

Interreg - IPA CBC Greece - Albania 4PLUS



ΠΡΑΞΗ

Public awareness, Preparedness, Participation and Coordination for
Civil Protection for All (4PLUS)

(Subsidy Contract A2-1.4-1)

ΠΑΡΑΔΟΤΕΟ ΣΥΜΒΑΣΗΣ ΑΝΑΘΕΣΗΣ ΥΠΗΡΕΣΙΩΝ

Π8.α Έκθεση σχεδιασμού και πρωταρχική διαδραστική έκδοση της Πλατφόρμας



Σύμβαση	Μεταξύ της ΕΣΑμεΑ και της ένωσης φορέων «EUROPRACTIS M.I.K.E. - EUROPEAN PROFILES A.E.» (με ημερομηνία υπογραφής την 03.08.2020, για το Έργο «Παροχή υπηρεσιών υλοποίησης παραδοτέων της Πράξης 4PLUS», που εντάσσεται στην Πράξη "Public awareness, Preparedness, Participation and Coordination for Civil Protection for All", με ακρωνύμιο "4PLUS" (Subsidy Contract A2-1.4-1) στο πλαίσιο του Προγράμματος Interreg IPA II Cross – Border Cooperation Programme GREECE – ALBANIA 2014 – 2020
Υπηρεσία της Σύμβασης	8. Σχεδίαση και ανάπτυξη της διαδικτυακής πύλης πληροφόρησης και ειδοποιήσεων
Παραδοτέο της Πράξης	4.1.3 Cross-Border Information & Notification Portal
Ημερομηνία παράδοσης	01.10.2020

Το Έργο συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση και από εθνικούς πόρους των κρατών που συμμετέχουν στο Πρόγραμμα

ΑΝΑΘΕΤΟΥΣΑ ΑΡΧΗ

	<p>ΕΘΝΙΚΗ ΣΥΝΟΜΟΣΠΟΝΔΙΑ ΑΤΟΜΩΝ ΜΕ ΑΝΑΠΗΡΙΑ</p> <p>Κεντρικά Γραφεία: Ελ. Βενιζέλου 236, Τ.Κ. 163 41, Ηλιούπολη Τηλ.: +30 210 99 49 837 Email: esaea@otenet.gr Ιστοσελίδα: www.esaea.gr</p>
---	---

ΑΝΑΔΟΧΟΣ

	<p>EUROPRAXIS μ.ΙΚΕ</p> <p>Βέργας 4, Τ.Κ. 17673 Καλλιθέα, Αθήνα Τηλ.: +30 210 82 10 895 Email: info@euro-praxis.com Ιστοσελίδα: www.euro-praxis.com</p>
	<p>EUROPEAN PROFILES A.E.</p> <p>Βατατζή 40, Τ.Κ. 114 72, Αθήνα Τηλ.: +30 210 82 10 895 Email: epmail@europeanprofiles.gr Ιστοσελίδα: www.europeanprofiles.gr</p>

Περιεχόμενα

1	Εισαγωγή	5
1.1	Ένθεση Παραδοτέου	5
1.2	Σχετικά με τη ζητούμενη Υπηρεσία	5
1.3	Αντικείμενο και σκοπός του Παραδοτέου	7
1.4	Δομή του Παραδοτέου	8
2	Πλαίσιο υλοποίησης του Συστήματος	9
2.1	Κρίσιμοι παράγοντες για την επιτυχία του Συστήματος.....	9
2.2	Πιθανοί κίνδυνοι - Αντιμετώπιση	10
3	Μεθοδολογική προσέγγιση	14
3.1	Εξειδίκευση της γενικής προσέγγιση με στόχο τη Χρηστικότητα (Ανθρωποκεντρική προσέγγιση)	14
3.2	Εξειδίκευση της γενικής προσέγγισης με στόχο την ευχρηστία (Μηχανική ευχρηστίας).....	16
3.3	Εξειδίκευση της γενικής προσέγγισης με στόχο την προσβασιμότητα (Μηχανική προσβασιμότητας)	17
3.4	Εξειδίκευση της γενικής προσέγγιση με στόχο την Αποδοχή (Συμμετοχική προσέγγιση)	18
3.5	Σχεδίαση.....	18
3.5.1	Σχεδίαση της εμπειρίας του χρήστη	21
3.5.2	Σχεδίαση της διεπαφής του χρήστη.....	23
3.5.3	Σχεδίαση λογισμικού.....	27
4	Προσδιορισμός απαιτήσεων	28
4.1	Εισαγωγή.....	28
4.2	Απαιτήσεις	30
4.3	Εγγραφή και αυθεντικοποίηση χρηστών / φορέων.....	33
5	Σχεδιασμός της εμπειρίας του χρήστη	36
5.1	Περιβάλλον της Πλατφόρμας.....	36
5.2	Σχεδίαση της Διεπαφής του χρήστη.....	36
5.3	Χρήστες.....	38

5.4	Δομή της Διεπαφής.....	38
5.5	Υλοποίηση της Διεπαφής	39
5.5.1	Ταχύτητα απόκρισης	39
5.5.2	Ανταποκρίσιμη διάταξη.....	39
5.5.3	Καλές πρακτικές.....	40
6	Αρχιτεκτονική συστήματος και εφαρμογών.....	42
6.1	Γενικά.....	42
6.2	Ανοιχτό λογισμικό και τεχνολογίες.....	42
6.3	Ανοιχτή Αρχιτεκτονική (open architecture)	44
6.3.1	Διασφάλιση της επεκτασιμότητας	44
6.4	Διασφάλιση της επικοινωνίας και της διασύνδεσης των υποσυστημάτων 46	
6.5	Ασφάλεια του συστήματος.....	48
6.6	Απρόσκοπτη αναβάθμιση του συστήματος (scaling)	50
6.7	Αρχιτεκτονική 3-βαθμίδων (3-tier)	50
7	Μεθοδολογία, μέσα και εργαλεία υλοποίησης.....	54
7.1	Γενικά.....	54
7.2	Προσέγγιση υλοποίησης.....	56
7.3	Μεθοδολογία Scrum	58
7.4	Διασφάλιση ποιότητας και ελέγχου λογισμικού	59
7.4.1	Sandboxing	60
7.4.2	Εκσφαλμάτωση	61
7.4.3	Ποιοτικός έλεγχος.....	62
7.4.4	Αγωγός έκδοσης (pipeline approach).....	63
7.5	Τεκμηρίωση	63

1 Εισαγωγή

1.1 Ένθεση Παραδοτέου

Η Εθνική Συνομοσπονδία Ατόμων με Αναπηρία, αποκαλούμενη στο εξής χάριν συντομίας «ΕΣΑμεΑ», σε συνέχεια Ηλεκτρονικού Ανοικτού Διαγωνισμού κάτω των ορίων (Αρ. πρωτ. 640/12-05-20, ΑΔΑΜ: 20PROC006688679 2020-05-12, ημ/νια διαγωνισμού 29.05.2020), ανέθεσε στην ένωση φορέων «EUROPRACTIS M.I.K.E. - EUROPEAN PROFILES A.E.» την εκτέλεση του Έργου «**Παροχή υπηρεσιών υλοποίησης παραδοτέων της Πράξης 4PLUS**» (εφεξής το «Έργο»).

Το ανωτέρω Έργο, με ημερομηνία υπογραφής της Σύμβασης την 03.08.2020, υλοποιείται στο πλαίσιο της Πράξης “*Public awareness, Preparedness, Participation and Coordination for Civil Protection for All*”, με κωδική ονομασία “4PLUS” και αριθμό Σύμβασης Επιχορήγησης (αγγλ. Subsidy Contract number) A2-1.4-1 (εφεξής η «Πράξη»), η οποία εντάσσεται στο Πρόγραμμα Interreg IPA II Cross – Border Cooperation Programme GREECE– ALBANIA 2014–2020.

Σε συνέχεια της υπογραφής της Σύμβασης του Έργου, η Ομάδα Έργου του Αναδόχου εκκίνησε τις εργασίες της, σύμφωνα με τη Σύμβαση, για την προετοιμασία του παρόντος Παραδοτέου.

Το παρόν Παραδοτέο εντάσσεται στην **8^η Υπηρεσία της Σύμβασης**, με τίτλο «*Σχεδίαση και ανάπτυξη της διαδικτυακής πύλης πληροφόρησης και ειδοποιήσεων*», η οποία εντάσσεται στο Παραδοτέο “*4.1.3 Cross-Border Information & Notification Portal*” της Πράξης 4PLUS.

Το παρόν αποτελεί το **πρώτο** στη σειρά Παραδοτέο της εν λόγω Υπηρεσίας, η οποία συνολικά περιλαμβάνει δύο (2) παραδοτέα:

- **Π8α** Έκθεση σχεδιασμού και πρωταρχική διαδραστική έκδοση της Πλατφόρμας (**το παρόν παραδοτέο**)
- **Π8β** Τελική, ολοκληρωμένη Πλατφόρμα – Έκθεση τεκμηρίωσης

1.2 Σχετικά με τη ζητούμενη Υπηρεσία

Το αντικείμενο της υπηρεσίας της Σύμβασης αφορά στην παροχή υπηρεσιών για την ανάπτυξη μιας διαδικτυακής πύλης (αγγλ. web portal) Πληροφόρησης και Ειδοποίησης για τη διασυνοριακή περιοχή. Πιο συγκεκριμένα, η πύλη περιλαμβάνει μια περιοχή για ευάλωτες ομάδες πολιτών και φροντιστές ή/και μέλη των οικογενειών τους που περιέχει πηγές και πόρους πληροφορίας, δημόσια δεδομένα, οδηγούς, εγχειρίδια, κλπ., καθώς και τον μηχανισμό αναφοράς συμβάντων γεγονότων. Επιπλέον, μια περιοχή για άλλους άμεσους δικαιούχους/εμπλεκόμενους που περιέχει διαδραστικό χάρτη με την τοποθεσία

των συμβάντων, περιβάλλον χρήστη για την ανταπόκριση σε αναφορές πολιτών, περιβάλλον χρήστη για ανακοινώσεις, προειδοποιήσεις και διαδικτυακές έρευνες.

