

Εργασία στο Μάθημα "Τεχνολογία Λογισμικού"

ΕΜΠ - Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Χειμερινό εξάμηνο 2024-2025

Διδάσκων: Β. Βεσκούκης, Καθ.ΕΜΠ

Θεματικό πεδίο

Το θεματικό πεδίο της εργασίας είναι η διαλειτουργικότητα στα πληροφοριακά συστήματα διοδίων αυτοκινητοδρόμων με διαφορετικά συστήματα αυτόματης διέλευσης, καθώς και η ανάλυση δεδομένων διελεύσεων από διαφορετικές οπτικές γωνίες. Τα ηλεκτρονικά συστήματα διέλευσης διοδίων των οδών: aodos.gr, gefyra.gr, egnatia.eu, kentrikiodos.gr, moreas.com.gr, neaodos.gr, olympiaodos.gr διαλειτουργούν, δηλαδή το καθένα επιτρέπει τη διέλευση με τον πομποδέκτη οποιουδήποτε άλλου.

Με τον τρόπο αυτό δημιουργούνται οφειλές μεταξύ εταιριών διαχείρισης διοδίων, οι οποίες πρέπει να συμψηφίζονται. Μια τέτοια οφειλή της εταιρίας Α προς την εταιρία Β, δημιουργείται όταν ένα όχημα-συνδρομητής του αυτοκινητοδρόμου Α, περάσει από διόδια του αυτοκινητοδρόμου Β. Τότε η εταιρία του Α οφείλει στην εταιρία του Β το αντίτιμο της διέλευσης. Οι οφειλές των Α και Β προς αλλήλους συμψηφίζονται και καταβάλλεται η διαφορά.

Επιπροσθέτως της εκκαθάρισης, τα δεδομένα που καταγράφονται κατά τις διελεύσεις οχημάτων από τα διόδια, μπορούν να διατεθούν και να αναλυθούν από ενδιαφερόμενα εμπλεκόμενα μέρη (stakeholders), προκειμένου να εξαχθούν συμπεράσματα σχετικά με τη χρήση των αυτοκινητοδρόμων, τη συμπεριφορά οδηγών, τις αποσβέσεις επενδύσεων, προβλέψεις μεγεθών κ.ά.

Το λογισμικό που καλείστε να κατασκευάσετε είναι αυτό που θα εκτελείται σε έναν φορέα ανεξάρτητο από τους λειτουργούς των αυτοκινητοδρόμων, προκειμένου αυτός να υλοποιήσει τη διαχείριση της διαλειτουργικότητας, καθώς και να προσφέρει υπηρεσίες διάθεσης και ανάλυσης δεδομένων χρήσης των αυτοκινητοδρόμων. Το λογισμικό αυτό θα λαμβάνει δεδομένα διελεύσεων από λειτουργούς και θα τα αποθηκεύει κατάλληλα. Στη συνέχεια αφ' ενός θα υπολογίζει, σε τακτά διαστήματα, τους μεταξύ των λειτουργών οικονομικούς συμψηφισμούς και αφ' ετέρου θα διαθέτει δεδομένα για ανάλυση σε τρίτα μέρη και θα πραγματοποιεί και το ίδιο κάποιες τέτοιες αναλύσεις.

Σημειώνεται ότι το αντικείμενο "διόδια-διαλειτουργικότητα" αποτελεί μόνο αφορμή για το θέμα της εργασίας, η οποία δεν συσχετίζεται με οποιοδήποτε πραγματικό σύστημα. Η εκφώνηση της εργασίας αποτελείται από δύο τμήματα. Το δεύτερο τμήμα (part2) θα ανακοινωθεί σε κατάλληλο χρόνο.

Διατύπωση ζητουμένου

Το αντικείμενο της εργασίας είναι η ανάπτυξη ενός πληροφοριακού συστήματος στο οποίο θα έχουν πρόσβαση τα εμπλεκόμενα μέρη (stakeholders), για την υλοποίηση της διαλειτουργικότητας στα διόδια όπως γενικά περιεγράφηκε και με την παραδοχή ότι ισχύουν τα εξής:

- Κάθε λειτουργός αυτοκινητοδρόμου χρησιμοποιεί το δικό του πληροφοριακό σύστημα διαχείρισης διελεύσεων, το οποίο θα πρέπει να υποστεί όποιες τροποποιήσεις απαιτηθούν.
- Σε όλα τα σημεία διελεύσεων είναι διαθέσιμες οι λειτουργίες "provider" (σε ποιον πάροχο ανήκει ένας πομποδέκτης), "balance" (υπόλοιπο χρημάτων στον πομποδέκτη) και "pass" (διέλευση και καταγραφή της οφειλής).
- Η φόρτιση ενός πομποδέκτη ("load") είναι δυνατή μόνο στο δίκτυο στο οποίο αυτός ανήκει.

