

# Εργασία στο Μάθημα της Τεχνολογίας Λογισμικού Παραδοτέα και διαδικασία εξέτασης

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών  
Εθνικό Μετσόβιο Πολυτεχνείο

Χειμερινό εξάμηνο 2019-2020

Διδάσκοντες: Ν. Παπασπύρου, Β. Βεσκούκης, Κ. Σαΐδης

## Παραδοτέα

### Δομή φακέλων στο Git repository της ομάδας

Η δομή των φακέλων στο git repository της κάθε ομάδας θα πρέπει να ακολουθεί την εξής δομή:

- Φάκελος **back-end**: περιέχει τον κώδικα της back-end εφαρμογής.
- Φάκελος **cli-client**: περιέχει τον κώδικα της cli εφαρμογής.
- Φάκελος **documentation**: περιέχει την τεκμηρίωση της εργασίας (συνολικά).
- Φάκελος **front-end**: περιέχει τον κώδικα της front-end εφαρμογής (για 4μελείς ομάδες).
- Φάκελος **mobile-app**: περιέχει τον κώδικα της mobile εφαρμογής (για 5μελείς ομάδες που θα το επιλέξουν).

Σε περίπτωση που το κρίνετε απαραίτητο, μπορείτε να προσθέσετε νέους φακέλους στην παραπάνω δομή (για παράδειγμα, λόγω δημιουργίας κάποιου component/module που (επανα)χρησιμοποιείται σε δύο ή περισσότερες εφαρμογές).

### Λίστα παραδοτέων

Η λίστα των παραδοτέων του μαθήματος φαίνεται στον παρακάτω πίνακα. Τα παραδοτέα θα πρέπει να καταχωριστούν υποχρεωτικά στον αντίστοιχο φάκελο του github repository κάθε ομάδας, με ημ/νία καταχώρισης (commit date) προγενέστερη της προθεσμίας. Δεν θα υπάρξει άλλη διαδικασία υποβολής της εργασίας.

Παραδοτέο	Φάκελος github	Ομάδα 3 ατόμων	Ομάδα 4 ατόμων	Ομάδα 5 ατόμων
Documentation - Diagrams				
Εγγραφο StRS - Stakeholders Requirements Specification	documentation		N/A	
Εγγραφο SRS - Software Requirements Specification		2 Use Cases	3 Use Cases	3 Use Cases
Διαγράμματα UML	αντίστοιχα με τα Use Cases			

<b>Παραδοτέο</b>	<b>Φάκελος github</b>	<b>Ομάδα 3 ατόμων</b>	<b>Ομάδα 4 ατόμων</b>	<b>Ομάδα 5 ατόμων</b>
Activity				
Διαγράμματα UML Sequence		αντίστοιχα με τα Use Cases		
Διαγράμματα UML Deployment			NAI	
Διαγράμματα UML Component			NAI	
Διαγράμματα UML Class			NAI	
Source code - implementation				
Πηγαίος κώδικας εφαρμογής για εισαγωγή, διαχείριση και πρόσβαση σε πρωτογενή δεδομένα (backend)	back-end		NAI	
RESTful API	back-end		NAI	
Command line interface (CLI)	cli-client		NAI	
Εκτελέσιμη μορφή (build & deploy your code from source)		back-end, cli-client	back-end, cli-client, front-end	back-end, cli-client, front-end, mobile-app
Back-end unit tests	back-end		NAI	
CLI unit tests	cli-client		NAI	
Back-end functional tests	back-end		NAI	
CLI functional tests	cli-client		NAI	
Front-end παρουσίασης δεδομένων σε περιβάλλον web	front-end		NAI	NAI
Front-end tests	front-end		ΠΡΟΑΙΡΕΤΙΚ Α	ΠΡΟΑΙΡΕΤΙΚ Α
Mobile εφαρμογή παρουσίασης δεδομένων σε smartphone	mobile-app			NAI (*)
Mobile app tests	mobile-app			ΠΡΟΑΙΡΕΤΙΚ Α

Παραδοτέο	Φάκελος github	Ομάδα 3 ατόμων	Ομάδα 4 ατόμων	Ομάδα 5 ατόμων
Λειτουργίες χρέωσης του REST API	back-end, client, front-end			NAI (*)
Ανάλυση επιδόσεων του REST API ("benchmarking")	back-end, client, front-end			NAI (*)

(\*) Ένα από τα τρία

Όλα τα σενάρια ελέγχων (unit and functional tests) θα πρέπει να εκτελούνται αυτοματοποιημένα από το build εργαλείο σας.

### Μορφότυποι Παραδοτέων

Η μορφή των παραδοτέων φαίνεται στον παρακάτω πίνακα.

Παραδοτέο	Μορφότυπος	Παρατηρήσεις
Documentation - Diagrams		
Έγγραφα StRS & SRS	docx, odt ή άλλη ανοικτή <b>επεξεργάσιμη</b> μορφή, σύμφωνα με το πρότυπο που διατέθηκε	
Διαγράμματα UML	Αρχείο Visual Paradigm (vpp)	Θα πρέπει να παραδοθεί <b>*ένα*</b> αρχείο visual paradigm (vpp) με όλα τα διαγράμματα. Θα πρέπει να υπάρχει κατάλληλη ονομασία των διαγραμμάτων ώστε να είναι εύκολα ανιχνεύσιμη η ένταξή τους στα κείμενα, όπου γίνεται.  Επίσης, στα διαγράμματα είναι επιθυμητές αναφορές (παραπομπές) στα αντίστοιχα συστατικά στοιχεία πηγαίου κώδικα, όπου αυτό έχει νόημα, καθώς και

		συνδέσεις μεταξύ τους.
Source code - implementation		
Πηγαίος κώδικας για το σύνολο των λειτουργιών που υλοποιήσατε	Το σύνολο των πηγαίων αρχείων σύμφωνα με το περιβάλλον, εργαλεία κλπ που θα χρησιμοποιήσετε.	Θα πρέπει να περιλάβετε στο αρχείο README σε κάθε επιμέρους εφαρμογής, παρέχοντας μια συνοπτική περιγραφή των συστατικών της εφαρμογής και των βημάτων που απαιτούνται για το στήσιμο του περιβάλλοντος ανάπτυξης.