Επίσης, στην Πύλη θα πρέπει να συμπεριλαμβάνονται πληροφορίες σε σχέση με την Πράξη (στόχοι, πλαίσιο υλοποίησης, δράσεις, αποτελέσματα, χρηματοδότηση, κ.λπ.), σε σχέση με τα μέλη της κοινοπραξίας της Πράξης και σε σχέση με την παραγωγή της πύλης.

Στόχος είναι, μέσω της πύλης, οι ενδιαφερόμενοι να υποστηρίζονται αποτελεσματικά και με προσβάσιμο τρόπο, τόσο πριν όσο και κατά την εξέλιξη ενός συμβάντος/μιας καταστροφής στην περιοχή.

Πιο συγκεκριμένα, η πύλη θα συγκεντρώνει χρήσιμες πληροφορίες για κατοίκους και επισκέπτες με αναπηρία/μειωμένη κινητικότητα και άτομα γ' ηλικίας με θεματολογία που περιλαμβάνει πληροφορίες για όλες τις φάσεις:

1. **Πρόληψη:** Προσωπικά σχέδια ενημέρωσης και ρύθμισης (πληροφορίες σχετικά με τις τοπικές υπηρεσίες έκτακτης ανάγκης, τα προγράμματα προ-καταχώρισης, τα ειδικά τοπικά ή περιφερειακά συστήματα προειδοποίησης, τα σχέδια εκκένωσης, πού να στραφούν για οδηγίες σε περίπτωση έκτακτης ανάγκης, κ.λπ.
2. **Έκτακτη ανάγκη:** Κατά την εξέλιξη ενός κινδύνου/καταστροφής: Σύστημα προειδοποίησης, εντοπισμός, λήψη καθοδήγησης κ.λπ., φόρμες ταχείας αξιολόγησης, επικοινωνία κινδύνου και έκτακτης ανάγκης, τοποθεσία και παροχή βοήθειας
3. **Αντιμετώπιση επιπτώσεων:** πληροφόρηση/ειδήσεις, ανακοινώσεις, εύρεση προσώπων, κ.λπ.
4. **Λειτουργία άσκησης:** προσομοίωση περιπτώσεων.

Η πύλη θα αφορά στο σύνολο της περιοχής του Προγράμματος, και από τις δύο πλευρές των συνόρων, και θα είναι διαθέσιμη σε τρεις γλώσσες (στην Αγγλική, την Αλβανική και την Ελληνική).

Η τεχνική λύση του Έργου απαιτείται να σχεδιαστεί και να υλοποιηθεί με τρόπο ώστε να διασφαλίζεται «καθολική πρόσβαση» (αγγλ. universal access), δηλαδή προσβασιμότητα στο ευρύτερο δυνατό φάσμα χρηστών, ανεξάρτητα των όποιων χαρακτηριστικών και ιδιαιτεροτήτων του χρήστη, όπως για παράδειγμα την ηλικία, το κοινωνικό προφίλ, το πολιτιστικό υπόβαθρο, την αναπηρία, κ.λπ. Για τον λόγο αυτό, η ανάπτυξη των ψηφιακών συστημάτων θα πρέπει να βασισθεί στις αρχές του «Σχεδιασμού για Όλους» (αγγλ. Design for All), καθώς και σε διεθνώς αναγνωρισμένους κανόνες ανάπτυξης περιεχομένου για τον Ιστό, συμπεριλαμβανομένου της νέας έκδοσης 2.0 των οδηγιών της Κοινοπραξίας του

Παγκόσμιου Ιστού (W3C) για την Προσβασιμότητα Περιεχομένου του Ιστού (αγγλ. Web Content Accessibility Guidelines, version 2.0 ή αλλιώς WCAG 2.0).

Ειδικότερα, η πύλη ζητείται να συμμορφώνεται πλήρως με διεθνή πρότυπα και οδηγίες προσβασιμότητας, και ειδικά με τις οδηγίες WCAG 2.0 να συμμορφώνεται στο ανώτατο επίπεδο AAA, ώστε να διασφαλίζεται η ευρύτερη δυνατή πρόσβαση στα μέλη της ομάδας-στόχου του εγχειρήματος.

Στο τεύχος της προκήρυξης (βλ. παράρτημα) προσδιορίζονται αναλυτικά τα ποιοτικά χαρακτηριστικά και οι απαιτούμενες τεχνικές προδιαγραφές της ζητούμενης Πύλης καθώς και των σχετικών συστημάτων Διαχείρισης Περιεχομένου.

Τα στάδια ανάπτυξης που προβλέπονται είναι τα ακόλουθα

- (1) Συλλογή/καταγραφή αναγκών - Σχεδίαση της εμπειρίας/διεπαφής χρηστών, της αρχιτεκτονικής, και των υποσυστημάτων και μηχανισμών της ολοκληρωμένης πλατφόρμας
- (2) Ανάπτυξη, ολοκλήρωση (αγγλ. integration) και δοκιμή (αγγλ. testing) των απαιτούμενων υποσυστημάτων - μηχανισμών
- (3) Εγκατάσταση - Πιλοτική λειτουργία
- (4) Αξιολόγηση - Διορθώσεις / βελτιώσεις (εάν απαιτούνται)
- (5) Εκπαίδευση των διαχειριστών και χειριστών του συστήματος
- (6) Τεκμηρίωση και παράδοση ολοκληρωμένης/τελικής Πλατφόρμας

1.3 Αντικείμενο και σκοπός του Παραδοτέου

Το παρόν τεύχος παρουσιάζει τη μεθοδολογία σχεδίασης και τον αναλυτικό σχεδιασμό της Πλατφόρμας με δεδομένα πληροφόρησης και ειδοποίησης για τη διασυνοριακή περιοχή του προγράμματος, με απώτερο σκοπό την συγκέντρωση χρήσιμων πληροφοριών για κατοίκους και επισκέπτες με αναπηρία/μειωμένη κινητικότητα και άτομα γ' ηλικίας με θεματολογία που περιλαμβάνει την πρόληψη, την αντιμετώπιση των επιπτώσεων μιας έκτακτης ανάγκης αλλά και τη λειτουργία άσκησης στην διασυνοριακή περιοχή και των επιμέρους συστατικών μερών του, συμπεριλαμβανομένης την οριστικής αρχιτεκτονικής, του αναλυτικού σχεδιασμού των υποσυστημάτων του και του σχεδιασμού της διεπαφής χρήστη για όλες τις κατηγορίες χρηστών.

Το παρόν Παραδοτέο, απευθύνεται:

- στα στελέχη της ΕΣΑμεΑ τον αναλυτικό σχεδιασμό της Πλατφόρμας, τεκμηριωμένα, με πληροφορίες ως προς τις αρχές και τα ζητήματα που οδήγησαν σε συγκεκριμένες σχεδιαστικές επιλογές, με τρόπο κατανοητό

και εύληπτο, συμπεριλαμβανομένου προσχεδίων των οθονών (mock-ups) του συστήματος, ώστε τα στελέχη της ΕΣΑμεΑ, με τη σειρά τους, να έχουν τη δυνατότητα, εύκολα και άμεσα, να τον μελετήσουν και να παραθέσουν τυχόν σχόλια και παρατηρήσεις που κρίνονται απαραίτητα για τη βελτίωση του συνολικού συστήματος και τη διασφάλιση της εκπλήρωσης των στόχων του.

- στα μέλη της Ομάδας Ανάπτυξης Προσβάσιμων Εφαρμογών ΤΠΕ του Αναδόχου που θα αναλάβουν τη υλοποίηση της Πλατφόρμας (προγραμματιστές) σύμφωνα με τον εν λόγω σχεδιασμό.

1.4 Δομή του Παραδοτέου

Το παρόν τεύχος ακολουθεί την εξής δομή:

- Στο **1^ο Κεφάλαιο** («Εισαγωγή») παρουσιάζονται τα γενικά στοιχεία του έργου στο οποίο εντάσσεται το παρόν Παραδοτέο, καθώς και άλλα στοιχεία που συμβάλλουν στην ανάγνωση και κατανόηση του.
- Στο **Κεφάλαιο 2** παρουσιάζεται το πλαίσιο υλοποίησης του Συστήματος με έμφαση στην ανάλυση των κρίσιμων παραγόντων για την επιτυχία της Πλατφόρμας καθώς και την παράθεση των τρόπων αντιμετώπισης κινδύνων.
- Στο **Κεφάλαιο 3** αναλύεται η μεθοδολογική προσέγγιση και ιδιαιτέρως η ανθρωποκεντρική προσέγγιση, η μηχανική ευχρηστίας και η μηχανική προσβασιμότητας καθώς και η συμμετοχική προσέγγιση και η εν γένει σχεδίαση του πληροφοριακού συστήματος.
- Στο **Κεφάλαιο 4** γίνεται ο προσδιορισμός του πλαισίου χρήσης και των απαιτήσεων.
- Στο **Κεφάλαιο 5** παρουσιάζεται ο σχεδιασμός της εμπειρίας του χρήστη (UX – User eXperience) βάσει καλών πρακτικών.
- Στο **Κεφάλαιο 6** αναλύεται η αρχιτεκτονική του συστήματος και οι τρόποι διασφάλισης της μελλοντικής του επέκτασης.
- Τέλος, στο **Κεφάλαιο 7** αναφέρεται η μεθοδολογία, τα μέσα και τα εργαλεία υλοποίησης του πληροφοριακού συστήματος.

2 Πλαίσιο υλοποίησης του Συστήματος

2.1 Κρίσιμοι παράγοντες για την επιτυχία του Συστήματος

Επιπλέον της ανάγκης επιλογής και εφαρμογής κατάλληλης μεθοδολογίας και τεχνικών σύμφωνα με τους γενικότερους περιορισμούς του Έργου, η Ομάδα Έργου του Αναδόχου καλείται να εντοπίσει και προετοιμαστεί κατάλληλα για μια σειρά από επιπλέον κρίσιμους παράγοντες για την επιτυχής υλοποίηση του.

