

ΑΝΑΛΥΣΗ & ΣΧΕΔΙΑΣΗ ΣΥΣΤΗΜΑΤΩΝ ΛΟΓΙΣΜΙΚΟΥ

Υποχρεωτική Εργασία

Εαρινό Εξάμηνο 2025

1. Περιγραφή

Η παρούσα εργασία περιγράφει μία υποθετική ψηφιακή πλατφόρμα (εφαρμογή) τεχνολογίας λογισμικού. Μελετήστε προσεκτικά τις περιγραφές που ακολουθούν και υλοποιήστε τα ζητούμενα αξιοποιώντας πλήρως τη θεωρία του μαθήματος.

Η εργασία θα πρέπει να υλοποιηθεί αποκλειστικά από **ομάδες δύο ατόμων**.

1.1. Γενικές Απαιτήσεις της Εφαρμογής

1. Στην εφαρμογή μπορούν να έχουν πρόσβαση οι χρήστες μέσω τεσσάρων βασικών ρόλων: επόπτης, developers, αναλυτές-σχεδιαστές, stakeholders.
 1. Κάθε χρήστης έχει έναν μοναδικό κωδικό (ID).
 2. Ο επόπτης είναι αποκλειστικά ένας χρήστης.
 3. Όλοι οι χρήστες συνδέονται με username/password και ακολουθεί 2-factor-authentication με μήνυμα (sms) στο κινητό τους τηλέφωνο. Επιπροσθέτως, για τον επόπτη εφαρμόζεται και αυθεντικοποίηση μέσω βιομετρικών στοιχείων.
 4. Όλοι οι χρήστες μπορούν να δημιουργήσουν αναφορές. Ωστόσο, η αναφορά του επόπτη έχει τη μορφή κειμένου, η αναφορά του developer έχει τη μορφή εκτελέσιμου αρχείου, η αναφορά του αναλυτή-σχεδιαστή έχει τη μορφή αρχείου StarUML, ενώ η αναφορά του stateholder έχει τη μορφή ηχητικού αρχείου.
2. Ο επόπτης μπορεί να:
 1. δημιουργήσει ή να καταργήσει κάποιον χρήστη στην εφαρμογή. Σε αυτήν την περίπτωση, στέλνεται ενημερωτικό μήνυμα (sms) στον χρήστη του λογαριασμού,
 2. έχει πρόσβαση (read only) στα αρχεία των υπόλοιπων χρηστών (developers, αναλυτών-σχεδιαστών, stakeholders),
 3. στείλει ή να αποδεχτεί κάποιο απλό γραπτό μήνυμα (χωρίς επισυναπτόμενα) από τους υπόλοιπους χρήστες (chat),

