

# Εργασία στο Μάθημα της Τεχνολογίας Λογισμικού

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Εθνικό Μετσόβιο Πολυτεχνείο

Χειμερινό εξάμηνο 2021-2022

Διδάσκων: Β. Βεσκούκης

## Θεματικό πεδίο

Το θεματικό πεδίο της εργασίας είναι το πρόβλημα της διαλειτουργικότητας στα διόδια αυτοκινητοδρόμων με διαφορετικά συστήματα αυτόματης διέλευσης. Ήδη, από το Νοέμβριο του 2020, τα ηλεκτρονικά συστήματα διέλευσης διοδίων διαλειτουργούν, δηλαδή το καθένα επιτρέπει τη διέλευση με τον πομποδέκτη οποιουδήποτε άλλου. Τα συστήματα αυτά είναι των οδών: aodos.gr, gefyra.gr, egnatia.eu, kentrikiodos.gr, moreas.com.gr, neaodos.gr, olympiaodos.gr. Το γεγονός αυτό αποτέλεσε την αφορμή για το θέμα της εργασίας, η οποία με κανένα τρόπο δεν συσχετίζεται με το πραγματικό σύστημα.

Η διαλειτουργικότητα αυτή δημιουργεί οφειλές μεταξύ των εταιριών διαχείρισης, οι οποίες πρέπει να εκκαθαρίζονται. Μια οφειλή μιας εταιρίας διαχείρισης Α προς μία εταιρία Β, δημιουργείται όταν ένα όχημα-κάτοχος πομποδέκτη του αυτοκινητοδρόμου Α, περάσει από διόδια του αυτοκινητοδρόμου Β. Τότε η εταιρία διαχείρισης του Α οφείλει στην εταιρία διαχείρισης του Β το αντίτιμο της διέλευσης. Επειδή στη γενική περίπτωση συμβαίνει και το αντίστροφο, δηλ οχήματα με πομποδέκτη του αυτοκινητόδρομου Β διέρχονται από διόδια του Α, τότε δημιουργούνται και οφειλές του Β προς τον Α. Τελικά, οι οφειλές των Α και Β προς αλλήλους συμψηφίζονται και καταβάλλεται μόνο η διαφορά από την εταιρία διαχείρισης του αυτοκινητόδρομου οι συνδρομητές του οποίου έκαναν διελεύσεις μεγαλύτερης αξίας στον άλλο αυτοκινητόδρομο.

Το λογισμικό που καλείστε να κατασκευάσετε είναι αυτό που θα εκτελείται σε έναν φορέα ανεξάρτητο από τους λειτουργούς των αυτοκινητόδρομων που αναφέρθηκαν, προκειμένου να υλοποιήσει τη διαχείριση της διαλειτουργικότητας, όπως περιγράφηκε. Το λογισμικό αυτό θα λαμβάνει με κατάλληλο τρόπο τα δεδομένα από τους λειτουργούς και θα υπολογίζει, σε συμφωνημένες χρονικές στιγμές, τις μεταξύ τους οφειλές μετά τους συμψηφισμούς. Δεχόμαστε ότι ισχύουν τα εξής:

- Οι λειτουργοί των αυτοκινητόδρομων δεν χρησιμοποιούν το ίδιο σύστημα διαχείρισης διελεύσεων. Καθένα χρησιμοποιεί ένα δικό του σύστημα και θα πρέπει να πραγματοποιήσει το ίδιο όποιες τροποποιήσεις - επεκτάσεις απαιτηθούν. Το σύστημα διαχείρισης διελεύσεων του κάθε αυτοκινητοδρόμου είναι εκτός του αντικειμένου ευθύνης της εφαρμογής που αναπτύσσουμε.
- Σε όλα τα σημεία διελεύσεων είναι διαθέσιμες οι λειτουργίες "provider" (δήλωση σε ποιον πάροχο ανήκει ένας πομποδέκτης), "balance" (υπόλοιπο χρημάτων στον πομποδέκτη) και "pass" (διέλευση και καταγραφή της οφειλής).
- Η φόρτιση ενός πομποδέκτη με χρήματα ("load") είναι δυνατή μόνο στο δίκτυο στο οποίο αυτός ανήκει.