Ενδεικτικά αναφέρονται οι εξής παράγοντες:

- Το **περιβάλλον λειτουργίας του συστήματος**, στο οποίο η παρεχόμενη τεχνική λύση θα τεθεί σε παραγωγική λειτουργία, δηλαδή τα χαρακτηριστικά και οι ανάγκες του Φορέα Λειτουργίας, γεγονός που απαιτεί την εφαρμογή εξειδικευμένων τεχνικών, μεθόδων και διαδικασιών ώστε να διασφαλισθεί η ορθότητα των προτεινόμενων λύσεων και την αποδοχή του συστήματος και των υπηρεσιών από όλους τους χρήστες-στόχος.
- Η **διασφάλιση και έλεγχος ποιότητας**. Μεγάλο βάρος πρέπει να δοθεί συνολικά στη διασφάλιση της ποιότητας των λειτουργιών της Πλατφόρμας και των υποσυστημάτων που την αποτελούν.
- **Αποτελεσματική κάλυψη των αναγκών και απαιτήσεων των τελικών χρηστών**. Θα πρέπει να διασφαλισθεί ότι η παρεχόμενη τεχνική λύση θα ικανοποιεί πλήρως τις απαιτήσεις και τις ανάγκες του προσωπικού και των στελεχών που θα κάνουν χρήση αυτής. Δηλαδή, θα πρέπει να διασφαλισθεί ότι η Πλατφόρμα θα διευκολύνει χωρίς να εισάγονται, άσκοπα, νέες, μη επιθυμητές, μη αποδοτικές ή/και μη αποτελεσματικές διαδικασίες.
- **Προσβασιμότητα για χρήστες / χειριστές με αναπηρία και συμβατότητα με διαδομένες Υποστηρικτικές Τεχνολογίες**. Είναι σημαντικό το πληροφοριακό σύστημα να σχεδιαστεί με τρόπο κατάλληλο, και βάσει ενδεδειγμένων πρακτικών, ώστε να διασφαλίζεται ότι όλες οι λειτουργίες θα μπορούν να χρησιμοποιηθούν χωρίς εμπόδια προσβασιμότητας, τόσο από τους πρωταρχικούς χρήστες τους, όσο και από μελλοντικούς χρήστες (νέο προσωπικό, χειροτέχνες, νέοι επισκέπτες).

2.2 Πιθανοί κίνδυνοι - Αντιμετώπιση

Είναι σαφές ότι το ζητούμενο Πληροφοριακό Σύστημα εξαρτάται από αντικειμενικούς παράγοντες δυσκολίας και κινδύνους. Η Ομάδα Έργου έχει ήδη καταγράψει και αναλύσει μια σειρά από πιθανούς κινδύνους, ώστε να ετοιμάσει σχετικά πλάνα για την έγκαιρη αναγνώρισή τους, τον μετριασμό της επίπτωσή τους, και την αντιμετώπισή τους εφόσον προκύψουν κατά την ανάπτυξη του Πληροφοριακού Συστήματος, το οποίο αποτυπώνεται στον παρακάτω Πίνακα.

Πίνακας 1. Πρωταρχική καταγραφή και αξιολόγηση Πιθανών Κινδύνων της Πλατφόρμας και σχετικά πλάνα αντιμετώπισης.

Κίνδυνος	Εκτίμηση Επίπτωσης	Πλάνο πρόληψης / αντιμετώπισης
Κίνδυνοι Οργανωσιακής Επάρκειας		
Ανεπαρκής ή επάνδρωση της ad hoc Επιτροπής αντιπροσωπευτικών Χρηστών και ΑμεΑ ή/και ανεπαρκής ανταπόκριση (συμμετοχικότητα) από μέλη της	Αναποτελεσματική εφαρμογή των προτεινόμενων προσεγγίσεων και τεχνικών που αφορούν στην ενεργή εμπλοκή αντιπροσωπευτικών και πραγματικών χρηστών στο έργο (βλ., UCD, Participatory Design, User Testing, Accessibility testing with Users, κ.λπ.)	<ul style="list-style-type: none"> Εξ αρχής συνεργασία με την Αναθέτουσα για την εξασφάλιση της υποστήριξης για πρόσκληση και εγγραφή στην Επιτροπή ικανού αριθμού μελών Δημιουργία reserve list (τουλάχιστον 25%)
Ελλιπής εκπαίδευση, τεχνική κατάρτιση ή ανταπόκριση προσωπικού του Φορέα Λειτουργίας σε θέματα που άπτονται του Έργου και της χρήσης των Τεχνικών Λύσεων και εκροών του Έργου.	Ανεπιτυχής ή/και μη αποτελεσματική Πιλοτική Λειτουργία (ΦΑΣΗ Α) για τη σταδιακή μετάπτωση της λειτουργίας των συστημάτων στην Αναθέτουσα Αρχή.	<ul style="list-style-type: none"> Ενίσχυση της Υπηρεσίας υποστηριζόμενης λειτουργίας από πλευράς Αναδόχου Ενίσχυση των πόρων για την Επιτηρούμενη και Επιβλεπόμενη Εφαρμογή από πλευράς Αναδόχου Ad hoc εκπαιδευτικές – ενημερωτικές δράσεις σε εξειδικευμένα θέματα (π.χ., στατιστικής, ψηφιακής προσβασιμότητας, επικοινωνιακού σχεδιασμού, κ.λπ.) για το προσωπικό από πλευράς Αναδόχου Δέσμευση πόρων (προσωπικό, χώρους, κ.λπ.) από πλευράς Αναθέτουσας.
Κίνδυνοι έναντι Τρίτων		
Χαμηλός βαθμός συμμόρφωσης – ευθυγράμμισης το Έργου με	Επιπτώσεις στην Αναθέτουσα Αρχή	<ul style="list-style-type: none"> Πιστή εφαρμογή του προτεινόμενου συστήματος ελέγχου ποιότητας του Έργου

τη στρατηγική του Προγράμματος		<ul style="list-style-type: none"> • Ad hoc παρεμβάσεις (εσωτερικές τεχνικές συναντήσεις, workshops κλπ) για την καθοδήγηση μελών της Ομάδας Έργου
Χαμηλός βαθμός συμμόρφωσης – ευθυγράμμισης το Έργου με τη στρατηγική και τις αρχές της ΕΣΑμεΑ	Επιπτώσεις στην Αναθέτουσα Αρχή ή/και στον Φορέα Λειτουργίας	<ul style="list-style-type: none"> • Πιστή εφαρμογή του προτεινόμενου συστήματος ελέγχου ποιότητας του Έργου • Συναντήσεις με στελέχη της ΕΣΑμεΑ και παρουσίαση αναφορών απολογιστικού και προγραμματιστικού χαρακτήρα για την τακτή και έγκαιρη επαλήθευση
Χαμηλός βαθμός συμμόρφωσης – ευθυγράμμισης το Έργου με τη στρατηγική, τις ανάγκες και τους στόχους του Παρατηρητηρίου	Επιπτώσεις στον Φορέα Λειτουργίας	<ul style="list-style-type: none"> • Πιστή εφαρμογή του προτεινόμενου μοντέλου συνεργατικού σχεδιασμού για την υλοποίηση του Έργου • Πιστή εφαρμογή του προτεινόμενου συστήματος ελέγχου ποιότητας του Έργου • Συναντήσεις με στελέχη του Παρατηρητηρίου (Επιτροπή καθοδήγησης) και παρουσίαση αναφορών απολογιστικού και προγραμματιστικού χαρακτήρα για την τακτή και έγκαιρη επαλήθευση
Χαμηλός βαθμός συμμόρφωσης – ευθυγράμμισης του Έργου με τις αρχές και τους όρους της Διεθνούς Σύμβασης του ΟΗΕ για τα Δικαιώματα των Ατόμων με Αναπηρία	Επιπτώσεις στον Φορέα Λειτουργίας ή/και στην Αναθέτουσα Αρχή	<ul style="list-style-type: none"> • Πιστή εφαρμογή του προτεινόμενου συστήματος ελέγχου ποιότητας του Έργου • Ενίσχυση της Επιτροπής οριζόντιας εφαρμογής των αρχών Ίσων Ευκαιριών και Μη διάκρισης με επιπλέον μέλη με εξειδίκευση στη Διεθνή Σύμβαση
Δυσκολίας στην κατανόηση και χρήση των εφαρμογών του Έργου ή/και μη ικανοποίησης των χρηστών από τον σχεδιασμό των εφαρμογών του Έργου	Χαμηλή αποδοχή και διείσδυση του συστήματος στις ομάδες-στόχος	<ul style="list-style-type: none"> • Προσέγγιση KISS (Keep It Simple Stupid Principle) κατά τον σχεδιασμό των πρώτων πρωτότυπων, ξεκινώντας από όσο το δυνατόν απλούστερες έννοιες και σχέδια (για να διασφαλισθεί η συμμετοχικότητα), με προοδευτική πολυπλοκότητα κατά τη πιλοτική λειτουργία (για να διασφαλισθεί η κατανόηση και αποδοχή από πλευράς τελικών χρηστών)

