



Μηχανική Λογισμικού 2 (2024 – 2025) - Εργασία Μαθήματος

1. Σκοπός της εργασίας

Σκοπός της εργασίας είναι η εφαρμογή των εννοιών που διδάσκονται στα πλαίσια των μαθημάτων Μηχανική Λογισμικού Ι και Μηχανική Λογισμικού ΙΙ. Πέρα από τις έννοιες που παρουσιάστηκαν στα πλαίσια της Μηχανικής Λογισμικού Ι (καθορισμός απαιτήσεων χρηστών και λογισμικού, καθορισμός προδιαγραφών, σχεδίαση συστήματος), η παρούσα εργασία εστιάζει στη διαχείριση του έργου (project management), τις διαδικασίες συνεχούς ενσωμάτωσης και ανάπτυξης (Continuous Integration – Continuous Deployment, CI/CD), τον έλεγχο του λογισμικού (testing), αλλά και την ενσωμάτωση των μετρικών ποιότητας στη διαδικασία βελτίωσης του έργου.

2. Έργο λογισμικού

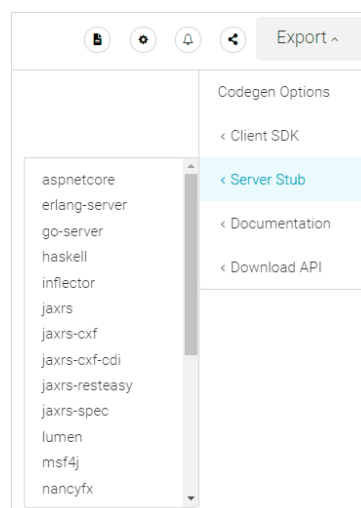
Για την εργασία του μαθήματος θα χρησιμοποιηθούν ως βάση οι εργασίες του μαθήματος "Μηχανική Λογισμικού Ι". Στις εργασίες αυτές είχατε καθορίσει τις απαιτήσεις χρηστών και λογισμικού στο οποίο επιλέξατε να εργαστείτε, και στη συνέχεια σχεδιάσατε τις προδιαγραφές του backend του συστήματός σας (δημιουργία ενός αρχείου swagger), το οποίο περιγράφει τη βασική λειτουργικότητα, καθώς και τις διαθέσιμες κλήσεις που μπορούν να γίνουν σε έναν API server. Στο πλαίσιο της παρούσας εργασίας, μπορείτε να χρησιμοποιήσετε τα swagger αρχεία που είχατε δημιουργήσει στο προηγούμενο μάθημα ή να δημιουργήσετε εκ νέου για μία εφαρμογή της επιλογής σας. Σε κάθε περίπτωση, η περιγραφή του API server θα πρέπει να υπακούει στις ακόλουθες προδιαγραφές:

- Θα πρέπει να υπάρχουν τουλάχιστον 10 (δέκα) διαθέσιμα routes/API calls που ένας εξωτερικός χρήστης μπορεί να χτυπήσει και να περιμένει απάντηση.
- Θα πρέπει να υπάρχει τουλάχιστον ένα route τύπου GET, ένα route τύπου POST, ένα route τύπου PUT και ένα route τύπου DELETE. Ιδανικά, σε ένα resource, ώστε να μπορεί να ελεγχθεί πλήρως η λειτουργικότητα.
- Στη συνολική εφαρμογή, θα πρέπει να εμπεριέχονται τουλάχιστον 3 (τρεις) διαφορετικές οντότητες (π.χ. χρήστες-βιβλία-βιβλιοθήκες), οι οποίες αλληλοεπιδρούν μεταξύ τους στα διάφορα routes του συστήματος (π.χ. ένα route καταγράφει τον δανεισμό ενός βιβλίου από έναν χρήστη, ένα route ρωτάει αν υπάρχει διαθεσιμότητα ενός βιβλίου σε μια βιβλιοθήκη κλπ).



3. Κώδικας

Επόμενο βήμα της διαδικασίας είναι η παραγωγή του JavaScript κώδικα του backend αλλά και του frontend της εφαρμογής σας. Ένας τρόπος είναι να χρησιμοποιήσετε το SwaggerHub¹. Εκεί, αφού κάνετε log in με λογαριασμό της επιλογής σας, μπορείτε να δημιουργήσετε ένα νέο project, στο οποίο θα επικολλήσετε το περιεχόμενο του swagger αρχείου σας. Αφού ελέγξετε την ορθότητα του παραγόμενου API (ελέγχοντας τα πιθανά routes που παράγει το SwaggerHub), μπορείτε να παράξετε και να κατεβάσετε τον αντίστοιχο κώδικα που υλοποιεί το API σας, χρησιμοποιώντας την επιλογή "Export", "Server Stub" και "nodejs-server", όπως φαίνεται στην εικόνα παρακάτω:



Στο zip αρχείο που θα κατέβει στον υπολογιστή σας, μπορείτε να βρείτε την υλοποίηση του API server σας, μαζί με τις απαραίτητες βιβλιοθήκες που θα πρέπει να εγκατασταθούν.