Η εργασία θα περιλαμβάνει την αναγνώριση και εξειδίκευση - προδιαγραφή των απαιτήσεων, την αρχιτεκτονική, τη λεπτομερή σχεδίαση, την υλοποίηση επιλεγμένων λειτουργιών, τον έλεγχο, καθώς και την τεκμηρίωση όλων αυτών. Οι εργασίες αυτές, καθώς και η διοίκηση του έργου (project management), θα γίνουν με χρήση κατάλληλων εργαλείων, συμπεριλαμβανομένων εργαλείων AI, τα οποία θα υποδειχθούν.

Η αναγνώριση των εμπλεκόμενων (stakeholders) και των λειτουργιών που θα υλοποιεί το σύστημα θα γίνει μέσω συζητήσεων ("workshops") στο μάθημα. Με το πέρας αυτών, κατάλογος με τους εμπλεκόμενους και τις λειτουργίες θα γίνει διαθέσιμος στο χώρο του μαθήματος στο helios. Κάποιες λειτουργίες θα υλοποιηθούν από όλες τις ομάδες (προδιαγραφές API και CLI), ενώ κάποιες άλλες επιπλέον, θα επιλέγονται ελεύθερα από κάθε ομάδα για την υλοποίηση περιπτώσεων χρήσης, σύμφωνα με τον αριθμό των μελών της.

Τα τμήματα του πληροφοριακού συστήματος που θα κατασκευάσετε έχουν ως εξής:

1. Ένα υποσύστημα back-end, το οποίο θα υποστηρίζει λειτουργίες διαχείρισης δεδομένων και υπολογισμών, με αρχιτεκτονική της επιλογής σας. Οι λειτουργίες αυτές θα διατίθενται μέσω μιας προγραμματιστικής διεπαφής REST API, με προδιαγραφές που θα σας δοθούν στο δεύτερο τμήμα της εκφώνησης.
2. Μία διεπαφή CLI (Command Line Interface) για προσπέλαση δεδομένων. Η γλώσσα της διεπαφής CLI θα είναι η αγγλική. Η διεπαφή CLI θα λειτουργεί ως client του REST API που θα παρέχεται από το back-end υποσύστημα.
3. Μία δικτυακή εφαρμογή (web application), η οποία θα προσφέρει στο χρήστη βασικές δυνατότητες εκτέλεσης υπολογισμών και παρουσίασης των δεδομένων, όπως ενδεικτικά: πίνακες, διαγράμματα, οπτικοποίηση σε χάρτη κ.ά.. Η εφαρμογή αυτή θα αποτελεί το frontend του συστήματος, το οποίο επίσης θα είναι client του REST API. Θα πρέπει να υποστηρίζεται το πρωτόκολλο HTTPS για όλες τις διεπαφές, μέσω self-signed certificate.

Ομάδες εργασίας

Η εργασία συνιστάται να γίνει σε **ομάδες των 4 ατόμων**, οι οποίες θα υλοποιήσουν έναν πλήρη κύκλο ανάπτυξης λογισμικού (ανάλυση απαιτήσεων, σύνταξη προδιαγραφών, αρχιτεκτονική, σχεδιασμός αρχιτεκτονική, υλοποίηση, έλεγχος, εγκατάσταση και λειτουργία). Επιτρέπονται ομάδες με λιγότερα από 4 και με έως 6 άτομα, με κατάλληλη προσαρμογή των παραδοτέων, όπως φαίνεται στη συνέχεια. Για τη συνεργατική διαχείριση της τεκμηρίωσης και του πηγαίου κώδικα, καθώς και για τις εργασίες διαχείρισης έργου, είναι υποχρεωτική για τις ομάδες η χρήση του Github.

Η συγκρότηση των ομάδων θα πρέπει να έχει ολοκληρωθεί το αργότερο μέχρι τις **18.10.2024**, επί βαθμολογική ποινή, όπως αναφέρεται στις διαφάνειες και στο [helios \(σελ 5, 6\)](#).