		<ul style="list-style-type: none"> Ενίσχυση των δράσεων εκπαίδευσης, με πιο εξατομικευμένες δράσης ή/και με περισσότερες ώρες πρακτικής εξάσκησης
Εκδήλωσης ζητημάτων που σχετίζονται με δικαιώματα πνευματικής ιδιοκτησίας	Διακουβέεται το κύρος του Αναδόχου και της Αναθέτουσας Αρχής	<ul style="list-style-type: none"> Ενίσχυση της Επιτροπής οριζόντιας εφαρμογής των αρχών Ίσων Ευκαιριών και Μη διάκρισης με επιπλέον μέλη με εξειδίκευση / καθήκον την παρακολούθηση όλων των δραστηριοτήτων και την προώθηση κατάλληλων διαδικασιών συναίνεσης όταν χρειαστεί.
Προσβολής / απώλειας ελέγχου απορρήτου (Loss of privacy control)	Έκθεση ευαίσθητων δεδομένων χρηστών με άμεσες ή έμμεσες συνέπειες στον Ανάδοχο και την Αναθέτουσα Αρχή	<ul style="list-style-type: none"> Ενημερωμένη συναίνεση (Informed consent - που παρέχεται σε σαφή και πλήρη γλώσσα) πριν από οποιαδήποτε συλλογή / παρακολούθηση δεδομένων. Γραπτή δέσμευση του προσωπικού του Αναδόχου για την τήρηση απορρήτου. Πιστή εφαρμογή του προτεινόμενου συστήματος ελέγχου ποιότητας του Έργου Ενίσχυση της Επιτροπής οριζόντιας εφαρμογής των αρχών Ίσων Ευκαιριών και Μη διάκρισης με επιπλέον μέλη με εξειδίκευση / καθήκον την παρακολούθηση όλων των δραστηριοτήτων και την προώθηση κατάλληλων διαδικασιών συναίνεσης όταν χρειαστεί.
Επιθέσεων από τρίτους σε δεδομένα αποθηκευμένα ή / και μεταφερόμενα μέσω διαδικτύου	Έκθεση, διαρροή και προσβολή της ασφάλειας προσωπικών δεδομένων χρηστών (στοιχεία επικοινωνίας, επικοινωνίες, κλπ.), με άμεσες ή έμμεσες συνέπειες στον Ανάδοχο και τον Φορέα Λειτουργίας	<ul style="list-style-type: none"> Ανάπτυξη των υψηλότερων δυνατών προτύπων ασφάλειας δεδομένων
Άνισης μεταχείρισης και διάκρισης των δύο φύλων	Διακουβέεται το κύρος του Φορέα Λειτουργίας και της Αναθέτουσας Αρχής	<ul style="list-style-type: none"> Πιστή εφαρμογή του προτεινόμενου συστήματος ελέγχου ποιότητας του Έργου και αδιάληπτη εμπλοκή της Επιτροπής οριζόντιας εφαρμογής των αρχών Ίσων Ευκαιριών και Μη διάκρισης.

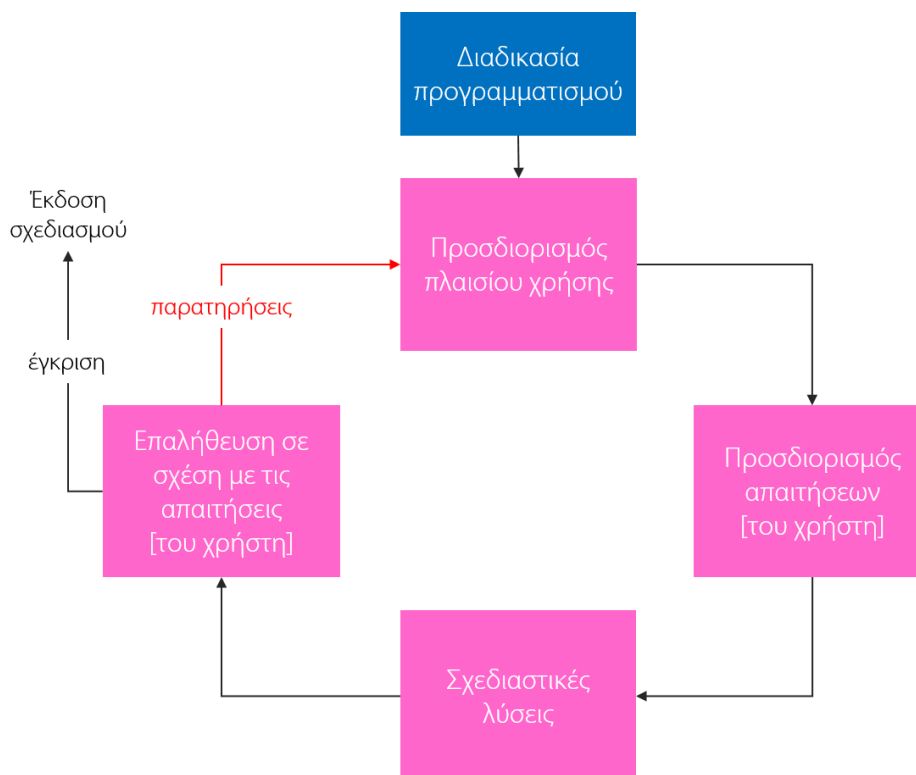
<p>Κακών πρακτικών κατά την αξιολόγηση λογισμικού με αντιπροσωπευτικούς ή πραγματικούς χρήστες</p>	<p>Εγκεκριμένη συναίνεση, Προστασία δεδομένων, Απόρρητο, Εμπιστευτικότητα</p>	<ul style="list-style-type: none"> • Εγκεκριμένη συναίνεση, Προστασία δεδομένων, Απόρρητο, Εμπιστευτικότητα • Έλεγχος συμμόρφωση με: <ul style="list-style-type: none"> ○ τις διατάξεις του νόμου περί προστασίας προσωπικών δικαιωμάτων Ν.2472/1997(ΦΕΚ Α 50/1997) περί προστασίας του ατόμου από την επεξεργασία δεδομένων προσωπικού χαρακτήρα, όπως τροποποιήθηκε με το Νόμο 3471/2006 (ΦΕΚ Α 133/2006) «Προστασία δεδομένων προσωπικού χαρακτήρα και της ιδιωτικής ζωής στον τομέα των ηλεκτρονικών επικοινωνιών και τροποποίηση του Ν.2472/1997» ○ Directive 95/46/EC, ○ Directive 2002/58/EC ○ Directive 99/5/EC
<p>Κακών πρακτικών κατά τη συλλογή στοιχείων του Προφίλ Ωφελούμενου / χειροτέχνη (κατά τη Πιλοτική Λειτουργία)</p>	<p>Εγκεκριμένη συναίνεση, Προστασία δεδομένων, Απόρρητο, Εμπιστευτικότητα</p>	<ul style="list-style-type: none"> • Εγκεκριμένη συναίνεση, Προστασία δεδομένων, Απόρρητο, Εμπιστευτικότητα • Έλεγχος συμμόρφωσης με: <ul style="list-style-type: none"> ○ τις διατάξεις του νόμου περί προστασίας προσωπικών δικαιωμάτων Ν.2472/1997(ΦΕΚ Α 50/1997) περί προστασίας του ατόμου από την επεξεργασία δεδομένων προσωπικού χαρακτήρα, όπως τροποποιήθηκε με το Νόμο 3471/2006 (ΦΕΚ Α 133/2006) «Προστασία δεδομένων προσωπικού χαρακτήρα και της ιδιωτικής ζωής στον τομέα των ηλεκτρονικών επικοινωνιών και τροποποίηση του Ν.2472/1997». ○ Directive 95/46/EC, ○ Directive 2002/58/EC ○ Directive 99/5/EC ○ Χάρτα Θεμελιωδών Δικαιωμάτων ○ World Medical Association Ethics

3 Μεθοδολογική προσέγγιση

3.1 Εξειδίκευση της γενικής προσέγγιση με στόχο τη Χρηστικότητα (Ανθρωποκεντρική προσέγγιση)

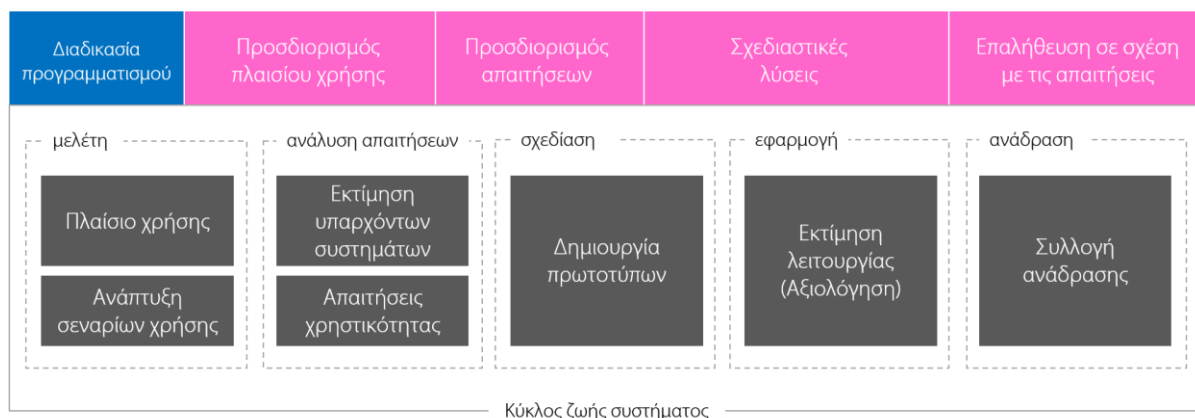
Για την ανάπτυξη του Πληροφοριακού Συστήματος, η Ομάδα Έργου βασίζεται στην προσέγγιση της ανθρωποκεντρικής σχεδίασης (human-centered design), η οποία επικεντρώνεται στη δημιουργία και ανάπτυξη χρήσιμων προϊόντων και συστημάτων και έχει τις εξής βασικές αρχές:

- Εστιάζουμε στους χρήστες του συστήματος και τις εργασίες που επιτελούν με αυτό, από τις αρχικές φάσεις σχεδίασης.
- Σε όλες τις φάσεις σχεδιασμού επαληθεύουμε με τους χρήστες, μέσω προσχεδίων, διαδραστικών προτύπων, παρουσιάσεων, κλπ., για να επιβεβαιώσουμε τις επιλογές μας.
- Ακολουθούμε επαναληπτική διαδικασία σχεδίασης, όπου κάθε φάση είναι μια μικρογραφία του κύκλου ζωής του λογισμικού με: Ανάλυση και συγγραφή ή βελτίωση των απαιτήσεων, Σχεδίαση ή αναθεώρηση του σχεδιασμού, ενδιάμεση υλοποίηση πρωτοτύπου ή τελικού προϊόντος, και αξιολόγηση/έλεγχος πρωτοτύπου ή τελικού προϊόντος.