4. Παραδοτέα

Η εργασία αποτελείται από τρία διαφορετικά παραδοτέα, κάθε ένα εκ των οποίων καλύπτει συγκεκριμένα τμήματα από τη διδαχθείσα ύλη του μαθήματος. Στις παρακάτω ενότητες περιγράφεται η δομή, ο σκοπός και τα ζητούμενα κάθε παραδοτέου, καθώς και η ημερομηνία παράδοσής τους. Περισσότερες πληροφορίες για τα αρχεία που θα πρέπει να πρέπει να παραδώσετε σε κάθε περίπτωση θα δίνονται κατά τη διάρκεια του εξαμήνου.

¹ <https://app.swaggerhub.com>



4.1. Παραδοτέο 1 (Παράδοση: 13/11/2024)

Στο πρώτο παραδοτέο, το οποίο έχει ημερομηνία παράδοσης την Τετάρτη 13/11/2024, σκοπός είναι το αρχικό στήσιμο του έργου, τόσο σε επίπεδο κώδικα, όσο και σε επίπεδο διαχείρισης έργου (project management). Για αυτόν τον σκοπό, το συγκεκριμένο παραδοτέο θα πρέπει να περιέχει:

- **Δημιουργία αρχείων swagger και API**

Χρησιμοποιήστε τα υπάρχοντα αρχεία swagger (από την εργασία στο μάθημα "Μηχανική Λογισμικού Ι") ή δημιουργήστε ένα νέο διασφαλίζοντας ότι το API συμμορφώνεται με τις γενικές απαιτήσεις.

- **Εφαρμογή API**

Μετατρέψτε την περιγραφή του API (μέσω του αρχείου swagger) σε πραγματικό server, χρησιμοποιώντας το SwaggerHub, το οποίο περιλαμβάνει τόσο το backend όσο και το frontend με βάση το API. Στη συνέχεια, ανεβάστε τον παραγόμενο κώδικα σε ένα νέο αποθετήριο στο GitHub², στο οποίο θα πρέπει να προστεθούν όλα τα μέλη της ομάδας, καθώς και οι διδάσκοντες (usernames: asymeon, karanikio, dcnatsos, GSiach13).

- **Διαχείριση έργου**

Για το Usereq project σας θα δημιουργήσετε στην πλατφόρμα της Cyclopt³ ένα project, σε ομάδα αντίστοιχη με αυτήν που έχετε δηλώσει στο elearning. Στο παραπάνω project θα περάσουν τα υπάρχοντα features του συστήματός σας (η διαδικασία του Integration θα συζητηθεί στο μάθημα), ενώ στη συνέχεια θα πρέπει να περάσετε και τα μελλοντικά tasks, αυτά δηλαδή που απαιτούνται για την υλοποίηση της εργασίας. Όλα τα tasks θα πρέπει να χρονοπρογραμματιστούν με βάση τη λογική που θα συζητήσουμε στο μάθημα. Θυμίζουμε πως η εργασία αποτελεί ένα agile project, κατά συνέπεια το Cyclopt board του project σας πρέπει να παραμένει ενημερωμένο καθ' όλη τη διάρκεια του εξαμήνου (assessment θα γίνεται σε διεβδομαδιαία βάση).

- **Βαθμολόγηση**

Η βαθμολόγηση του πρώτου παραδοτέου θα βασιστεί στους παραπάνω άξονες. Πιο συγκεκριμένα, θα αξιολογηθεί η τελική έκδοση του swagger αρχείου που περιγράφει την εφαρμογή σας (και η τήρηση των αξόνων που έχουν τεθεί), η δημιουργία ενός Github repo για την εναπόθεση του κώδικα και η δημιουργία των κατάλληλων repo branches (περισσότερα στο μάθημα), αλλά και η δημιουργία και ενημέρωση του πλάνου διαχείρισης του έργου.

² <https://github.com>

³ <https://platform.cyclopt.com>



4.2. Παραδοτέο 2 (Παράδοση: 11/12/2024)

Στο δεύτερο παραδοτέο, το οποίο έχει ημερομηνία παράδοσης την Τετάρτη 11/12/2024, σκοπός είναι να οριστούν και να εκτελεστούν σωστά οι διαδικασίες ελέγχου και ανάπτυξης του project σας. Για αυτόν τον σκοπό, το συγκεκριμένο παραδοτέο θα πρέπει να περιέχει:

- **Testing strategy implementation**

Θα πρέπει να αναπτύξετε τα κατάλληλα tests τα οποία διασφαλίζουν την ορθή εκτέλεση κάθε πιθανού τμήματος κώδικα σε κάθε πιθανή διαδρομή/λειτουργία του κώδικά σας. Για αυτόν τον σκοπό, θα πρέπει, χρησιμοποιώντας τα εργαλεία/frameworks που θα παρουσιαστούν στο πλαίσιο του μαθήματος, να εξασφαλίσετε ότι τα unit tests καλύπτουν τον κώδικά σας σε ποσοστό τουλάχιστον 90%. Ταυτόχρονα, θα πρέπει να υλοποιήσετε και τα αντίστοιχα acceptance tests (UI tests).