4. δέχεται και εγκρίνει ή απορρίπτει κάποιο αίτημα των υπόλοιπων χρηστών. Σε αυτήν την περίπτωση, στέλνεται ενημερωτικό μήνυμα (sms) στον χρήστη του λογαριασμού.
 5. Επίσης, ανά πάσα στιγμή γνωρίζει την αναλογία developers ανά αναλυτή-σχεδιαστή του συστήματος, καθώς και την αναλογία αναλυτές-σχεδιαστές ανά stakeholder του συστήματος.
3. Οι developers μπορούν να:
1. αναρτήσουν/επεξεργαστούν/διαγράψουν αρχεία με τον κώδικα που έγραψαν,
 2. έχουν πρόσβαση (read only) στα αρχεία των άλλων developers και των αναλυτών-σχεδιαστών,
 3. στείλουν ή να αποδεχτούν κάποιο απλό γραπτό μήνυμα (χωρίς επισυναπτόμενα) από τους υπόλοιπους χρήστες (chat),
 4. στείλουν κάποιο αίτημα στον επόπτη.
4. Οι αναλυτές-σχεδιαστές μπορούν να:
1. αναρτήσουν/επεξεργαστούν/διαγράψουν αρχεία με τις απαιτήσεις, τις προδιαγραφές και τη σχεδίαση του εκάστοτε συστήματος,
 2. έχουν πρόσβαση (read only) στα αρχεία των άλλων αναλυτών-σχεδιαστών,
 3. στείλουν ή να αποδεχτούν κάποιο απλό γραπτό μήνυμα (χωρίς επισυναπτόμενα) από τους υπόλοιπους χρήστες (chat),
 4. στείλουν κάποιο αίτημα στον επόπτη.
5. Οι stakeholders μπορούν να:
1. στείλουν ή να αποδεχτούν κάποιο απλό γραπτό μήνυμα (χωρίς επισυναπτόμενα) από τους υπόλοιπους χρήστες (chat),
 2. έχουν πρόσβαση σε συγκεκριμένα έγγραφα (κείμενα, διαγράμματα) των αναλυτών-σχεδιαστών προκειμένου να τα εγκρίνουν, να τα απορρίψουν ή να υποβάλουν σχόλια.
6. Στο ψηφιακό σύστημα έχει πρόσβαση και ένας ακόμα τύπος χρήστη ως helpdesk: ένα εξωτερικό σύστημα τεχνητής νοημοσύνης (AI).
1. Το σύστημα τεχνητής νοημοσύνης προσφέρει υπηρεσίες helpdesk στους υπόλοιπους χρήστες του συστήματος.
7. Το ψηφιακό σύστημα διαθέτει ένα υποσύστημα προκειμένου να συντονίζει την επικοινωνία των χρηστών μεταξύ τους (chat). Σε αυτό υπάγονται όλοι οι χρήστες. Ακόμα και αν πάψει να υφίσταται το chat, οι βασικοί χρήστες εξακολουθούν να υπάρχουν στο σύστημα.

2. Ζητούμενα

2.1. UML

1. Δημιουργήστε το διάγραμμα περιπτώσεων χρήσης (Use Case Diagram, UML) του συστήματος, εφαρμόζοντας όπου ενδεχομένως χρειάζεται σχέσεις γενίκευσης, επέκτασης ή/και συμπερίληψης.
2. Δημιουργήστε ένα πλήρες διάγραμμα κλάσεων του συστήματος (Class Diagram, UML).
 1. Σε αυτό θα πρέπει να αποτυπώνονται, μεταξύ άλλων, οι σχέσεις μεταξύ των κλάσεων, οι ορατότητες (private, protected, public), οι τύποι των χαρακτηριστικών και των μεθόδων (π.χ. Integer, Boolean κ.λπ.) κ.ά.
 2. Κατά τον σχεδιασμό θα πρέπει να ληφθούν υπόψη οι αρχές της αντικειμενοστραφούς σχεδίασης με βάση τη θεωρία και να αποτυπωθούν αντίστοιχα στο διάγραμμα.
3. Δημιουργήστε από ένα πλήρες διάγραμμα ακολουθίας (Sequence Diagram, UML) συμπεριλαμβάνοντας όλες τις εμπλεκόμενες κλάσεις/αντικείμενα και τις μεθόδους που σχετίζονται με κάθε ένα από τα παρακάτω δύο (2) σενάρια. Τα συστατικά μέρη των διαγραμμάτων ακολουθίας θα πρέπει να είναι σε πλήρη αντιστοιχία με το διάγραμμα κλάσεων.
 1. Σενάριο νο1 -> ο επόπτης συνδέεται στον λογαριασμό του. Στη συνέχεια:
 1. επικοινωνεί μέσω chat με τον stateholder και χωρίς να περιμένει να του απαντήσει, επικοινωνεί μέσω chat και με τον αναλυτή-σχεδιαστή. Ο αναλυτής-σχεδιαστής επικοινωνεί με τον stateholder και εάν η διαδικασία ολοκληρωθεί σωστά, επικοινωνεί στη συνέχεια με τον επόπτη. Έπειτα:
 2. ο επόπτης επικοινωνεί δημιουργεί έναν νέο χρήστη developer και επικαιροποιεί την αναλογία developers ανά αναλυτή-σχεδιαστή. Ο developer συνδέεται στον λογαριασμό του και στη συνέχεια επικοινωνεί μέσω του chat με τον αρμόδιο αναλυτή/σχεδιαστή για πληροφορίες. Στη συνέχεια, ο developer στέλνει αίτημα στον επόπτη προκειμένου να του κοινοποιηθούν τα διαγράμματα του αναλυτή-σχεδιαστή: εάν ο επόπτης εγκρίνει το αίτημα, ο επόπτης επικοινωνεί στο chat με τον αντίστοιχο αναλυτή-σχεδιαστή). Όταν ο developer ολοκληρώσει το έργο του (συγγραφή κώδικα), ενημερώνει μέσω chat τον επόπτη, ο οποίος τον διαγράφει από το σύστημα.
 2. Σενάριο νο2 -> για όσο είναι σε λειτουργία το σύστημα:
 1. Παράλληλα και εφόσον κάποιος ή κάποιοι χρήστες το θελήσουν (developers, αναλυτές-σχεδιαστές, stakeholders) μπορούν να στείλουν στον επόπτη κάποιο αίτημα προς έγκριση. Το συγκεκριμένο αίτημα, δεν χρειάζεται να ολοκληρωθεί προκειμένου να σταλθεί νέο αίτημα στον χρήστη helpdesk/AI (βλ. παρακάτω).

