώρηση/διαγραφή/ενημέρωση στοιχείων χρηστών και την ανάθεσή τους σε ρόλους.

Λάβετε υπόψη σας ότι κατά την εκτέλεση της εφαρμογής οι χρήστες θα πρέπει να αποτρέπονται από ενέργειες που δεν έχουν δικαίωμα να κάνουν και να ενημερώνονται με εύληπτα προειδοποιητικά ή απολογιστικά μηνύματα για τυχόν σφάλματα.

ΠΡΟΣΟΧΗ: Χρησιμοποιήστε το Spring Security για τη δημιουργία ρόλων και χρηστών. Θα το δούμε στο εργαστήριο του μαθήματος.

Τεχνολογία υλοποίησης:

- Spring Boot + Spring Data JPA
- Spring Security
- Frontend για διαχειριστή (αν χρειαστεί): Είτε κάποιο Spring Template Engine (JSP, Thymeleaf, Velocity, κτλ) είτε κάποιο JS library ή Framework

ΠΡΟΣΟΧΗ: Αποφασίστε τις τεχνολογίες υλοποίησης του συστήματος σας και σχεδιάστε την αρχιτεκτονική του όσο πιο νωρίς γίνεται ώστε να μπορείτε να σχεδιάσετε πιο εύκολα το σύστημά σας.

Παραδοτέα πρώτου μέρους

Περιγραφή	Ημερομηνία
<p>Σχεδιασμός της εφαρμογής. Σε αυτόν θα περιλαμβάνονται τα ακόλουθα:</p> <ul style="list-style-type: none">◦ Σχήμα με τη συνολική αρχιτεκτονική όλων των τμημάτων του συστήματος σας (έχετε τουλάχιστον δύο διακριτά μέρη, λειτουργικότητα και διαχείριση χρηστών)◦ Περιγραφή και τεκμηρίωση της δομής της βάσης για την εφαρμογή και τη διαχείριση καταλόγου χρηστών, χρησιμοποιώντας τυποποιημένες μεθοδολογίες (Class - ER διαγράμματα)◦ Αναφορά και σχολιασμός των υπηρεσιών που παρέχονται σε κάθε διαφορετικό ρόλο (αντιστοιχούν τουλάχιστον στις επιλογές που	6/12/2023

<p>έχει κάθε χρήστης διαθέσιμες όταν εισέρχεται στο περιβάλλον της εφαρμογής), χρησιμοποιώντας τυποποιημένες μεθοδολογίες (όπως UML use case diagram)</p> <ul style="list-style-type: none"> ○ Περιγραφή της λειτουργικότητας για κάθε υπηρεσία που παρέχεται σε υψηλό επίπεδο (κατ' ελάχιστον λειτουργικότητα, συστατικά στοιχεία, δεδομένα εισόδου, εξόδου), χρησιμοποιώντας τυποποιημένες μεθοδολογίες (όπως UML activity, sequence, state machine diagrams) ○ Παραδοχές που έχετε κάνει για το σύστημα 	
<ul style="list-style-type: none"> ● Υλοποίηση της εφαρμογής σας χρησιμοποιώντας την web-based αρχιτεκτονική και σχετικά εργαλεία (όπως Spring, JSPs, JSF, MySQL, Postgres και όποιο εργαλείο / Τεχνολογία εσείς επιθυμείτε για το front). Επίσης απαιτείται Component/Deployment διάγραμμα. Η υλοποίηση μπορεί να είναι ενδεικτική και να ολοκληρωθεί/τροποποιηθεί στο 2ο παραδοτέο ● Εγχειρίδιο χρήσης, όπου να αναφέρεται και εξηγείται το παρεχόμενο REST API 	<p>10/1/2024</p>

Διευκρινίσεις/Παρατηρήσεις που δίνονται βασισμένες σε ερωτήσεις σας

- Στο class diagram ξεκινήστε αποτυπώνοντας τις βασικές οντότητες και τα βασικά τους χαρακτηριστικά (attributes). Αφήστε προς το παρόν τις λειτουργίες(operations), εκτός και είναι ξεκάθαρες για εσάς από τώρα.
- Όσον αφορά τον κατάλογο χρηστών, όταν μιλήσουμε για το spring security θα το δούμε αναλυτικότερα. Αυτό που πρέπει να έχετε κατά νου είναι ότι θέλουμε να κρατάμε τους χρήστες, τους ρόλους τους και τις ενέργειες που κάνει ο καθένας. Αυτή η πληροφορία κατανέμεται στο use case και το class diagrams.

