**Μηχανική Λογισμικού για Διαδικτυακές Εφαρμογές**

**Καζλάρης Ιωάννης, Παναγιωτίδης Ιωάννης, Εμμανουηλίδης Παναγιώτης, Τασιός Χρυσόστομος**

Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων

Διεθνές Πανεπιστήμιο της Ελλάδος

**Σίνδος Δεκέμβριος 2022**

ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή

Στα πλαίσια του μαθήματος “Μηχανική Λογισμικού για Διαδικτυακές Εφαρμογές” του μεταπτυχιακού προγράμματος σπουδών «Ευφυείς τεχνολογίες διαδικτύου» του Διεθνούς Πανεπιστημίου Θεσσαλονίκης συστάθηκε η ομάδα «**WebEng22\_Group2**» η οποία αποτελείται από τους ακόλουθους φοιτητές, σε αλφαβητική σειρά:

* ΕΜΜΑΝΟΥΗΛΙΔΗΣ ΠΑΝΑΓΙΩΤΗΣ
* ΚΑΖΛΑΡΗΣ ΙΩΑΝΝΗΣ
* ΠΑΝΑΓΙΩΤΙΔΗΣ ΙΩΑΝΝΗΣ
* ΤΑΣΙΟΣ ΧΡΥΣΟΣΤΟΜΟΣ

Σκοπός της παρούσας εργασίας είναι η ανάπτυξη ενός web application που να διαχειρίζεται δεδομένα που αφορούν στις βάσεις εισαγωγής των Πανεπιστημίων και τα αντίστοιχα στατιστικά στοιχεία τους

Στόχος είναι η ανάπτυξη ενός **data/user driven web application** το οποίο ανακτά δεδομένα από το WEB API και τα χρησιμοποιεί λαμβάνοντας υπ’ όψιν τις ελάχιστες μη λειτουργικές απαιτήσεις οι οποίες είναι:

1. χρήση μιας βάσης δεδομένων όπου η εφαρμογή σας θα αποθηκεύει και θα ανακτά δεδομένα,
2. η εφαρμογή να είναι user/data driven, δηλαδή ο χρήστης(ες) ταυτοποιείται και θα κάνει κάποιες ενέργειες με τα δεδομένα του συστήματος και
3. θα πρέπει να υπάρχει ένα βασικό UX Design/Testing.

Η εφαρμογή αναπτύχθηκε σε συνεργασία με όλα τα μέλη της ομάδας μέσω του συστήματος source/versioning control Git και του Github στο ακόλουθο url, όπου και υπάρχουν όλα τα απαιτούμενα στοιχεία της υλοποιηθείσας εφαρμογής: <https://github.com/iee-ihu-gr-coursem102/WebEng22_Group2>

1. **Μεθοδολογία Scrum**

Για την εκπόνηση του project η ομάδα μας ακολούθησε την μεθοδολογία Scrum, η οποία είναι μια σύγχρονη ευέλικτη μέθοδος διαχείρισης έργων.

Όπως γνωρίζουμε η μεθοδολογία της Scrum χρησιμοποιείται σε πλήθος έργων ως επαναληπτική και αυξητική μέθοδος ανάπτυξης έργων, χρησιμοποιείται σαν μεθοδολογία ανάπτυξης λογισμικού, όσον αφορά την οργάνωση και λειτουργία των ομάδων ανάπτυξης αλλά και στον σχεδιασμό και διαχείριση των ευέλικτων (agility) έργων.