## Διατύπωση ζητουμένου

Το αντικείμενο της εργασίας του μαθήματος είναι η ανάπτυξη ενός πληροφοριακού συστήματος για την υλοποίηση της διαλειτουργικότητας στα διόδια, όπως περιγράφηκε, στο οποίο υποθέτουμε ότι θα έχουν πρόσβαση οι διάφοροι εμπλεκόμενοι (stakeholders). Η εργασία θα περιλαμβάνει την αναγνώριση και προδιαγραφή των απαιτήσεων, την αρχιτεκτονική και λεπτομερή σχεδίαση, την υλοποίηση επιλεγμένων λειτουργιών, τον έλεγχο και, ασφαλώς, την τεκμηρίωση όλων αυτών. Η προδιαγραφή των απαιτήσεων και η σχεδίαση, η διαχείριση του κώδικα, το testing, καθώς και η διοίκηση του έργου, θα γίνουν με χρήση κατάλληλων εργαλείων.

Η αναγνώριση των εμπλεκόμενων (stakeholders) και των λειτουργιών που θα περιλαμβάνει το σύστημα θα γίνει μέσω συζητήσεων στο μάθημα. Με την ολοκλήρωση αυτών, κατάλογος με τους εμπλεκόμενους και τις λειτουργίες θα είναι διαθέσιμος στο χώρο του μαθήματος στο [helios.ntua.gr](http://helios.ntua.gr). Από τον κατάλογο αυτό κάποιες λειτουργίες θα υλοποιηθούν από όλες τις ομάδες, ενώ κάποιες άλλες θα επιλέγονται από κάθε ομάδα, σύμφωνα με τον αριθμό των μελών της και οδηγίες που θα ανακοινωθούν.

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

1. Ένα υποσύστημα back-end, το οποίο θα υποστηρίζει λειτουργίες διαχείρισης δεδομένων μεταξύ των διαφορετικών λειτουργιών των αυτοκινητόδρομων. Οι λειτουργίες αυτές θα διατίθενται μέσω ενός REST API, οι προδιαγραφές του οποίου θα σας δοθούν.
2. Μία εφαρμογή CLI (Command Line Interface) για τις λειτουργίες προσπέλασης δεδομένων. Η εφαρμογή θα λειτουργεί ως client του REST API που παρέχεται από το back-end υποσύστημα, προσφέροντας στο χρήστη της τη δυνατότητα να εκτελεί λειτουργίες στα δεδομένα.
3. Μία δικτυακή εφαρμογή (Web Application), η οποία θα προσφέρει στο χρήστη δυνατότητες παρουσίασης των δεδομένων, όπως ενδεικτικά: πίνακες, διαγράμματα, οπτικοποίηση σε χάρτη. Η εφαρμογή αυτή θα λειτουργεί σε περιβάλλον Web browser και θα αποτελεί το front-end του συστήματος, το οποίο επίσης θα είναι client του REST API.

## Ομάδες εργασίας

Η εργασία θα γίνει σε **ομάδες των 5 ατόμων**, οι οποίες θα υλοποιήσουν τον πλήρη κύκλο ανάπτυξης του συστήματος (ανάλυση απαιτήσεων, σύνταξη προδιαγραφών, σχεδιασμός και αρχιτεκτονική, υλοποίηση και έλεγχοι αποδοχής, εγκατάσταση και λειτουργία). Επιτρέπονται ομάδες με λιγότερα από 5 άτομα και με έως 6 άτομα, με κατάλληλη προσαρμογή των παραδοτέων, όπως εξηγείται στη συνέχεια.

Για τη συνεργατική διαχείριση των εκδόσεων της τεκμηρίωσης και του πηγαίου κώδικα, είναι υποχρεωτική η χρήση του Github. Επίσης, είναι υποχρεωτική η χρήση και άλλων εργαλείων όπως περιγράφεται παρακάτω. Η συγκρότηση των ομάδων θα πρέπει να έχει ολοκληρωθεί το αργότερο μέχρι τις **27.10.2021**.

