

Ειδικά Θέματα Τεχνολογίας Λογισμικού

Οδηγίες για την εκπόνηση της εργασίας

Χειμερινό εξάμηνο ακαδημαϊκού έτους 2024-2025

1. Αντικείμενο της εργασίας

Θα πρέπει να σχεδιάσετε και να αναπτύξετε μια εμπορική (business) εφαρμογή τύπου CRUD, τριών (3) επιπέδων (front-end, business logic, database) σε μικρή κλίμακα. Με τον όρο «μικρή κλίμακα» εννοούμε ότι δεν είναι απαραίτητο να ασχοληθείτε με όλες τις πιθανές λειτουργίες μιας τέτοιας εφαρμογής αλλά να επιλέξετε ορισμένες από αυτές που είναι σημαντικές, κατά την κρίση σας. Εξάλλου ο στόχος της εργασίας δεν είναι μόνο το λογισμικό που θα παραχθεί αλλά και η μεθοδολογία ανάπτυξης.

Μια ιδέα είναι να επανακατασκευάσετε την εφαρμογή που κάνατε στο μάθημα της Τεχν. Λογισμικού. Για όσους δεν επέλεξαν το μάθημα της ΤΛ ή για όσους δυσκολεύονται να βρουν θέμα, μερικά από τα θέματα που προτείνονται (ενδεικτικά) είναι τα ακόλουθα:

1. Online Food Order (Delivery).
2. Διαχείριση Ραντεβού Πλυντηρίου Αυτοκινήτων.
3. E-shop για αθλητικά είδη.
4. Διαχείριση κρατήσεων ξενοδοχείου.
5. Τραπεζική εφαρμογή (τύπου web banking).
6. Διαχείριση κρατήσεων εισιτηρίων κινηματογράφου multiplex.
7. Εφαρμογή ενοικίασης αυτοκινήτων.
8. Διαχείριση ραντεβού σε κουρείο / κομμωτήριο / ιατρείο κλπ.

2. Μεθοδολογία ανάπτυξης

Θα χωριστείτε σε ομάδες με τρία (3) μέλη το πολύ. Η δήλωση των ομάδων γίνεται εδώ,

[[αρχείο δήλωσης 2024_25](#)], και θα πρέπει να συμπληρώσετε όλες τις στήλες του αρχείου. Η σχεδίαση και η ανάπτυξη θα πρέπει να ακολουθεί τις αρχές της ευέλικτης ανάπτυξης δλδ κατ' ελάχιστον, οι απαιτήσεις θα είναι σε μορφή user stories, θα πρέπει να υπάρχει product backlog και ένα board για κάθε sprint. Τα sprints θα έχουν διάρκεια 15 ημέρων και καλό θα είναι κάθε user story να μπορεί να ολοκληρωθεί σε ένα sprint.

Εργαλεία που μπορείτε να χρησιμοποιήσετε για την διαχείριση έργου (στις δωρεάν εκδόσεις τους) είναι ενδεικτικά τα ακόλουθα:

- [Trello](#)
- [Jira](#)
- [Azure DevOps](#)¹

Ημερομηνία έναρξης της εργασίας είναι η Δευτέρα 14/10. Ημερομηνία τερματισμού είναι το τέλος της εξεταστικής. Στα online εργαλεία που θα χρησιμοποιήσετε, θα πρέπει να δώσετε πρόσβαση στο prezerak@uniwa.gr ώστε να μπορώ να ελέγχω την πρόοδο της δουλειάς σας.

3. Repo

Ως version control system, θα χρησιμοποιήσετε το [Git](#). Μια ικανοποιητική λύση, αν δε θέλετε να το δουλεύετε από τη γραμμή εντολών, είναι το παραθυρικό περιβάλλον [SourceTree](#). Ο κώδικας θα πρέπει να ανεβαίνει σε ένα repo στον ιστό (κατά προτίμηση [GitHub](#), [Gitlab](#) ή παρόμοιο). Εφόσον το repo είναι ιδιωτικό, θα πρέπει να δώσετε πρόσβαση στο prezerak@uniwa.gr ώστε να μπορώ να ελέγχω την πρόοδο της δουλειάς σας.