Με τη χρήση της Scum επιτυγχάνεται η διαδικασία ανάπτυξης ενός προϊόντος, σε όλες τις επί μέρους φάσεις, που συνήθως επικαλύπτονται και εκτελείται από μία λειτουργική «ομάδα». Προσδίδοντας ευελιξία στην οργάνωση και διαχείριση της διαδικασίας ανάπτυξης λογισμικού, αυξάνεται η παραγωγικότητα της ομάδας αλλά και η ικανότητα προσαρμογής της σύμφωνα με τις εκάστοτε ανάγκες που προκύπτουν. Ακόμα η μέθοδος αυτή επιδιώκει όσο το δυνατόν μικρότερο χρονικό διάστημα κύκλου ανάπτυξης και παράδοση τμημάτων κώδικα του συστήματος που είχαν συμφωνηθεί ανά κύκλο. Με αυτό τον τρόπο παραδίδεται στον πελάτη κάποιο μέρος ανά κύκλο το οποίο να είναι μερικώς εκτελέσιμο και να μπορεί να δουλέψει αυτό το μέρος του έργου που παραδόθηκε, χωρίς να περιμένει την πλήρη υλοποίηση του που ίσως να καθυστερήσει.

Η μέθοδος του Scrum διαχωρίζεται σε 3 σημαντικές φάσεις (Schwaber and Beedle, 2008):

1. **Αρχική Διερεύνηση (pre game)**
2. **Σχεδιασμός (game)**
3. **Ολοκλήρωση (post game)**

Στη **Φάση Αρχικής Διεύρυνσης** διακρίνουμε δύο υποκατηγορίες

* 1. **Φάση Ανάλυσης**, όπου γίνεται ο καθορισμός των προδιαγραφών, οι απαιτήσεις, η δημιουργία του καταλόγου χαρακτηριστικών (product backlog list), προτεραιότητα απαιτήσεων, και η πιστοποίηση του συστήματος που πρόκειται να αναπτυχθεί. σύστημα.
  2. Φάση υψηλού επιπέδου σχεδιασμού, όπου υλοποιούνται η αρχιτεκτονική και ο σχεδιασμός του συστήματος με βάση τις αρχικές προδιαγραφές του συστήματος οι οποίες έχουν ήδη καθοριστεί στη προηγούμενη φάση.

Στη **Φάση Ανάπτυξης** σε κανονικές συνθήκες το σύστημα αναπτύσσεται σε πολλές επαναλήψεις (Sprints). Οι επαναλήψεις είναι επαναληπτικοί κύκλοι ανάπτυξης όπου ο κάθε κύκλος συμπεριλαμβάνει όλες τις φάσεις ανάπτυξης. Σε κάθε επανάληψη υλοποιούνται συγκεκριμένες απαιτήσεις, που ονομάζονται απαιτήσεις της επανάληψης (sprint backlog), και έχουν επιλεχθεί από τον κατάλογο των συνολικών απαιτήσεων του προϊόντος, με την βοήθεια του πελάτη. Οι απαιτήσεις αυτές πρέπει θα υλοποιηθούν μέσα στο χρονικό διάστημα της τρέχουσας επανάληψης. Η κάθε επανάληψη διαρκεί ένα συγκεκριμένο χρονικό διάστημα που κυμαίνεται από μία εβδομάδα έως ένα μήνα. Στο τέλος κάθε επανάληψης η ομάδα πρέπει να έχει "ένα εκτελέσιμο παραδοτέο προϊόν". Ένα έργο ανάπτυξης λογισμικού συνήθως υλοποιείται και παραδίδεται στον πελάτη σε τρεις έως οκτώ επαναλήψεις. Στη δική μας περίπτωση, θα υλοποιηθεί το σύστημα σε δύο φάσεις λόγω των χρονικών περιορισμών.

Τέλος, στη **φάση της ολοκλήρωσης** εκτελούνται όλες εκείνες οι διαδικασίες που απαιτούνται για τη περάτωση και την τελική παράδοσή του έργου στον πελάτη. Η ομάδα ανάπτυξης αφού επιβεβαιώσει, πάντα σε συνεργασία με τον πελάτη, ότι όλες οι απαιτήσεις έχουν υλοποιηθεί, κινεί τις διαδικασίες παράδοσης του συστήματος.

1. Οργάνωση ομάδας και αναθέσεις αρμοδιοτήτων

Λόγω του περιορισμού των ατόμων, η οργάνωση της ομάδας και η ανάθεση ρόλων σύμφωνα με τη μεθοδολογία Scrum επιλέχθηκαν και συμπτύχθηκαν ορισμένοι ρόλοι προκειμένου να γίνει η εφαρμογή της μεθοδολογίας.