## Ελάχιστες κοινές τεχνικές προδιαγραφές

Το σύστημα που θα αναπτύξετε θα πρέπει να υποστηρίζει τα εξής χαρακτηριστικά (ελάχιστες κοινές προδιαγραφές):

1. Οι λειτουργίες που θα υλοποιηθούν θα επιλεγούν από κατάλογο που θα ανακοινωθεί στο [helios.ntua.gr](http://helios.ntua.gr), σύμφωνα με το πλήθος των μελών κάθε ομάδας και λοιπές οδηγίες που θα δοθούν.
2. Το υποσύστημα back-end θα παρέχει κατάλληλο REST API, το οποίο προαιρετικά μπορεί να είναι συμβατό με το πρότυπο OpenAPI 3.0. Για τα endpoints που θα αναπτύξετε θα ανακοινωθούν προδιαγραφές κατά τη διάρκεια υλοποίησης της εργασίας και θα είναι κοινά για όλες τις ομάδες.
3. Η γλώσσα των διεπαφών χρήστη στην εφαρμογή CLI θα είναι η αγγλική. Η γλώσσα των διεπαφών χρήστη για τις άλλες εφαρμογές, όπου απαιτείται, θα είναι η ελληνική ή η αγγλική.
4. Κάθε ομάδα θα πρέπει να κάνει χρήση ενός εργαλείου αυτοματισμού του «χτισίματος» του λογισμικού (build automation).
5. Κάθε ομάδα θα συντάξει σενάρια ελέγχου και θα ενσωματώσει την εκτέλεση των αντίστοιχων δοκιμών με αυτόματο τρόπο για τις λειτουργίες του back-end υποσυστήματος.
6. Θα πρέπει να υποστηρίζεται το πρωτόκολλο HTTPS για όλες τις διεπαφές, μέσω self-signed certificate.

Οι απαιτήσεις αυτές εξειδικεύονται ανάλογα με το πλήθος των μελών της ομάδας, όπως φαίνεται στον πίνακα παραδοτέων στην επόμενη ενότητα.

## Τεχνικές απαιτήσεις - εργαλεία

Θα χρησιμοποιηθούν τα ακόλουθα εργαλεία:

- **Github** ως περιβάλλον διαχείρισης κώδικα. Παρακαλείστε να μην δημιουργήσετε κάποιο repository πριν ανακοινωθούν οι σχετικές οδηγίες.
- **Visual Paradigm 16.3 CE** ως εργαλείο παραγωγής διαγραμμάτων UML. Μπορείτε επίσης να χρησιμοποιήσετε την trial έκδοση Visual Paradigm enterprise (δωρεάν για ένα μήνα).
- **YouTrack** για τη διαχείριση έργων. Το YouTrack προσφέρεται ως δικτυακή εφαρμογή από το εργαστήριο Λογισμικού. Σχετικά θα υπάρξουν αναλυτικές οδηγίες.
- **UpSource** για reviews πηγαίου κώδικα. Το UpSource προσφέρεται ως δικτυακή εφαρμογή από το εργαστήριο Λογισμικού. Σχετικά θα υπάρξουν αναλυτικές οδηγίες.

Development stack που μπορούν να χρησιμοποιηθούν

- **Υλοποίηση κώδικα:** Python, javascript με nodejs/express. Επιτρέπονται και άλλες επιλογές (java, .net) με δήλωσή σας.
- **Διαχείριση δεδομένων:** ένα εκ των MySQL, MariaDB, PostgreSQL, Mongo, Elastic Search
- **Frontend:** η επιλογή frameworks και τεχνολογιών για frontend (π.χ. βιβλιοθήκες JS για διαγράμματα / γραφήματα στο web), είναι ελεύθερη.

# Παραδοτέα

## Λίστα παραδοτέων

Η λίστα των παραδοτέων του μαθήματος φαίνεται στον παρακάτω πίνακα.