- **CI/CD**

Θα πρέπει να δημιουργήσετε τα κατάλληλα CI/CD pipelines για να διασφαλίσετε ότι ο server σας λειτουργεί συνεχώς, αλλά και να διασφαλίσετε πώς κάθε νέα αλλαγή/feature ελέγχεται πλήρως από τα tests που έχετε δημιουργήσει και ενσωματώνεται ομαλά στη λειτουργία του server.

- **Βαθμολόγηση**

Κριτήριο βαθμολόγησης του παραδοτέου αποτελεί η σωστή και όσο το δυνατόν πιο πλήρης υλοποίηση ενός μηχανισμού ελέγχου της ορθότητας του κώδικα (testing strategy). Για την αξιολόγηση θα ληφθεί υπόψιν η ορθότητα των tests που έχουν ορισθεί, καθώς και το ποσοστό κάλυψης του κώδικα από αυτά (μετρικές όπως το testing coverage). Επίσης, στη βαθμολόγηση συμπεριλαμβάνεται ο σωστός ορισμός της διαδικασίας ci/cd, χρησιμοποιώντας τα κατάλληλα GitHub actions, έτσι ώστε η πλατφόρμα να λειτουργεί κανονικά και να ανανεώνεται πλήρως σε κάθε επόμενη αλλαγή στον κώδικα (commit).

4.3. Παραδοτέο 3 (Παράδοση: 08/01/2024)

Στο τρίτο παραδοτέο, το οποίο έχει ημερομηνία παράδοσης την Τετάρτη 08/01/2024, σκοπός είναι η ενσωμάτωση των μετρικών ποιότητας στη διαδικασία βελτίωσης του έργου. Για αυτόν τον σκοπό, το συγκεκριμένο παραδοτέο θα πρέπει να περιέχει:

- **Code quality assessment**

Χρησιμοποιώντας την πλατφόρμα και τα εργαλεία της Cyclot, θα πρέπει να γίνει μια αποτίμηση της ποιότητας του κώδικά σας (σε όλα τα ενεργά branches). Χρησιμοποιώντας το feedback της πλατφόρμας σε κάθε σας commit, θα πρέπει να θέσετε σαφείς στόχους



βελτίωσης συγκεκριμένων χαρακτηριστικών ποιότητας και να προχωρήσετε σε διορθώσεις/αλλαγές στον κώδικά σας.

- **Βαθμολόγηση**

Ο τρίτος άξονας βαθμολόγησης της εργασίας αφορά τη βελτίωση των μετρικών ποιότητας (στατικής ανάλυσης) και, συγκεκριμένα, τις μετρικές που εξετάζονται και αναλύονται στην πλατφόρμα Cyclort. Με βάση τις αναφορές και τις προτάσεις της πλατφόρμας θα πρέπει να αναδιαμορφώσετε κατάλληλα τον κώδικα του server σας, έτσι ώστε να βελτιώσετε τις εν λόγω μετρικές κατά το βέλτιστο δυνατό. Φυσικά, οι αλλαγές που θα κάνετε στον κώδικα δεν θα πρέπει να επηρεάζουν τη φυσιολογική λειτουργία της εφαρμογής σας. Στο κομμάτι αυτό θα βοηθήσει ιδιαίτερα η συνεχής ανανέωση της πλατφόρμας μέσω του ci/cd (ούτως ώστε να μπορείτε να ελέγχετε live την πλατφόρμα), καθώς και τα tests, τα οποία θα πρέπει να μπορούν να εντοπίσουν πιθανά σφάλματα που θα επηρεάσουν τη λειτουργικότητα της πλατφόρμας.

Για περισσότερες πληροφορίες:

Ανδρέας Λ.
Συμεωνίδης

Καθηγητής

symeonid@ece.auth.gr

Καρανικιώτης Θωμάς

Μεταπτυχιακός
Φοιτητής

karanikio@ece.auth.gr

Νάτσος Δημοσθένης

Μεταπτυχιακός
Φοιτητής

dcnatsos@ece.auth.gr

Σιαχάκης Γεώργιος

Μεταπτυχιακός
Φοιτητής

gsiacha@ece.auth.gr