Πιο συγκεκριμένα οι ρόλοι που υπήρξαν ήταν οι εξής:

* product/project manager
* front end developer
* back-end developer
* UX tester

Αφού έγινε συνάντηση των μελών και υπήρξε ανάπτυξη των δεξιοτήτων του κάθε μέλους της ομάδας, ομόφωνα καταλήξαμε στην ακόλουθη ανάθεση των ρόλων:

|  |  |  |  |
| --- | --- | --- | --- |
| product/project manager | front end developer | back-end developer | UX tester |
| ΚΑΖΛΑΡΗΣ  ΙΩΑΝΝΗΣ | ΠΑΝΑΓΙΩΤΙΔΗΣ ΙΩΑΝΝΗΣ | ΤΑΣΙΟΣ ΧΡΥΣΟΣΤΟΜΟΣ | ΕΜΜΑΝΟΥΗΛΙΔΗΣ ΠΑΝΑΓΙΩΤΗΣ |

1. Οι λειτουργικές απαιτήσεις της εφαρμογής

Κατόπιν σύσκεψης και διεργασιών από την ομάδα τόσο αρχικά όσο και σε καθημερινή βάση, η ομάδα κατάληξε στις παρακάτω λειτουργικές απαιτήσεις της εφαρμογής που σκοπεύουμε να υλοποιήσουμε, με προτεραιότητα που καθορίζεται αμέσως μετά με τον χρωματισμό, ενώ στην διπλανή στήλη θα εμφανίζονται τα αρχικά του κάθε μέλους που προέβη στην υλοποίηση της κάθε απαίτησης:

* ΚΑΖΛΑΡΗΣ ΙΩΑΝΝΗΣ → **ΚΙ**
* ΠΑΝΑΓΙΩΤΙΔΗΣ ΙΩΑΝΝΗΣ → **ΠΙ**
* ΤΑΣΙΟΣ ΧΡΥΣΟΣΤΟΜΟΣ → **ΤΧ**
* ΕΜΜΑΝΟΥΗΛΙΔΗΣ ΠΑΝΑΓΙΩΤΗΣ → **ΕΠ**

|  |
| --- |
| Μικρή |
| Μεσαία |
| Μεγάλη |

|  |  |
| --- | --- |
| **Απαιτήσεις** | **Υπεύθυνος-οι υλοποίησης απαίτησης** |
| Καταγραφή Απαιτήσεων (Software Requirements) |  |
| Κατασκευή Βάσης Δεδομένων (Building Database) |  |
| Κατασκευή Πινάκων (Building Tables) |  |
| Κατασκευή Όψεων (Building Views) |  |
| Καταγραφή Απαιτήσεων Πληροφορίας της Πλατφόρμας |  |
| Κατασκευή Φορμών |  |
| Δημιουργία Feature αναζήτησης Χρήστη |  |
| (Forms Visibility & Look’n’Feel) |  |
| Δημιουργία υποδομών αποθήκευσης/εισαγωγής/εξαγωγής πληροφορίας |  |
| (Forms Data Validation) |  |
| (DB Data population) |  |
| (Requirements Accomplished) |  |
| Έλεγχος (Beta Testing) |  |
| Εγχειρίδια (Documentation) |  |
| Δημιουργία Εγχειρίδιου Χρήσης (Create User Manual) |  |
| (Deployment) |  |
| Παράδοση (Delivery) |  |

1. Διάγραμμα Περιπτώσεων Χρήσης της Εφαρμογής
2. Mockups

**ΕΚΦΩΝΗΣΗ**

Η μεθοδολογία ανάπτυξης της εφαρμογής που θα ακολουθηθεί από κάθε ομάδα είναι η Scrum. Συνεπώς, ένα μέλος της ομάδας θα πρέπει να έχει τον ρόλο του product/project manager ενώ τα μέλη της ομάδας θα πρέπει να έχουν ρόλους όπως front end developer, back-end developer, UX tester.