ΠΑΡΑΔΟΤΕΟ	Ομάδες 1-2 ατόμων	Ομάδες 3-4 ατόμων	Ομάδες 5-6 ατόμων
Τεκμηρίωση			
Εγγραφο StRS - Stakeholder Requirements Specification	για 1 stakeholder	για 2 stakeholders	για 3 stakeholders
Εγγραφο SRS - Software Requirements Specification	1 use case	2 use cases	3 use cases
Διαγράμματα ER ή NoSQL JSON schemas	NAI		
Διαγράμματα UML Activity / State	αντίστοιχα με τα use cases		
Διαγράμματα UML Sequence	αντίστοιχα με τα use cases		
Διαγράμματα UML Deployment	NAI		
Διαγράμματα UML Component	NAI		
Υλοποίηση			
Εισαγωγή, διαχείριση και πρόσβαση σε δεδομένα (backend)	NAI		
Database dump (sql ή json)	NAI		
RESTful API	NAI		
API documentation - demo	NAI		
Command line interface (CLI)		NAI	NAI
Επικοινωνία με το χρήστη (frontend)	για 1 use case	για 2 use cases	για 2 use cases
Εκτελέσιμη μορφή (build & deploy your code from source)	NAI		
Testing			
Back-end functional tests	NAI	NAI	NAI
CLI unit tests		NAI	NAI
CLI functional tests			NAI
API functional tests	NAI	NAI	NAI

## Μορφότυποι Παραδοτέων

Η απαιτούμενη μορφή των παραδοτέων φαίνεται στον παρακάτω πίνακα.

ΠΑΡΑΔΟΤΕΟ	Μορφότυπος	Ονομα αρχείου παραδοτέου	Σχόλιο
Τεκμηρίωση			
Εγγραφο StRS - Stakeholder Requirements Specification	docx ή odt, σύμφωνα με template που διατίθεται	strs-softeng-XX.zip	Χρησιμοποιήστε τα πρότυπα έγγραφα 2021. Δεν γίνονται δεκτά pdf, παρά μόνο αν συνοδεύονται από αρχεία σε επεξεργάσιμη μορφή.
Εγγραφο SRS - Software Requirements Specification		srs-softeng-XX.zip	
Διαγράμματα ER ή NoSQL JSON schemas	Ενιαίο αρχείο Visual Paradigm (.vpp)	softeng21-XX.vpp	Προσοχή!: ένα αρχείο vpp για όλα τα διαγράμματα. Δεν γίνονται δεκτά διαγράμματα από σχεδιαστικά λογισμικά.
Διαγράμματα UML Activity / State			
Διαγράμματα UML Sequence			
Διαγράμματα UML Deployment			
Διαγράμματα UML Component			
Υλοποίηση			
Εισαγωγή, διαχείριση και πρόσβαση σε δεδομένα (backend)	Πηγαίος κώδικας στη γλώσσα υλοποίησης	(σύμφωνα με το περιβάλλον ανάπτυξης)	Θα πρέπει να περιλάβετε ένα αρχείο README, παρέχοντας μια συνοπτική περιγραφή των συστατικών της εφαρμογής και των βημάτων που απαιτούνται για το στήσιμο του περιβάλλοντος ανάπτυξης (IDE, βιβλιοθήκες, κλπ).
Database dump (sql ή json)	SQL ή json		
RESTful API	Πηγαίος κώδικας στη γλώσσα υλοποίησης		
API documentation - demo	Postman scripts		
Command line interface (CLI)	Πηγαίος κώδικας στη γλώσσα υλοποίησης		
Επικοινωνία με το χρήστη (frontend)			
Εκτελέσιμη μορφή (build & deploy your code from source)			
Testing			
Back-end functional tests	test scripts	(σύμφωνα με το περιβάλλον ανάπτυξης)	
CLI unit tests			
CLI functional tests			
API functional tests			

## Προθεσμίες

**Παράδοση 1:** Περιλαμβάνει τα έγγραφα τεκμηρίωσης και τα διαγράμματα UML που τα συνοδεύουν, καθώς και τη σχεδίαση της ΒΔ. Η υποβολή θα πρέπει να έχει γίνει στο σύστημα helios το αργότερο **έως τα μεσάνυχτα της 15ης Νοεμβρίου 2021.**

**Παράδοση 2:** Περιλαμβάνει το σύνολο της τεκμηρίωσης (με ενημερωμένα τα της πρώτης παράδοσης, αν απαιτείται), καθώς και την υλοποίηση του backend και του REST API. Η υποβολή των εγγράφων θα γίνει στο σύστημα helios, ενώ ο πηγαίος κώδικας των backend και REST API θα καταχωρηθεί στο Github. Η προθεσμία είναι **έως τα μεσάνυχτα της 21ης Δεκεμβρίου 2021.**

**Τελική παράδοση:** Η τελική έκδοση του συνόλου της εργασίας περιλαμβάνει το σύνολο της τεκμηρίωσης (με όποιες ενημερώσεις απαιτούνται) και το σύνολο του πηγαίου κώδικα. Η τεκμηρίωση θα υποβληθεί στο helios και επίσης θα περιέχεται στον αντίστοιχο φάκελο στο github. Η παράδοση θα πρέπει να έχει γίνει **έως τα μεσάνυχτα της ημέρας πριν από την εξέταση του μαθήματος.**

Αναλυτικά οι παραδόσεις φαίνονται στον ακόλουθο πίνακα.