4. Αρχιτεκτονική συστήματος

Όπως αναφέρθηκε παραπάνω, η εφαρμογή σας θα έχει τρία επίπεδα (front-end, business logic, database). Οι περιορισμοί που θα πρέπει να ακολουθήσετε είναι (α) Το front-end θα επικοινωνεί με το business logic με χρήση RESTful web services (β) το business logic να βασίζεται σε αντικειμενοστρεφή γλώσσα προγραμματισμού (γ) η βάση δεδομένων να είναι σχεσιακή και (δ) το business logic να επικοινωνεί με τη βάση δεδομένων μέσω ORM. **Κατά τα λοιπά, μπορείτε να επιλέξετε το software stack της αρεσκείας σας.**

5. Front-end

Θα πρέπει ν' αναπτύξετε ένα SPA front-end χρησιμοποιώντας κάποιο web framework όπως Angular, Vue.js, React, Blazor κλπ.

¹ Έχετε δωρεάν πρόσβαση σε όλα τα εργαλεία της Microsoft, στο Azure market place, μέσω του ακαδημαϊκού σας λογαριασμού.

6. Backend

Θα έχει τη μορφή API σε 3 Layers (controllers, business logic, data) όπως θα πούμε στο μάθημα. Θα πρέπει υποχρεωτικά να χρησιμοποιήσετε Dependency Injection.

7. Βάση δεδομένων

Μπορείτε να επιλέξετε όποια σχεσιακή βάση δεδομένων προτιμάτε. Ενδεικτικά αναφέρονται:

- Ms Sql Server Express / Enterprise
- Oracle
- PostgreSQL
- MariaDB
- MySQL

8. ORM

Η επικοινωνία με τη βάση θα πρέπει να γίνεται μέσω ORM². Υπάρχουν διαθέσιμα διάφορα ORMs ανάλογα με το software stack και τη βάση δεδομένων, όπως (ενδεικτικά):

- .Net¹ Entity Framework
- Java Hibernate
- Ruby ActiveRecord
- Python Django ORM
- PHP Doctrine

9. Authorization / Authentication

Η εφαρμογή σας θα πρέπει να διαθέτει ένα βασικό AA βασισμένο σε username / password. Εφόσον μπορείτε, προσπαθήστε να υλοποιήσετε και πρόσβαση στις λειτουργίες επί τη βάσει ρόλων (Role-based access control - RBAC). Σας συνιστώ την προσέγγιση του JWT αλλά δεν είναι υποχρεωτικό. Εναλλακτικά μπορείτε να υλοποιήσετε μια ροή OAuth σε συνεργασία με τρίτο provider.

² Σε περίπτωση που δεν το καταφέρετε τότε μπορείτε να χρησιμοποιήσετε SQL αλλά θα πρέπει να έχετε εξαντλήσει κάθε άλλη δυνατότητα. Σε μια τέτοια περίπτωση, θα υπάρξει μικρή (αρνητική) επίδραση στη βαθμολογία.

10. Testing

Σε γενικές γραμμές οι CRUD εφαρμογές δε χρειάζονται unit-testing. Επίσης τη στιγμή που γράφετε αυτό το κείμενο, δεν έχετε εντυφώσει ιδιαίτερα στο πως σχεδιάζονται και εκτελούνται unit tests. Ως εκ τούτου θα αναπτύξετε κάποια βασικά αυτοματοποιημένα integration tests στο τέλος του εξαμήνου (καταχρηστικά 😊 γιατί κανονικά πρέπει να τεστάρουμε πριν ολοκληρωθεί το κάθε sprint).

Συνήθως τα frameworks που χρησιμοποιούνται για unit testing, χρησιμοποιούνται και για integration testing. Ενδεικτικά αναφέρονται:

- .Net¹ xUnit / NUnit / MsTest
- Java Junit
- Ruby Ενσωματωμένο
- Python PyTest
- PHP PHPUnit

11. Αξιολόγηση

| Πεδίο | Μονάδες |
|---|---------|
| Υπαρξη και χρήση αποθετηρίου κώδικα. | 2 |
| Υπαρξη εργαλείου για την παρακολούθηση των sprints και χρήση αυτού. | 2 |
| Χρήση Dependency Injection. | 1 |
| Χρήση ORM. | 1 |
| Υπαρξη integration tests. | 2 |
| Μορφή κώδικα / UI Look & feel. | 2 |

Προκειμένου να περάσετε το μάθημα θα πρέπει:

- Να δείξετε ότι έχετε δουλέψει σε όλα τα παραπάνω πεδία (π.χ. αν δεν παραδώσετε εφαρμογή που να τρέχει στον browser, δε θα πάρετε προβιβάσιμο βαθμό στο μάθημα, αν δεν υπάρχει repo το ίδιο κλπ.).
- Να πάρετε προβιβάσιμο βαθμό σε κάθε ένα από τα ως άνω πεδία (π.χ. αν κοπείτε στα integration tests, δε θα πάρετε προβιβάσιμο βαθμό στο μάθημα).