Τεχνικές απαιτήσεις - εργαλεία

Για το **backend** μπορεί να χρησιμοποιηθεί οποιοδήποτε development stack με python ή js με node/express. Επιτρέπονται και άλλες επιλογές (java, .NET) με αιτιολογημένη δήλωσή σας. Για **διαχείριση δεδομένων** ένα εκ των MySQL, MariaDB, PostgreSQL, Mongo / Atlas, Elastic Search. Για **frontend** η επιλογή frameworks και τεχνολογιών είναι ελεύθερη, υπό το πρίσμα του περιορισμένου μεγέθους της εφαρμογής στο πλαίσιο της εργασίας.

Ανεξάρτητα από το development stack, θα χρησιμοποιηθούν υποχρεωτικά τα ακόλουθα εργαλεία:

- **Github** για διαχείριση κώδικα. Καλείστε να ακολουθήσετε επακριβώς τις οδηγίες που θα ανακοινωθούν στο μάθημα και το helios.
- **Visual Paradigm 17.2 CE** για μοντελοποίηση με χρήση UML. Μπορείτε επίσης να χρησιμοποιήσετε την trial έκδοση Visual Paradigm enterprise. Δεν επιτρέπεται η χρήση άλλων εργαλείων παραγωγής διαγραμμάτων UML, συμπεριλαμβανομένης της δικτυακής έκδοσης Visual Paradigm Online, εκτός όσων ενδεχομένως υποδειχθούν για ειδικό σκοπό.
- **GitHub projects** για τη διαχείριση έργων.
- **AI tools** για υποστήριξη σε όλες τις φάσεις του κύκλου ζωής, με υποχρεωτική καταγραφή και υποβολή μεταδεδομένων και διαλόγων (prompts & replies). Εκτός των εμπορικών εργαλείων AI (ChatGPT, Gemini, Copilot), θα δοθούν οδηγίες για εργαλεία AI ανοικτού κώδικα.

Παραδοτέα

Πίνακας παραδοτέων

ΠΑΡΑΔΟΤΕΟ	Ομάδες (άτομα)		
	1-2	3-4	5-6
Τεκμηρίωση (35%)			
Εγγραφο SRS - Software Requirements Specification	1 use case	2 use cases	3 use cases
Data design diagrams (ER ή UML class)	NAI		
UML Activity / State diagrams	NAI		
UML class & API diagrams	NAI		
UML component & deployment diagrams	NAI		
Διαγράμματα UML Deployment	NAI		
Υλοποίηση (40%)			
Διαχείριση δεδομένων και υπηρεσίες (backend)	NAI		
Data dump	NAI		
RESTful API	NAI		
API documentation	NAI		
Command line interface	NAI		
Επικοινωνία με τον χρήστη (frontend)	1 use case	2 use cases	3 use cases
Testing (6%)			
API functional tests	NAI	NAI	NAI
CLI testing		NAI	NAI
Εργαλεία (19%)			
Χρήση εργαλείων διαχείρισης κώδικα	NAI		
Χρήση εργαλείων διαχείρισης έργων	NAI		
Χρήση εργαλείων AI	NAI		

Μορφότυποι παραδοτέων

ΠΑΡΑΔΟΤΕΟ	Μορφότυπος	Όνομα αρχείου	Παράδοση 29.11.2024	Τελική παράδοση
Τεκμηρίωση				
Εγγραφο SRS - Software Requirements Specification	επεξεργάσιμο αρχείο (όχι pdf)	srs-softeng24-XX.zip	helios	helios & github.com/ntua/softeng24-XX/doc
Data design diagrams (ER ή UML class)	Ενα αρχείο Visual Paradigm ανά εργασία, με όλα τα διαγράμματα (.vpp)	softeng24-XX.vpp		
UML Activity / State diagrams				
UML class & API diagrams				
UML component & deployment diagrams				
Διαγράμματα UML Deployment				
Υλοποίηση				
Διαχείριση δεδομένων και υπηρεσίες (backend)	Πηγαίος κώδικας	(σύμφωνα με το περιβάλλον ανάπτυξης)	-	github.com/ntua/softeng24-XX/backend
Data dump	Αρχεία SQL ή json		-	github.com/ntua/softeng24-XX/database
RESTful API	Πηγαίος κώδικας		-	github.com/ntua/softeng24-XX/api
API documentation	Postman scripts, OpenAPI yaml		-	
Command line interface	Πηγαίος κώδικας		-	github.com/ntua/softeng24-XX/cli
Επικοινωνία με τον χρήστη (frontend)			-	github.com/ntua/softeng24-XX/frontend
Testing				
API functional tests	test scripts	(σύμφωνα με το περιβάλλον ανάπτυξης)	-	github.com/ntua/softeng24-XX/test
CLI testing			-	
Εργαλεία				
Χρήση εργαλείων διαχείρισης κώδικα	-	-	github.com/ntua/softeng24-XX	
Χρήση εργαλείων διαχείρισης έργων	-	-	github.com/ntua/softeng24-XX/projects	
Χρήση εργαλείων AI	-	-	github.com/ntua/softeng24-XX/ailog και ailog.softlab.ntua.gr	