Μέχρι **6 Δεκεμβρίου 2022** θα πρέπει να γίνει η ενδιάμεση παράδοση η οποία απαιτεί να ανέβει στο GitHub repository της εργασίας της κάθε ομάδας,

* οι ρόλοι των μελών της ομάδας Scrum, 🗹
* οι λειτουργικές απαιτήσεις και η προτεραιότητα τους,
* η συνεισφορά των μελών στην υλοποίηση της κάθε λειτουργικής απαίτησης,
* το διάγραμμα περιπτώσεων χρήσης της εφαρμογής καθώς και
* η αρχική σχεδίαση των οθονών της εφαρμογής (Mockups).

Συνεπώς, η εφαρμογή που θα αναπτυχθεί θα πρέπει να ανακτά δεδομένα από το WEB API και να τα χρησιμοποιεί λαμβάνοντας υπ’ όψιν τις ελάχιστες μη λειτουργικές απαιτήσεις οι οποίες είναι:

* η χρήση μιας βάσης δεδομένων όπου η εφαρμογή σας θα αποθηκεύει και θα ανακτά δεδομένα,
* η εφαρμογή να είναι user/data driven, δηλαδή ο χρήστης(ες) ταυτοποιείται και θα κάνει κάποιες ενέργειες με τα δεδομένα του συστήματος και
* (γ) θα πρέπει να υπάρχει ένα βασικό UX Design/Testing.

Το ποιές θα είναι οι λειτουργικές απαιτήσεις της εφαρμογής που θα αναπτύξει η κάθε ομάδα είναι ένα από τα βασικά κριτήρια αξιολόγησης.

Υποχρεωτικά η εφαρμογή θα αναπτυχθεί συνεργατικά μέσω του συστήματος source/versioning control Git και του Github.

Χρόνος παράδοσης των εργασιών ορίζεται η **Κυριακή 15 Ιανουαρίου 2023**. Μετά από την συγκεκριμένη ημερομηνία δεν μπορούν να πραγματοποιηθούν αλλαγές στα github repositories της εργασίας σας.

**20 Ιανουαρίου 2023** θα πραγματοποιηθούν οι παρουσιάσεις των εργασιών και οι προφορικές εξετάσεις. Η προφορική εξέταση, αν και θα γίνει με όλη την ομάδα παρούσα ταυτόχρονα, είναι ατομική και δεν αφορά την ομάδα συνολικά. Συνεπώς, η αξιολόγηση του κάθε μέλους της ομάδας εξαρτάται από το τι ΑΚΡΙΒΩΣ έπρεπε να υλοποιήσει ο καθένας και τι τελικά υλοποίησε.

Κριτήρια αξιολόγησης της εργασίας είναι

* Η επιλογή και η υλοποίηση των λειτουργικών απαιτήσεων
* User Experience (Design and Testing)
* Η ποιότητα κώδικα όπως για παράδειγμα σε τι βαθμό είναι διαχειρίσιμος, κατανοητός και εύκολα επαναχρησιμοποιήσιμος
* Υλοποίηση web API

Η εφαρμογή θα πρέπει να φιλοξενηθεί στον εξυπηρετητή ιστού του τμήματος users.iee.ihu.gr στον οποίο έχουν πρόσβαση όλοι οι φοιτητές και φοιτήτριες τους τμήματος και υποστηρίζει τις γλώσσες και τις τεχνολογίες που διδάσκονται στο μάθημα. Σε περίπτωση κάποια ομάδα επιθυμεί να χρησιμοποιήσει γλώσσες και τεχνολογίες που δεν υποστηρίζονται από τον εξυπηρετητή ιστού users, θα πρέπει να επικοινωνήσει με τους διδάσκοντες ώστε να βρεθεί άλλος εξυπηρετητής.