Οδηγίες - Πληροφορίες

- Η εργασία αυτή είναι ομαδική. Απαιτείται η συμμετοχή όλων για την υλοποίησή της.
- Τεκμηριώστε τον κώδικά σας και στείλτε το link του κώδικά σας από το github/gitlab/bitbucket και βάλτε οδηγίες ώστε κάποιος να μπορεί να την εγκαταστήσει/τρέξει στον υπολογιστή του (αρχείο README). Συμπεριλάβετε και τυχόν sql αρχεία με αρχικά δεδομένα.
- Να παραδώσετε την εργασία σας μέσα από το e-class μέχρι την προβλεπόμενη ημερομηνία.
- Στο eclass θα ανεβάσετε σε ένα pdf αρχείο με δομή, τα ακόλουθα:
 - **το τεύχος σχεδιασμού** της εφαρμογής
 - **το τεύχος υλοποίησης της εφαρμογής** που θα περιλαμβάνει τον κώδικα (link προς αποθετήριο κώδικα με αναλυτικές οδηγίες για την εγκατάσταση - README αρχείο) και όποια άλλη πληροφορία σχετική με την υλοποίηση κρίνετε σκόπιμο να αναφερθεί και
 - **το εγχειρίδιο χρήσης** (user manual) της εφαρμογής που πρέπει να περιλαμβάνει εύληπτες οδηγίες για τη χρήση της (αναφέρεται στον frontend προγραμματιστή)
- Τα ανωτέρω να είναι σε μορφή **pdf** και το όνομά του να είναι της μορφής **Ομάδα-Χ-παραδοτέο-1.pdf**. Το κείμενο να είναι δομημένο και να περιέχει εξώφυλλο με τα στοιχεία της ομάδας. Ο αριθμός της ομάδας βρίσκεται στο έγγραφο ανάθεσης εργασιών **[εκκρεμεί ink - θα αναρτηθεί μόλις ολοκληρωθεί η διαδικασία ανάθεσης]**
- Η ίδια η εφαρμογή θα πρέπει να έχει εγκατασταθεί και να “τρέχει” σε κάποιον υπολογιστή ώστε να μπορεί να παρουσιαστεί. **Όποια ομάδα το επιθυμεί μπορεί να χρησιμοποιήσει κάποιο vm στο [azure](#) ή να λάβει κουπόνι στο google cloud.**

Δεύτερο μέρος: Εφαρμογή FrontEnd

Υλοποιήστε ένα ιστότοπο (web site) που υποστηρίζει την λειτουργικότητα που περιγράφεται όλους τους χρήστες του συστήματος σας, παρέχοντας σε αυτό ένα web-based frontend. Οι επισκέπτες προσπελαίνουν τον ιστότοπο χρησιμοποιώντας έναν οποιοδήποτε Web browser.

Σε ότι αφορά το ιστότοπο, μπορείτε να τον υλοποιήσετε χρησιμοποιώντας ότι τεχνολογία επιθυμείτε, λάβετε όμως υπόψη σας ότι πρέπει να υλοποιηθεί **ΑΝΕΞΑΡΤΗΤΑ** από την backend εφαρμογή για λόγους ασφαλείας.

Σε ότι αφορά τις παρεχόμενες υπηρεσίες θα πρέπει η υλοποίηση τους να βασιστεί εξ' ολοκλήρου στο REST API που παρέχεται από την backend εφαρμογή. Σε ότι αφορά την παρεχόμενη λειτουργικότητα θα πρέπει να παρέχονται διαφορετικές υπηρεσίες (δυνατότητες) στους χρήστες μέσα από μενού που είναι προσπελάσιμα από αυτό μέσα από ένα Web Browser.