Προθεσμίες

Παράδοση 1: 29.11.2024. Έγγραφο SRS και διαγράμματα όπως φαίνεται στον παραπάνω πίνακα, καταγραφή χρήσης GitHub, GitHub projects και ailog.

Τελική παράδοση: πριν από τη γραπτή εξέταση. Το σύνολο της τεκμηρίωσης (με δυνατότητα επικαιροποίησης από την παράδοση 1) και του πηγαίου κώδικα.

Βαθμολογία και αναλυτικές βαρύτητες

Η βαθμολογία προκύπτει κατά **50%** από τη γραπτή εξέταση και κατά **50%** από την εργασία, η οποία παρουσιάζεται προφορικά από όλα τα μέλη της ομάδας. Οι επιμέρους βαρύτητες για την εργασία φαίνονται στον παρακάτω πίνακα. **Για τη λήψη προβιβάσιμου βαθμού στο μάθημα απαιτείται προβιβάσιμος βαθμός σε αμφότερες την εργασία και την τελική εξέταση.**

Τεκμηρίωση	35%	
Έγγραφο SRS (περιεχόμενα)	10%	Σχετικές βαρύτητες επί της τεκμηρίωσης.
Αυτόματη παραγωγή SRS	10%	
Σχεδίαση δεδομένων	10%	
Διαγράμματα UML activity, state	10%	
Διαγράμματα UML class, API	10%	
Διαγράμματα UML sequence	10%	
Διαγράμματα UML component	10%	
Διαγράμματα UML deployment	5%	
Συνέπεια μοντέλου UML	25%	
Υλοποίηση	40%	
Backend & database	10%	
API documentation	5%	
REST APIs	10%	Αυτόματη βαθμολόγηση. Μείωση του μέγιστου βαθμού κατά 30% σε διαφορετική περίπτωση.
Comand Line Interface (CLI)	5%	
Frontend	10%	
Έλεγχος	6%	
API testing	3%	
CLI unit/func testing	3%	
Εργαλεία	19%	
Διαχείριση κώδικα GitHub	5%	Λαμβάνονται υπόψη οι κινήσεις, τα μέλη της ομάδας που τις πραγματοποιούν, καθώς και ο χρόνος (timestamp).
Διαχείριση έργων GitHub PM	4%	
AI	10%	Αυτόματη βαθμολόγηση 0/10 ή 10/10 (binary).
ΣΥΝΟΛΟ	100%	

Η βαθμολογία των μελών μιας ομάδας ενδέχεται να διαφοροποιηθεί ανάλογα με τα καταγεγραμμένα δεδομένα στα helios, Github και ailog.

Ο τύπος σύμφωνα με τον οποίο υπολογίζεται η τελική βαθμολογία της εργασίας είναι **Βαθμός_εργασίας=round(sumproduct(partialGrades;weights);1)** όπου **partialGrades** είναι οι επιμέρους βαθμολογίες (0-10) και **weights** οι βαρύτητες.

Κανονικοποίηση, εφόσον απαιτείται, γίνεται στη γραπτή εξέταση, σε επίπεδο θέματος.

Ο τελικός βαθμός του μαθήματος προκύπτει με απλή στρογγυλοποίηση στην ακέραια μονάδα, του όρου **(Βαθμός_εργασίας + Βαθμός_γραπτού)/2**. Για παράδειγμα 6.5→7 και 6.4→6.