Το παραδοτέο της εργασίας είναι το GitHub repository. Στο GitHub repository, εκτός από τον κώδικα της εφαρμογής, θα πρέπει να είναι διαθέσιμα και τα υπόλοιπα τεκμήρια της δουλειάς. Αν κάτι δεν βρίσκεται στο GitHub, είναι σαν να μην υπάρχει και δεν αξιολογείται. Επίσης, στο GitHub repository θα πρέπει να εμφανίζεται το url όπου φιλοξενείται η εφαρμογή που αναπτύχθηκε.

Υπενθυμίζεται ότι η εργασία αφορά στο 50% της βαθμολογίας του μαθήματος. Αν κάποιος ή κάποια δεν συμμετέχει σε ομάδα και δεν υλοποιήσει εργασία τότε ο βαθμός της εργασίας λογίζεται το μηδέν και έτσι χάνει το 50% της βαθμολογίας.

1. Βιβλιογραφικές αναφορές
2. Ο Οδηγός του Scrum. Ken Schwaber - Jeff Sutherland, 2020, <https://scrumguides.org/>
3. Μαυρίδης Ιωάννης. Ασφάλεια Πληροφοριών στο Διαδίκτυο. Ελληνικά ακαδημαϊκά ηλεκτρονικά συγγράμματα και βοηθήματα. Εκδόσεις Κάλλιπος.
4. Arvinder Kaur, Ruchikaa Nayyar. A Comparative Study of Static Code Analysis Tools for Vulnerability Detection in C/C++ and Java Source Code. Third International Conference on Computing and Network Communications (CoCoNet 2019). Elsevier.
5. Valentina Lenarduzzi, Savanna Lujan, Nyyti Saarimaki, Fabio Palomba. A Critical Comparison on Six Static Analysis Tools: Detection, Agreement, and Precision.
6. Sebastian Stiernborg. Automated Code Inspection: Investigating Deployment of Continuous Inspection. Master Thesis, KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science, 2018.
7. <https://www.perforce.com/blog/kw/software-safety-vs-security-whats-different> (Προσπελάστηκε στις 26 Νοεμβρίου 2022)
8. <https://www.perforce.com/blog/kw/sast-vs-dast>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://www.veracode.com/security/code-security>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://www.veracode.com/security/secure-development>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://www.veracode.com/security/static-code-analysis>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://www.synopsys.com/glossary/what-is-code-review.html>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://www.tftus.com/blog/software-safety-vs-security-whats-different>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://owasp.org/www-community/Source_Code_Analysis_Tools>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://techbeacon.com/security/4-ways-secure-your-code-regardless-programming-language>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://www.techtarget.com/searchsecurity/tip/How-to-conduct-a-secure-code-review>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://www.misra.org.uk/>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://www.owasp.org>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://owasp.org/www-project-top-ten/>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://wiki.owasp.org/index.php/Security_by_Design_Principles>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://www.ibm.com/reports/data-breach> IBM Security: Cost of a Data Breach Report 2022

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://cheatsheetseries.owasp.org/cheatsheets/Attack_Surface_Analysis_Cheat_Sheet.html>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://cheatsheetseries.owasp.org/cheatsheets/Server_Side_Request_Forgery_Prevention_Cheat_Sheet.html>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. <https://owasp.org/www-community/Source_Code_Analysis_Tools>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. Κανονισμός προστασίας προσωπικών δεδομένων (GDPR) <https://gdpr-info.eu/>
2. <https://docs.gitlab.com/ee/ci/testing/code_quality.html>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. Fiorella Zampetti, Simone Scalabrino, Rocco Oliveto, Gerardo Canfora, Massimiliano di Penta. How Open Source Projects use Static Code Analysis Tools in Continuous Integration Pipelines.
2. Gitlab documentation. <https://docs.gitlab.com>

(Προσπελάστηκε στις 26 Νοεμβρίου 2022)

1. Kiran Kumar, K. Subrahmanya, R. Kavana, Shwetha Baliga. An Approach to basic GUI-enabled CI/CD pipeline with Static Analysis Tools. Journal of University of Shanghai for Science and Technology, ISSN: 1007-6735.