ΠΑΡΑΔΟΤΕΟ	Παράδοση 1: 15.11.2021	Παράδοση 2: 21.12.2021	Τελική παράδοση: Ημ/νία εξέτασης
<b>Τεκμηρίωση</b>			
Εγγραφο StRS - Stakeholder Requirements Specification	helios	helios	helios & [your github]/doc
Εγγραφο SRS - Software Requirements Specification	helios	helios	helios & [your github]/doc
Διαγράμματα ER ή NoSQL JSON schemas	helios	helios	helios & [your github]/doc
Διαγράμματα UML Activity / State	helios	helios	helios & [your github]/doc
Διαγράμματα UML Sequence		helios	helios & [your github]/doc
Διαγράμματα UML Deployment		helios	helios & [your github]/doc
Διαγράμματα UML Component		helios	helios & [your github]/doc
<b>Υλοποίηση</b>			
Εισαγωγή, διαχείριση και πρόσβαση σε δεδομένα (backend)		[your github repo]/backend	[your github]/backend
Database dump (sql ή json)		[your github repo]/database	[your github]/database
RESTful API		[your github repo]/api	[your github]/api
API documentation - demo		[your github repo]/api	[your github]/api
Command line interface (CLI)			[your github]/cli
Επικοινωνία με το χρήστη (frontend)			[your github]/frontend
Εκτελέσιμη μορφή (build & deploy your code from source)			[your github]/{backend, api, cli}

ΠΑΡΑΔΟΤΕΟ	Παράδοση 1: 15.11.2021	Παράδοση 2: 21.12.2021	Τελική παράδοση: Ημ/νία εξέτασης
<b>Testing</b>			
Back-end functional tests			[your github]/test-backend
CLI unit tests			[your github]/test-cli
CLI functional tests			[your github]/test-cli
API functional tests			[your github]/test-api

## Βαθμολογία - βαρύτητες

Η βαθμολογία του μαθήματος προκύπτει κατά 50% από την εργασία και κατά 50% από την τελική εξέταση. Εκτός της υποβολής, η εργασία παρουσιάζεται και προφορικά από τα μέλη της ομάδας, σε ημερομηνίες μετά τη λήξη της εξεταστικής, οι οποίες θα ανακοινωθούν. Για τη λήψη προβιβάσιμου βαθμού στο μάθημα απαιτείται προβιβάσιμος βαθμός σε αμφότερες την εργασία και την τελική εξέταση.

Η βαθμολογία της εργασίας προκύπτει με τις ακόλουθες βαρύτητες:

- Τεκμηρίωση 35%
- Υλοποίηση 40%
- Testing 15%
- Χρήση εργαλείων 10%
- Γενική εικόνα 10%

**Σημείωση 1:** Κανονικοποίηση, εφόσον απαιτείται, γίνεται σε επίπεδο θέματος και όχι για το σύνολο της εργασίας ή της εξέτασης. Η βαθμολογία της εργασίας των μελών μιας ομάδας ενδέχεται να διαφοροποιηθεί ανάλογα με τα δεδομένα που θα καταγραφούν στο helios, στο Github και στο YouTrack/UpSource.

**Σημείωση 2:** Η τελική φόρμουλα βαθμολόγησης, όπως έχει ανακοινωθεί στο μάθημα, είναι η ακόλουθη:

$$\text{FinalGrade} = \text{ceiling}(\text{sumproduct}(\text{PartialGrades}; \text{weights}); 1)$$

όπου  $\text{ceiling}(7,1;1) = \text{ceiling}(7,9;1) = 8$

*Συμπληρώσεις κλπ τροποποιήσεις της εκφώνησης θα ανακοινώνονται στο helios.*