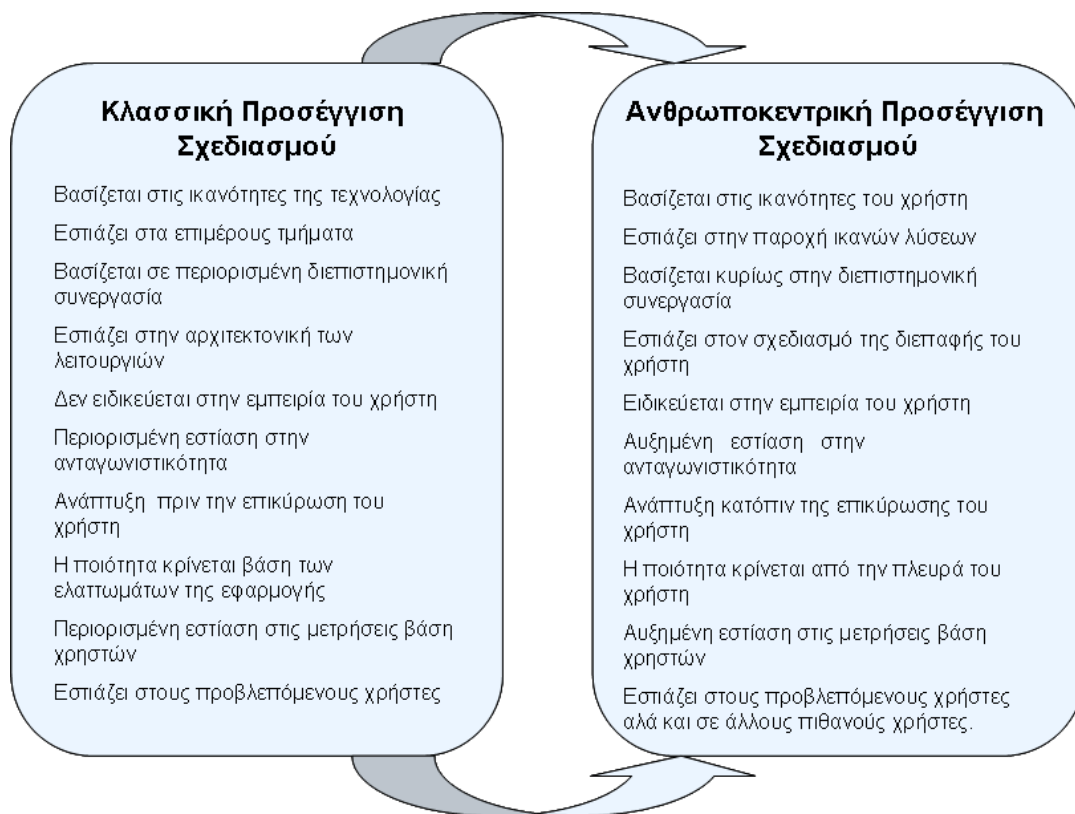
Εικόνα 1. Βασικές φάσεις της ανθρωποκεντρικής σχεδίασης, σύμφωνα με το πρότυπο ISO 9421-210 (2010)

Η συγκεκριμένη προσέγγιση είναι τυποποιημένη και περιγράφεται αναλυτικά στο standard ISO 9421-210¹ (2010), ως προς τις φάσεις και τις διαδικασίες (Εικ. 1 και Εικ 2).



Εικόνα 2. Ανάλυση της σχεδιαστικής διαδικασίας, σύμφωνα με το πρότυπο ISO 9421-210 (2010)

Στην παρακάτω εικόνα συνοψίζονται τα πλεονεκτήματα αυτής της προσέγγισης σε αντίθεση με τις κλασικές προσεγγίσεις ανάπτυξης εφαρμογών.



Εικόνα 3. Πλεονεκτήματα της ανθρωποκεντρικής σχεδίασης έναντι των κλασικών προσεγγίσεων

¹ Η έκδοση παραμένει ενεργή από το 2010, όταν αντικατέστησε το ISO 13407 (1999), και αναθεωρήθηκε για τελευταία φορά και επιβεβαιώθηκε το 2015.

3.2 Εξειδίκευση της γενικής προσέγγισης με στόχο την ευχρηστία (Μηχανική ευχρηστίας)

Η Ομάδα Έργου, έχοντας ως πρωτεύων στόχο οι εφαρμογές του Έργου να είναι εύκολες στη χρήση από τους τελικούς χρήστες στους οποίους απευθύνεται, σκοπεύει να μην εφαρμόσει κάποιο από τα κλασσικά και πλέον απαρχαιωμένα μοντέλα μηχανικής λογισμικού (software engineering), όπως για παράδειγμα το Waterfall Model (βλ. Pressman 1982, 1987, 1994), αλλά ένα σύγχρονο μοντέλο μηχανικής ευχρηστίας (usability engineering - βλ. Nielsen, 1993) και συγκεκριμένα το μοντέλο της Mayhew (1999).

Στο πρότυπο ISO 9241, Μέρος 11 (1998), η ευχρηστία ορίζεται ως «ο βαθμός στον οποίο ένα σύστημα μπορεί να χρησιμοποιηθεί από συγκεκριμένους χρήστες και σε συγκεκριμένο πλαίσιο χρήσης, για να επιτύχουν συγκεκριμένους στόχους με αποτελεσματικότητα, αποδοτικότητα και ικανοποίηση». Επίσης, στο ίδιο πρότυπο περιγράφεται και ένα εξελιγμένο πλαίσιο για τον καθορισμό και τη μέτρηση της ευχρηστίας των συστημάτων, σύμφωνα με το οποίο «η ευχρηστία μετρείται από το βαθμό στον οποίο επιτυγχάνονται οι επιδιωκόμενοι στόχοι των χρηστών (αποτελεσματικότητα), οι πόροι που έχουν δαπανηθεί για την επίτευξη των στόχων αυτών (αποδοτικότητα), και ο βαθμός στον οποίο οι χρήστες βρίσκουν τη χρήση του συστήματος αποδεκτή (ικανοποίηση)». Στο πρότυπο τονίζεται η σημασία των ειδικών συνθηκών υπό τις οποίες χρησιμοποιείται το σύστημα και τονίζεται το πλαίσιο της χρήσης, δηλαδή, τα καθήκοντα του χρήστη, ο εξοπλισμός του χρήστη, και το περιβάλλον στο οποίο λαμβάνει χώρα η αλληλεπίδραση.

Υπό το πρίσμα των παραπάνω, το μοντέλο μηχανικής ευχρηστίας συμπεριλαμβάνει:

- Οργανωμένες διεργασίες ανάλυσης των απαιτήσεων ευχρηστίας (usability requirements)
- Μια αποκλειστική διεργασία για τον προσδιορισμό των στόχων ευχρηστίας, βάσει των δεδομένων της ανάλυσης των απαιτήσεων (requirements analysis)
- Διεργασίες που υποστηρίζουν την οργανωμένη, top-down προσέγγιση για τον σχεδιασμό της διεπαφής του χρήστη, βάσει των στόχων ευχρηστίας και των υπολοίπων απαιτήσεων.
- Διεργασίες για την αντικειμενική αξιολόγηση στο πλαίσιο των επαναληπτικών σχεδιασμών με κατεύθυνση την πλήρη ικανοποίηση των στόχων ευχρηστίας.

Κατά κανόνα οι διενέργειες αυτές λαμβάνουν χώρα παράλληλα με τις κλασικές ενέργειες υλοποίησης της εφαρμογής (π.χ. διενέργειες ελέγχου, εκσφαλμάτωσης, αξιολόγησης, κλπ.).

3.3 Εξειδίκευση της γενικής προσέγγισης με στόχο την προσβασιμότητα (Μηχανική προσβασιμότητας)

Η Ομάδα Έργου, αναγνωρίζοντας της σημασία της προσβασιμότητας για τον Φορέα Λειτουργίας και το προσωπικό του, και με βάση την εμπειρία του στην ανάπτυξη προσβάσιμου περιεχομένου Ιστού, στην προσέγγισή του για την ανάπτυξη της Πλατφόρμας έχει εντάξει σύγχρονες πρακτικές μηχανικής προσβασιμότητας, ώστε όλες οι λειτουργίες του συστήματος να είναι προσβάσιμες και εύχρηστες και σε χρήστες με αναπηρία, εστιάζοντας στις ανάγκες των χρηστών με αναπηρίες κινητικές και όρασης.

Η προσβασιμότητα του Πληροφοριακού Συστήματος θα πρέπει να διασφαλίζεται για **κάθε χρήστη**, και για κάθε λειτουργία του τελικού συστήματος θα πρέπει να επιβεβαιωθεί ότι **όλοι** μπορούν ακολουθώντας συγκεκριμένα βήματα να κάνουν **πλήρη χρήση** τους.

Ως προς τη βασική προσέγγιση μηχανικής της προσβασιμότητας, η Ομάδα Έργου βασίζεται κυρίως στις οδηγίες και στις τεχνικές που συστήνονται από την Κοινοπραξία του Παγκόσμιου Ιστού (W3C), συμπεριλαμβανομένων αυτών που αφορούν στην «*Εμπλοκή χρηστών σε έργα Ιστού για καλύτερη και ευκολότερη προσβασιμότητα*»².

Ως προς τη διασφάλιση της προσβασιμότητας του Πληροφοριακού Συστήματος για τους χρήστες / χειριστές του (δηλ. για τα μέλη του προσωπικού του Φορέα Λειτουργίας), η κατασκευή των επιμέρους συστατικών του θα γίνει σύμφωνα με τις «*Οδηγίες για την Προσβασιμότητα του Περιεχομένου του Ιστού, έκδοση 2.0*» (WCAG 2.0) της Κοινοπραξίας του Παγκόσμιου Ιστού W3C (διαθέσιμες και με τη μορφή προτύπου ISO/IEC 40500:2012), δίνοντας ιδιαίτερη έμφαση σε οδηγίες και τεχνικές που αφορούν σε χρήστες με προβλήματα όρασης και κινητικά προβλήματα στα άνω άκρα και σε χρήστες που αλληλοεπιδρούν με τον υπολογιστή με τη βοήθεια ειδικών υποστηρικτικών τεχνολογιών, και με στόχο το τελικό σύστημα να συμμορφώνεται πλήρως τουλάχιστον στο επίπεδο «A» (WCAG 2.0, level A).