# Προθεσμίες

Το πρώτο σκέλος των παραδοτέων (τα έγγραφα και διαγράμματα τεκμηρίωσης) θα υποβληθεί δύο φορές: η πρώτη έκδοση θα παραδοθεί πριν την έναρξη της εξεταστικής και η δεύτερη -και οριστική- έκδοση θα υποβληθεί μετά το πέρας αυτής.

Το δεύτερο σκέλος των παραδοτέων (πηγαίος κώδικας, υλοποίηση) θα υποβληθεί άπαξ μετά την οριστική υποβολή των παραδοτέων τεκμηρίωσης.

Για όλες τις προθεσμίες, λαμβάνεται υπόψη η ημερομηνία καταχώρησης (commit) στο github repository της ομάδας σας.

## 1<sup>η</sup> έκδοση τεκμηρίωσης

Η πρώτη έκδοση των εγγράφων και διαγραμμάτων τεκμηρίωσης (documentation - diagrams) θα πρέπει να παραδοθεί το αργότερο **έως τα μεσάνυχτα της προηγούμενης από την ημερομηνία εξέτασης του μαθήματος**. Σε ότι αφορά τα διαγράμματα, δεν απαιτείται πρόσθετη περιγραφή, παρά μόνο το κάθε διάγραμμα αυτό καθ' αυτό μέσα στο αρχείο vpp.

## 2<sup>η</sup> έκδοση τεκμηρίωσης (τελική)

Η τελική έκδοση των εγγράφων και διαγραμμάτων τεκμηρίωσης (documentation - diagrams) θα πρέπει να παραδοθεί το αργότερο **έως τα μεσάνυχτα της 6ης ημέρας μετά την ημερομηνία λήξης της εξεταστικής** (πχ: αν η εξεταστική λήγει 10 Φεβρουαρίου, η ημερομηνία παράδοσης της τελικής έκδοσης της τεκμηρίωσης είναι τα μεσάνυκτα της 16ης προς 17η Φεβρουαρίου). Μπορείτε να υποβάλλετε και τροποποιημένες εκδοχές των εγγράφων και των διαγραμμάτων, δεδομένου ότι έχετε κάνει σημαντικές τροποποιήσεις σε σχέση με την πρώτη τους έκδοση, καταγράφοντας ρητά σε ένα ξεχωριστό αρχείο **documentation/updates.md** μια συνοπτική περιγραφή τους.

## Πηγαίος κώδικας - προφορική εξέταση εργασίας

Ο πηγαίος κώδικας της εργασίας θα πρέπει να έχει παραδοθεί **έως τις 09:00 της ημέρας προφορικής εξέτασης** των ομάδων, για όλες τις ομάδες, ανεξάρτητα από την ώρα εξέτασής τους. Η διαδικασία θα λάβει χώρα **κατά την πρώτη εβδομάδα μετά τη λήξη της εξεταστικής μεταξύ 09:00 και 17:00, σε ακριβή ημερομηνία που θα ανακοινωθεί** μετά την ανακοίνωση της εξεταστικής. Η ένταξη των ομάδων στα διαθέσιμα χρονοπαράθυρα (διάρκειας 30') θα γίνει από τους διδάσκοντες με τυχαίο τρόπο και δεν θα είναι δυνατό να αλλάξει.

## Χρόνος, διάρκεια και διαδικασία εξέτασης

Κάθε ομαδική εξέταση θα διαρκεί 30' (αυστηρά). Η παρουσία όλων των μελών της ομάδας είναι υποχρεωτική στην εξέταση (σπουδαστές που δεν θα εμφανιστούν στην εξέταση δε θα λάβουν βαθμό στην εργασία).

Η διαδικασία θα είναι η εξής:

- Κάθε ομάδα προσέρχεται με το δικό της laptop, στο οποίο έχουν προ-εγκατασταθεί και ήδη τεθεί σε λειτουργία όλα τα απαραίτητα για τη λειτουργία συστατικά της εφαρμογής (τα οποία χρησιμοποιούνται ως έχουν, δηλαδή δεν έχει γραφεί ο κώδικας τους στο πλαίσιο της εργασίας), όπως είναι, για παράδειγμα, η βάση δεδομένων.
- Στα πρώτα 5':
  - ο Γίνεται checkout του κώδικα από το git repository της ομάδας.
  - ο Γίνεται εκτέλεση του build του κώδικα καθώς και των functional tests αυτού, τα οποία θα πρέπει να επιτυγχάνουν χωρίς σφάλματα.
  - ο Παράγονται τα application artifacts, γίνεται το deployment αυτών και εκκίνηση της εφαρμογής.
- Στα επόμενα 20':
  - ο Γίνεται παρουσίαση (demonstration) των λειτουργιών της εφαρμογής.
- Στα τελευταία 5':
  - ο Γίνεται εκτέλεση των αυτοματοποιημένων ελέγχων στο REST API της εφαρμογής και της batch import λειτουργίας της.

Στις αρχές Φεβρουαρίου θα διατεθεί και μια κοινή σουίτα ελέγχων (σενάρια ελέγχου σε κοινό data set) για να σας βοηθήσει να δοκιμάσετε τα σενάρια ελέγχου που θα εκτελεστούν κατά την εξέταση.