2. Παράλληλα και εφόσον κάποιος ή κάποιοι χρήστες το θελήσουν (επόπτης, developers, αναλυτές-σχεδιαστές, stakeholders), μπορούν να στείλουν κάποιο αίτημα στον χρήστη helpdesk (AI).
4. Δημιουργήστε ένα πλήρες διάγραμμα μηχανής καταστάσεων (Statechart Diagram, UML) εστιάζοντας σε όλες τις πιθανές καταστάσεις που μπορεί να λάβει το σύστημα. Αξιοποιήστε όπου χρειάζεται τον κατάλληλο συνδυασμό των τριών μερών (βλ. ετικέτες): **σκανδάλη [φρουρός] / δραστηριότητα**. Κάθε ένα από τα εν λόγω τρία μέρη θα πρέπει να εμφανίζεται στο διάγραμμα τουλάχιστον μια φορά.
5. Δημιουργήστε δύο διαγράμματα δραστηριοτήτων (Activity Diagram, UML) με βάση τις ενέργειες που περιγράφονται αντίστοιχα στα σενάρια/υποκεφάλαια 2.1.3.1 και 2.1.3.2 (βλ. Διαγράμματα Ακολουθίας). Ενσωματώστε σε αυτά όποιες άλλες σχετικές πληροφορίες υπάρχουν στις περιγραφές των απαιτήσεων του συστήματος.

2.2. Δομημένη Ανάλυση

- ❖ Δημιουργήστε ένα ΔΡΔ (Επίπεδο Αφαίρεσης: 1) για τη συνολική λειτουργία του συστήματος.
- ❖ Επιλέξτε τη διαδικασία του προηγούμενου ΔΡΔ, εκείνη με τις περισσότερες ροές δεδομένων, και δημιουργήστε ένα νέο ΔΡΔ (Επίπεδο Αφαίρεσης: 2).
- ❖ Δημιουργήστε έναν πίνακα αποφάσεων με βάση τις εξής περιπτώσεις:
 - Εάν ο επόπτης, οι developers, οι αναλυτές-σχεδιαστές και οι stakeholders συμφωνούν ως προς την ολοκλήρωση του λογισμικού, το λογισμικό παραδίδεται στον πελάτη με πλήρη χρέωση.
 - Εάν ο επόπτης και οι stakeholders συμφωνούν ως προς την ολοκλήρωση του λογισμικού, αλλά δεν συμφωνούν οι αναλυτές-σχεδιαστές, το λογισμικό παραδίδεται στον πελάτη με μειωμένη χρέωση.
 - Εάν ο επόπτης ή οι stakeholders δεν συμφωνούν, το σύστημα δεν παραδίδεται στον πελάτη.