Επιπλέον, λόγω της φύσης του έργου, ως σύστημα μέσω του οποίου, μεταξύ άλλων, θα παράγεται και περιεχόμενο το οποίο θα δημοσιεύεται στην κεντρική σελίδα, η κατασκευή του θα πρέπει, σύμφωνα και με την Κοινοπραξία του Παγκόσμιου Ιστού W3C, να υποστηρίζει / διασφαλίζει ότι περιεχόμενο που θα δημιουργείται θα είναι και αυτό προσβάσιμο στους τελικούς χρήστες (δηλ. στους επισκέπτες της Πύλης). Επομένως, η κατασκευή των ανάλογων συστατικών μερών του ΟΠΣ θα γίνει και σύμφωνα με τις «*Οδηγίες Προσβασιμότητας για*

² Βλ. <https://www.w3.org/WAI/eval/users>

Εργαλεία Συγγραφής, έκδοση 2.0» (ATAG), με στόχο το τελικό σύστημα να συμμορφώνεται πλήρως τουλάχιστον στο επίπεδο «Α» (ATAG 2.0, level A).

Τέλος, στόχος είναι το Πληροφοριακό Σύστημα να υποστηρίζει πρόσβαση μέσω φορητών συσκευών τηλεφώνου και tablets, εφαρμόζοντας στον βαθμό που είναι εφικτό ανάλογες κατάλληλες πρακτικές (π.χ. ευέλικτη σχεδίαση).

3.4 Εξειδίκευση της γενικής προσέγγιση με στόχο την Αποδοχή (Συμμετοχική προσέγγιση)

Πάγια φιλοσοφία και πρακτική της Ομάδας Έργου για την ανάπτυξη του συστήματος είναι η *συμμετοχική σχεδίαση και ανάπτυξη* (participatory design & development), η οποία εύστοχα συνοψίζεται σε ένα παλιό Αράβικο γνωμικό: «*Η επαφή με το κοινό είναι μια αρχή. Η συνέχιση της επαφής είναι μια πρόοδος. Η από κοινού εργασία είναι επιτυχία*». Η προσέγγιση αυτή έχει μακριά διαδρομή και έχει χρησιμοποιηθεί σε ευρύ φάσμα ζητημάτων (έρευνα αγοράς, κοινωνικές έρευνες κ.λπ.) από διάφορες επιστημονικές ειδικότητες και σε διάφορα πεδία στα οποία έχει βρει εφαρμογή.

Στην περίπτωση της ανάπτυξης λογισμικού, αφορά στη συμμετοχή του «κοινού» (χρήστες και ομάδων συμφερόντων (stakeholders)) στη διαδικασία λήψης αποφάσεων σε όλες τις κρίσιμες φάσεις του Έργου. Στο πλαίσιο αυτό, η Ομάδα Έργου εφαρμόζει ένα στενό πλάνο συνεργασίας και επικοινωνίας με τους χρήστες-στόχος από πλευράς του Φορέα Λειτουργίας σε όλες τις φάσεις της ανάπτυξης του συστήματος (καθορισμός των απαιτήσεων, σχεδίαση, υλοποίηση, αξιολόγηση, πιλοτική εφαρμογή, κ.λπ.).

3.5 Σχεδίαση

Η **σχεδίαση** (αγγλ. *design*) ενός συστήματος είναι η διαδικασία κατά την οποία, με βάση μια σειρά από δεδομένες απαιτήσεις, δημιουργούμε ένα ή περισσότερα σχέδια του συστήματος προς κατασκευή, ώσπου να καταλήξουμε σε ένα προκρινόμενο σχέδιο το οποίο και *σχεδιασμό* του συστήματος.

Ο *σχεδιασμός*, καταρχήν, είναι η βάση αναφοράς για την υλοποίηση και λειτουργεί ως χάρτης για το τι και πως πρέπει να υλοποιηθεί. Επιπλέον, είναι ιδιαίτερα χρήσιμος και σε μετέπειτα φάσεις, όπως στις φάσεις της παραγωγικής χρήσης και της συντήρησης των εφαρμογών. Για παράδειγμα, τα σχέδια μιας οθόνης του συστήματος ή μιας διαδικασίας που υποστηρίζεται από το σύστημα μπορούν να χρησιμοποιηθούν στα εγχειρίδια χρήσης, ενώ τα αρχιτεκτονικά σχέδια μπορεί να χρησιμοποιηθούν από τον συντηρητή για την προκαθορισμένη συντήρηση και αποκατάσταση βλαβών.

Γενικά, η σχεδίαση αφορά σε δύο διακριτά μέρη: (α) την *εσωτερική* και την *εξωτερική* σχεδίαση:

- Η εσωτερική σχεδίαση έχει ως στόχο να σχεδιάσει το *λογισμικό*, δηλαδή να καταλήξει: (α) στον σχεδιασμό της αρχιτεκτονικής του λογισμικού και (β) στον αναλυτικό σχεδιασμό των συστατικών μερών του λογισμικού.
- Η εξωτερική σχεδίαση έχει ως στόχο να σχεδιάσει τα συστατικά μέρη της *εμπειρίας του χρήστη*, δηλαδή οτιδήποτε έχει να κάνει με την «εικόνα» που θα έχει το σύστημα, με το πως θα «βλέπει» ο χρήστης το σύστημα, και με το πως θα επικοινωνεί ο χρήστης με το σύστημα, ώστε να καταλήξει: (α) στον σχεδιασμό της αρχιτεκτονικής της *αλληλεπίδρασης του χρήστη*, (β) στον αναλυτικό σχεδιασμό των συστατικών μερών της *διεπαφής του χρήστη*, και (γ) στον σχεδιασμό όλων εκείνων των εξωτερικών στοιχείων που κρίνεται απαραίτητο να πλαισιώσουν τη διεπαφή του χρήστη και να συμβάλουν στο προοδευτικό χτίσιμο της εμπειρίας του χρήστη (π.χ., μέσα προώθησης, εγχειρίδια χρήσης, υπηρεσία εξυπηρέτησης χρηστών, κ.λπ.).

Για τη σχεδίαση του Πληροφοριακού Συστήματος, η Ομάδα Έργου απέδωσε ιδιαίτερη έμφαση στο εξής σημεία:

- στον διαχωρισμό εξωτερικής και εσωτερικής σχεδίασης, και στη σημασία της εξωτερικής σχεδίασης,
- στο ότι η εξωτερική σχεδίαση θα πρέπει να λαμβάνει υπόψη όλες τις υπάρχουσες απαιτήσεις, να είναι ανεξάρτητη της υλοποίησης³ και να παράγει τη βάση για την έναρξη της εσωτερικής σχεδίασης (δηλ. της σχεδίασης του λογισμικού),
- στο ότι τα αποτελέσματα της εξωτερικής σχεδίασης (δηλ. ο εξωτερικός σχεδιασμός) θα πρέπει να παρουσιάζουν το πως θα εισάγεται το σύστημα στον χρήστη, το πως θα ανταποκρίνεται το σύστημα σε κάθε ενέργεια του χρήστη, και το πως θα υποστηρίζεται ο χρήστης εκτός συστήματος,

³ Πολλές ομάδες υλοποίησης έχουν τη λάθος αντίληψη, πως η εξωτερική σχεδίαση όχι μόνο δεν είναι ανεξάρτητη της εσωτερικής σχεδίασης, αλλά μάλιστα συχνά έχουν την πεποίθηση - απαίτηση ο εσωτερικός σχεδιασμός να καθορίζει τον εξωτερικό σχεδιασμό. Δηλαδή, με βάση τις απαιτήσεις, ξεκινούν απευθείας να σχεδιάζουν και να υλοποιούν το παραγωγικό λογισμικό, και όταν πλέον έχουν προχωρήσει αρκετά, ξεκινούν να σχεδιάζουν τη διεπαφή του χρήστη για τη λειτουργικότητα που έχουν αναπτύξει. Η προσέγγιση αυτή αποδεδειγμένα οδηγεί σε καταστροφικά προβλήματα ευχρηστίας, τα οποία μάλιστα δύσκολα αντιμετωπίζονται αφού η υλοποίηση έχει ήδη προχωρήσει πολύ και χρειάζεται να ακυρωθεί μεγάλο μέρος της παραγωγής.