ΠΡΟΣΟΧΗ: Σε καμία περίπτωση από την frontend εφαρμογή δεν θα πρέπει να μιλάτε απευθείας με τη βάση, παρά μόνο καλώντας το REST API της backend εφαρμογής έχοντας κάνει authentication

Τεχνολογία υλοποίησης: Javascript (Angular, React, VueJS, Vanilla JavaScript) ή Spring Framework ή Python (Django, Flask, FastAPI) ή κάτι συναφές (π.χ. Ruby, PHP)

Παραδοτέα δεύτερου μέρους

Περιγραφή	Ημερομηνία
Επικαιροποίηση του σχεδιασμού της εφαρμογής και Υλοποίηση της εφαρμογής σας χρησιμοποιώντας το REST API καθώς και όποια άλλα εργαλεία επιθυμείτε για τη δημιουργία του διαδικτυακού τύπου.	31/1/2023

Οδηγίες - Πληροφορίες

- Η εργασία αυτή είναι ομαδική. Απαιτείται η συμμετοχή όλων για την υλοποίησή της.
- Τεκμηριώστε τον κώδικά σας και στείλτε το link του κώδικά σας από το github ή το bitbucket και βάλτε οδηγίες ώστε κάποιος να μπορεί να την εγκαταστήσει/τρέξει στον υπολογιστή του (αρχείο README).
- Να παραδώσετε την εργασία σας μέσα από το e-class μέχρι την προβλεπόμενη ημερομηνία και ώρα 12 τα μεσάνυχτα.
- Στο eclass θα ανεβάσετε:
 - ο **το τεύχος σχεδιασμού και υλοποίησης της εφαρμογής** σε μορφή pdf, το όνομα του οποίου να είναι της μορφής **Ομάδα-Χ-παραδοτέο-2.pdf**. Το κείμενο να είναι δομημένο και να περιέχει εξώφυλλο. Ο αριθμός της ομάδας βρίσκεται στο [έγγραφο ανάθεσης εργασιών](#)
 - ο **το εγχειρίδιο χρήσης** (user manual) της εφαρμογής που πρέπει να περιλαμβάνει εύληπτες οδηγίες για τη χρήση της (αναφέρεται στον τελικό χρήστη)
- Η ίδια η εφαρμογή θα πρέπει να έχει εγκατασταθεί και να “τρέχει” σε κάποιον υπολογιστή ώστε να μπορεί να παρουσιαστεί. **Όποια ομάδα το επιθυμεί μπορεί να χρησιμοποιήσει κάποιο vm στο [azure](#)**

Σε περίπτωση που χρησιμοποιήσετε λογισμικό (κώδικα ή library) από κάποιον τρίτο, να αναφέρετε την πηγή.

Παρουσίαση Συστήματος

Στο τέλος του εξαμήνου και με τη χρήση υποστηρικτικού υλικού θα πρέπει να παρουσιάσετε το σύστημα σας στους διδάσκοντες του μαθήματος.

Η παρουσίαση στοχεύει κυρίως να δείξετε το σκοπό, σχεδίαση και λειτουργικότητα που παρέχει το σύστημα σας και όχι τον κώδικα που φτιάξατε (ο οποίος είναι έτσι και αλλιώς διαθέσιμος στους διδάσκοντες)

- ★ Η παρουσίαση της εργασίας και όλο το συνοδευτικό υλικό θα πρέπει να είναι έτοιμο ΠΡΙΝ την παρουσίαση

ΠΡΟΣΟΧΗ: Φροντίστε να έχετε ενδεικτικούς χρήστες της κάθε κατηγορίας και ενδεικτικά δεδομένα ώστε να μπορείτε να τρέξετε τα σενάρια χρήσης κατά την παρουσίαση του συστήματος.

Ημερομηνίες Παράδοσης:

Πρώτο Μέρος

- τεύχος σχεδιασμού: **6/12/2023**
- υλοποίηση και εγχειρίδιο χρήσης: **10/1/2024**

Δεύτερο Μέρος

- τεύχος σχεδιασμού και υλοποίηση και εγχειρίδιο χρήσης: **31/1/2024**

Κριτήρια Βαθμολογίας Εργασιών

1ο Μέρος	2ο Μέρος	Γενικά				
Υλοποίηση Λειτουργικότητας	Υλοποίηση Λειτουργικότητας	Εγχειρίδιο χρήσης	Τεύχος Σχεδίασης	Ευχρηστία	Παρουσίαση	Διαχείριση χρηστών
14	10	4	10	3	4	5

Επικοινωνία

Απορίες Διευκρινήσεις

- eclass <https://eclass.hua.gr/modules/forum/?course=DIT138>
- email tsadimas@hua.gr