3. Παραδοτέα και Διαδικασία Υποβολής

Η παράδοση της εργασίας γίνεται αποκλειστικά μέσω του e-class. Πιο συγκεκριμένα, θα αναρτήσετε στην περιοχή Εργασίες του μαθήματος ένα αρχείο (pdf) με όνομα **αριθμόςΜητρώου1_αριθμόςΜητρώου2** που θα περιλαμβάνει:

- ☐ Εξώφυλλο με τα ονόματα της ομάδας εργασίας, τους αριθμούς μητρώου και τα e-mail.
- ☐ Πίνακα Περιεχομένων (με τα κεφάλαια και τον αριθμό σελίδας που αρχίζει το καθένα).
- ☐ Εισαγωγή (έως 200 λέξεις) στην οποία θα περιγράψετε:

- ☐ το τμήμα της εργασίας που ανέλαβε το κάθε μέλος της ομάδας,
- ☐ με ποιο τρόπο συνεργαστήκατε (π.χ. με εβδομαδιαίες συναντήσεις, εξ αποστάσεως, δια ζώσης κ.λπ.),
- ☐ ποια διαγράμματα της εργασίας σας δυσκόλεψαν περισσότερο.
- ☐ Ένα κεφάλαιο για κάθε ζητούμενο. Το κάθε κεφάλαιο θα περιέχει τα διαγράμματα (ενσωματωμένα στο κείμενο), καθώς και τυχόν σχόλια και παραδοχές που έχετε κάνει (αν δεν σας κάλυψαν οι υφιστάμενες οδηγίες/περιγραφές των απαιτήσεων).

4. Λοιπές Πληροφορίες

1. Η τελική προθεσμία υποβολής της εργασίας είναι την **Πέμπτη, 29 Μαΐου 2025** στις **11:59 μ.μ.**
2. Ένα άτομο εκ μέρους της ομάδας υποβάλλει την εργασία στο eclass (**Υποχρεωτική Εργασία (εαρινό εξάμηνο 2025)**).
3. Τα διαγράμματα της εργασίας θα πρέπει να γίνουν **αποκλειστικά** με το λογισμικό **StarUML**, διαφορετικά δεν θα λαμβάνονται υπόψη.
4. Η εργασία είναι υποχρεωτική. Για να περάσετε το μάθημα χρειάζεται να έχετε πάρει τουλάχιστον 5/10 στην εργασία και τουλάχιστον 5/10 στην γραπτή εξέταση.
5. Η εργασία θα πρέπει να γίνει από **2 άτομα**, ισχύει μόνο για το τρέχον ακαδημαϊκό έτος και συνεισφέρει κατά 30% στην τελική βαθμολογία του μαθήματος.
6. Οι εργασίες θα περάσουν από έλεγχο για αντιγραφή. Σε περίπτωση που διαπιστωθεί αντιγραφή μεταξύ δύο ή περισσότερων ομάδων, όλες οι εμπλεκόμενες εργασίες θα μηδενιστούν.
7. Οι φοιτητές/-τριες που έχουν μεγάλη διαφορά στους βαθμούς της εργασίας και του γραπτού, είναι πιθανό κληθούν για προφορική εξέταση.
8. Αν έχετε απορίες, μπορείτε να τις υποβάλετε στο e-class, στο κατάλληλο νήμα της περιοχής συζητήσεων (**Ερωτήσεις σχετικά με την υποχρεωτική εργασία 2025**).
9. Αν δεν έχετε βρει ακόμα συνεργάτη για την υποχρεωτική εργασία, μπορείτε να αναρτήσετε στο e-class, στο κατάλληλο νήμα της περιοχής συζητήσεων, την προσφορά συνεργασίας σας (**Διαθεσιμότητες φοιτητών για συγκρότηση ομάδων (υποχρεωτική εργασία)**).

Καλή επιτυχία!