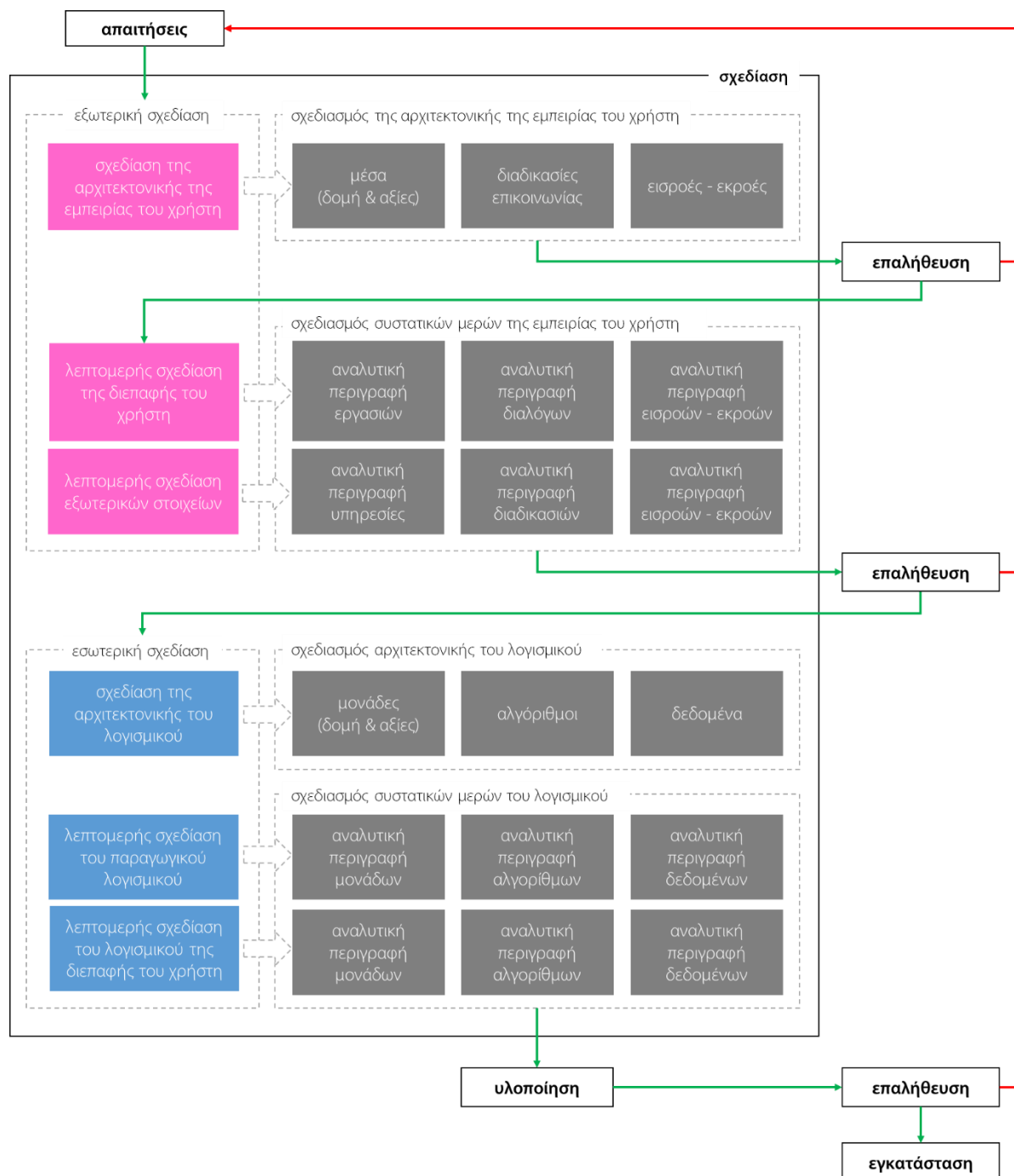
- ο στο ότι ο εξωτερικός σχεδιασμός θα πρέπει να παρουσιάζεται με τρόπο που να είναι κατανοητός στο χρήστη και γενικότερα σε όσους θα συμμετέχουν στην επαλήθευση του σχεδιασμού ,
- ο στο ότι η σχεδίαση της εμπειρίας τους χρήστη, είναι γενικότερη της σχεδίασης της διεπαφής χρήστη,
- ο στο ότι, όπως γίνεται με τη σχεδίαση του λογισμικού, έτσι και με τη σχεδίαση της εμπειρίας του χρήστη θα πρέπει να διαχωρίζεται σε σχεδίαση της «αρχιτεκτονικής της εμπειρίας του χρήστη» σε ανώτερο επίπεδο (high level) και σε λεπτομερή σχεδίαση των συστατικών μερών της,
- ο στο ότι η διεπαφή του χρήστη έχει διακριτό λογισμικό (κώδικα), ο οποίος θα πρέπει κανονικά να σχεδιάζεται ανεξάρτητα από τη σχεδίαση των παραγωγικών λειτουργιών του συστήματος ώστε να διασφαλίζεται η δυνατότητα να επικαιροποιείται το ένα χωρίς να επηρεάζεται το άλλο, και
- ο στο ότι το λογισμικό της διεπαφής του χρήστη θα πρέπει και αυτό να σχεδιάζεται με βάσει τις αρχές που διέπουν γενικά την ανάπτυξη ποιοτικού κώδικα.

Επομένως, σε πρώτο βαθμό, τα αποτελέσματα της εξωτερικής σχεδίασης απευθύνονται πρωτίστως στους χρήστες και στις ομάδες συμφέροντος (stakeholders), σε δεύτερο βαθμό απευθύνονται στους σχεδιαστές του λογισμικού, και τέλος, στους σχεδιαστές υλικού τεκμηρίωσης και εκπαίδευσης για τους χρήστες, κ.λπ.

Τα δε αποτελέσματα της εσωτερικής σχεδίασης απευθύνονται στην ομάδα υλοποίησης της εφαρμογής (προγραμματιστές). Παρουσιάζει την εσωτερική αρχιτεκτονική του συστήματος, τη δομή του λογισμικού, τους αλγόριθμους, τα δεδομένα, τις ροές και τις δομές των δεδομένων και περιγράφει πως θα υλοποιηθούν οι λειτουργίες του με κάθε λεπτομέρεια. Δηλαδή, περιλαμβάνει όλες τις τεχνικές λεπτομέρειες και τον τρόπο υλοποίησης του προς ανάπτυξη συστήματος. Ο αρχιτεκτονικός σχεδιασμός απεικονίζει τις μονάδες (δηλ. δομικά στοιχεία που συμβολίζουν τις λειτουργίες της εφαρμογής), χωρίς να παρέχει λεπτομερείς πληροφορίες για το πώς αυτές θα υλοποιηθούν. Αυτό γίνεται στη λεπτομερή σχεδίαση του λογισμικού. Στο λεπτομερές σχέδιο περιγράφεται ο αλγόριθμος που χρησιμοποιεί κάθε μονάδα, οι δομές δεδομένων, οι τύποι δεδομένων καθώς και οι σχέσεις μεταξύ λειτουργιών και δεδομένων.

Τα παραπάνω αποτυπώνονται στην προσέγγιση της Ομάδας Έργου για τη σχεδίαση του συστήματος, (βλ. **Εικόνα 4**), σύμφωνα με την οποία ξεκίνησε πρώτα με την εξωτερική σχεδίαση, και αφότου την επικύρωσε με πραγματικούς

χρήστες και ομάδες συμφέροντος (stakeholders), συνέχισε στον εσωτερικό σχεδιασμό, ώστε να ξεκινήσει την υλοποίηση.



Εικόνα 4. Η προσέγγιση της Ομάδας Έργου για τη σχεδίαση

3.5.1 Σχεδίαση της εμπειρίας του χρήστη

Η *εμπειρία του χρήστη* (user experience ή UX) αφορά στη συνολική εμπειρία που προσφέρει ένα σύστημα στον χρήστη. Η εμπειρία αυτή αποτελείται και

επιηρεάζεται από πολλούς παράγοντες. Η πρώτη απαίτηση για μια ποιοτική εμπειρία χρήστη είναι το νέο σύστημα να ανταποκριθεί στις ακριβείς ανάγκες του χρήστη και να του «λύσει τα χέρια». Εντούτοις, η λογική / λειτουργική πληρότητα ενός συστήματος δεν αποτελεί από μόνη της ικανή συνθήκη για την επιτυχία του συστήματος. Στη συνέχεια έρχεται η ποιότητα της αλληλεπίδρασης με το σύστημα. Βασικοί παράγοντες είναι η προσβασιμότητα, η απλότητα, και γενικότερα η ευχρηστία του συστήματος.

Είναι σημαντικό, όμως, να διακρίνεται η συνολική εμπειρία χρήστη από την *ποιότητα της αλληλεπίδρασης* και τα χαρακτηριστικά της διεπαφής του χρήστη, παρόλο που η διεπαφή του χρήστη είναι προφανώς ένα εξαιρετικά σημαντικό μέρος της σχεδίασης της εμπειρίας όπως αναφέρθηκε παραπάνω.

Επιπλέον, πέρα από τέτοια ενδογενή χαρακτηριστικά μιας εφαρμογής (με τα οποία ασχολείται σε η σχεδίαση της εμπειρίας του χρήστη ανώτερο επίπεδο και στη συνέχεια, σε περισσότερο βάθος η λεπτομερής σχεδίαση της διεπαφής), σημαντικό ρόλο παίζουν και μια σειρά άλλων εξωτερικών χαρακτηριστικών που επηρεάζουν τον βαθμό στον οποίο θα υποστηρίζονται νέοι χρήστες, και τον βαθμό στον οποίο θα διατηρούνται οι υφιστάμενοι χρήστες ικανοποιημένοι.

Καταλήγοντας, η σχεδίαση μιας υποδειγματικής εμπειρίας του χρήστη υπερβαίνει κατά πολύ αυτά που χρήστης αναμένει από το σύστημα καθαυτό. Η σχεδίαση υψηλής ποιότητας εμπειρίας του χρήστη, αφορά στη σχεδίαση για αξίες, και όχι αποκλειστικά στη σχεδίαση της διεπαφής. Τέτοιες αξίες είναι η ευρεσιμότητα, η χρησιμότητα και η ελκυστικότητα, η προσβασιμότητα και η ευχρηστία, η ασφάλεια και η αξιοπιστία, η διαχρονικότητα, κ.λπ.

Συνολικά, η προσέγγιση που έχει επιλέξει η Ομάδα Έργου για τη σχεδίαση της εμπειρίας του χρήστη, σε συνάρτηση με την επιλογή της προσέγγισης *agile development* για την υλοποίηση των εφαρμογών του Έργου είναι η *Agile UX*.

Στην κατεύθυνση αυτή, η Ομάδα Έργου βασίστηκε στη μεθοδολογία της Google, *Design Sprint*⁴, και οργάνωσε *focus groups* (εξ αποστάσεως, μέσω τηλεδιασκέψεων) με διάφορους εκπροσώπους-κλειδιά από τις διάφορες κατηγορίες χρηστών (*users*) και εμπλεκομένων (*stakeholders*) από πλευράς του Φορέα λειτουργίας, αποδίδοντας προτεραιότητα στη σχεδίαση του συνόλου της εμπειρίας του χρήστη. Τα ευρήματα αυτή της προσπάθειας, αφού επικυρώθηκαν (*concept testing*), αποτυπώθηκαν μαζί με τον υπόλοιπο σχεδιασμό στο παρόν παραδοτέο, ώστε στη συνέχεια, από κοινού με τα στελέχη της Αναθέτουσας, να αποφασιστούν οι απαραίτητες ενέργειες για την υλοποίησης του σχεδιασμού των εξωτερικών στοιχείων, πέραν της διεπαφής.

4 Βλ. <https://www.interaction-design.org/literature/article/make-your-ux-design-process-agile-using-google-s-methodology>

3.5.2 Σχεδίαση της διεπαφής του χρήστη

Με τον όρο «*διεπαφή χρήστη*» μιας εφαρμογής, εννοούμε τα μέρη αυτά τα οποία ο χρήστης **αντιλαμβάνεται άμεσα με τις αισθήσεις του** και τα στοιχεία εκείνα με τα οποία μπορεί να **αλληλεπιδράσει** για να επικοινωνήσει τις εντολές του στο σύστημα και να κάνει χρήση των λειτουργιών που αυτό προσφέρει.

Η διεπαφή του χρήστη είναι υπεύθυνη από πλευρά συστήματος για:

- να ενημερώσει τους (πιθανούς) χρήστες σχετικά με τις διαθέσιμες λειτουργίες του συστήματος, τους (εναλλακτικούς) τρόπους χρήσης της κάθε λειτουργίας, και τις αλλαγές στην κατάσταση του συστήματος κατά την εξέλιξη της χρήσης,
- να λαμβάνει μηνύματα από τον χρήστη, να καταγράφει τις αντιδράσεις του και να τα μεταφράζει σε εντολές προς το σύστημα και
- να παρουσιάζει τα αποτελέσματα.

Επομένως, η σχεδίαση της διεπαφής είναι πολύ σημαντική για τη λειτουργία / χρήση ενός συστήματος και πρέπει να σχεδιάζεται με τέτοιο τρόπο ώστε οι χρήστες να είναι σε θέση να επιτύχουν τους στόχους τους όσο το δυνατόν πιο αποτελεσματικά, χωρίς να εστιάζουν πολύ στην ίδια τη διεπαφή χρήστη. Με αυτή την έννοια, μια καλά σχεδιασμένη διεπαφή χρήστη καθίσταται στην πράξη ανεπαίσθητη στον χρήστη, ο οποίος εστιάζει στο περιεχόμενο των λειτουργιών και όχι στο να καταλάβει το πως παρουσιάζονται και λειτουργούν.

Βασικά ποιοτικά στοιχεία που θα διέπουν τη διεπαφή του χρήστη είναι:

- Συνέπεια: Οι εφαρμογές θα πρέπει να έχουν ομοιόμορφη εμφάνιση και να τηρείται συνέπεια στη χρήση των λεκτικών και των συμβόλων. Το λεξιλόγιο που χρησιμοποιείται για την περιγραφή εννοιών, σημείων και λειτουργιών σε όλο το εύρος των εφαρμογών και των συστημάτων πρέπει να είναι συνεπές. Αντίστοιχη συνέπεια πρέπει να επιδεικνύουν οι οποιεσδήποτε γραφικές απεικονίσεις, η διαμόρφωση σελίδων και η τοποθέτηση αντικειμένων στο χώρο των ιστοσελίδων. Στο επίπεδο των εφαρμογών και διαδραστικών λειτουργιών, παρόμοιες λεκτικές και λειτουργικές απεικονίσεις πρέπει να αντιστοιχούν σε ανάλογα αποτελέσματα. Το σύστημα θα πρέπει να προσφέρει όμοιο περιβάλλον σε όλα τα υποσυστήματα του, όπως: Λίστες λειτουργιών (menu), Εργαλειοθήκες (toolbars), συντομεύσεις λειτουργιών (keyboard shortcuts).
- Αξιοπιστία: Ο χρήστης θα πρέπει να έχει σαφείς διαβεβαιώσεις αλλά και επίκτητη αντίληψη δια μέσου της εμφάνισης και συμπεριφοράς του συστήματος ότι: (α) οι πληροφορίες που εισάγει στο σύστημα είναι σωστές και αρκετές (ελαχιστοποίηση λαθών χρήστη μέσω ολοκληρωμένου πρωτοβάθμιου ελέγχου), (β) οι πληροφορίες που λαμβάνει από το

σύστημα είναι ακριβείς και επικαιροποιημένες και (γ) η συμπεριφορά του συστήματος είναι προβλέψιμη.

- Απόκριση: Οι λειτουργίες των εφαρμογών θα πρέπει να έχουν ικανοποιητικό χρόνο απόκρισης ≤ 3 sec. Στο χρόνο απόκρισης δεν συμπεριλαμβάνεται ο χρόνος καθυστέρησης που οφείλεται στις δικτυακές υποδομές στην πλευρά του χρήστη. Τα βήματα και οι ενέργειες από την πλευρά του χρήστη για κάθε επιθυμητή λειτουργία πρέπει να είναι ελαχιστοποιημένα και ανάλογα με το προφίλ του. Η δυνατότητα αυτόματων βελτιστοποιήσεων για καλύτερους χρόνους απόκρισης είναι επιθυμητή.
- Προσανατολισμός: Σε κάθε σημείο της περιήγησής του, ο χρήστης πρέπει να έχει στη διάθεσή του εμφανή σημάδια που υποδεικνύουν που βρίσκεται (θεματική ενότητα ή εφαρμογή, κατηγορία, λειτουργία, κ.λπ.) που μπορεί να πάει και τι μπορεί/ τι πρέπει να κάνει.
- Πλοήγηση περιεχομένων. Το σύστημα θα πρέπει να υποστηρίζει πολλαπλούς τρόπους αναζήτησης και εντοπισμού περιεχομένων (browse, search, advanced search, κλπ.).
- Αισθητική. Η κατασκευή προσβάσιμου περιεχομένου Ιστού, για παράδειγμα ιστοσελίδων που συμμορφώνονται με τις σχετικές Οδηγίες της Κοινοπραξίας του Παγκόσμιου Ιστού, απαιτεί κατάρτιση, συστηματική προσέγγιση, και αφοσίωση στις αρχές καθολικής πρόσβαση και ίσων ευκαιριών. Σε πολλές περιπτώσεις, παρόλη την αγνή πρόθεση των κατασκευαστών, για παράδειγμα στο βωμό της προσβασιμότητας, θυσιάζεται η αισθητική της ιστοσελίδας.

Συνολικά, η προσέγγιση που έχει επιλέξει η Ομάδα Έργου για τη σχεδίαση της διεπαφής του χρήστη, βασίζεται στις αρχές της καθολικής σχεδίασης (universal design), γνωστή και ως σχεδίαση για όλους (design for all).

Η **σχεδίαση για όλους** «αφορά στην προσέγγιση σχεδίασης προϊόντων, περιβαλλόντων, προγραμμάτων και υπηρεσιών που μπορούν να χρησιμοποιηθούν από όλους τους ανθρώπους, στη μεγαλύτερη δυνατή έκταση, χωρίς ανάγκη προσαρμογής ή εξειδικευμένων λύσεων. Ο όρος «καθολική σχεδίαση» δεναποκλείει τις υποβοηθητικές συσκευές για συγκεκριμένες ομάδες ατόμων με αναπηρίες, όπου αυτό απαιτείται», (άρθρο 2 «Ορισμοί» της Διεθνούς Σύμβασης για τα Δικαιώματα των Ατόμων με Αναπηρία).

Η επιλογή αυτή είναι σημαντική, διότι η σχεδίαση για όλους λαμβάνει υπόψη την «προσβασιμότητα για όλους» από τον αρχικό σχεδιασμό, διασφαλίζοντας ότι ο

τελικός σχεδιασμός της διεπαφής δεν περιλαμβάνει σχεδιαστικές επιλογές οι οποίες θα δημιουργήσουν εμπόδια σε κάποιες κατηγορίες χρηστών, ενώ παράλληλα περιλαμβάνει τις απαραίτητες προβλέψεις - προσαρμογές που θα την καθιστούν προσβάσιμη και εύχρηστη και για ειδικές περιπτώσεις χρηστών και συνθηκών χρήσης (π.χ., μέσω της ενσωμάτωσης διαφορετικών τρόποι εκπλήρωσης της παρεχόμενης λειτουργικότητας).

Σε αυτή την κατεύθυνση, για τη σχεδίαση της διεπαφής του χρήστη του ΟΠΣ, η Ομάδα Έργου βασίστηκε στην εμπειρία της στη χρήση συγκεκριμένων, εδραιωμένων και εν ισχύ, διεθνών οδηγιών και προτύπων, όπως:

- A.** Στις διεθνείς οδηγίες **W3C/WCAG 2.0 (2008) Web Content Accessibility Guidelines, version 2.0⁵**, οι οποίες είναι διαθέσιμες και ως ανοικτό⁶ διεθνές πρότυπο **ISO/IEC 40500 (2012) W3C Web Content Accessibility Guidelines (WCAG) 2.0**.
- B.** Στις διεθνείς οδηγίες **W3C/ATAG 2.0 (2015) Authoring Tool Accessibility Guidelines, version 2.0⁷**, οι οποίες παρέχουν κατευθυντήριες γραμμές για τον σχεδιασμό εργαλείων δημιουργίας περιεχομένου ιστού που (α) να είναι και πιο προσιτά στους δημιουργούς με αναπηρίες (Μέρος Α) και (β) να έχουν σχεδιαστεί ώστε να επιτρέπουν, να υποστηρίζουν και να προωθούν την παραγωγή πιο προσβάσιμου Περιεχομένου Ιστού από όλους τους δημιουργούς (Μέρος Β).
- Γ.** Στις διεθνείς οδηγίες **W3C/MWBP 1.0 (2008) Mobile Web Best Practices, version 1.0⁸**, οι οποίες παρέχουν κατευθυντήριες γραμμές για το σχεδιασμό Περιεχομένου Ιστού κατάλληλου για πρόσβαση μέσω κινητών συσκευών.
- Δ.** Στο διεθνές πρότυπο **ISO 9241 Ergonomics of Human System Interaction⁹** και ειδικότερα:
 - Το Μέρος 20 (2008), που αφορά σε γενικές οδηγίες προσβασιμότητας για εξοπλισμό και υπηρεσίες πληροφορικής και επικοινωνιών.

5 Βλ. <https://www.w3.org/TR/WCAG20/>

6 Μεταφόρτωσή από: http://standards.iso.org/ittf/PubliclyAvailableStandards/c058625_ISO_IEC_40500_2012.zip (Αγγλικά)

7 Βλ. <https://www.w3.org/TR/ATAG20/>

8 Βλ. <https://www.w3.org/TR/mobile-bp/>

9 Οι εκδότες του προτύπου αναγνωρίζουν ότι η εφαρμογή του ISO 9241 (σειρά 100 που αφορά στη διεπαφή), αναμένεται να βελτιώσει τη συνολική ποιότητα της διεπαφής, αλλά αυτό το πρότυπο (όπως οποιοδήποτε άλλο πρότυπο) δεν εγγυάται την ποιότητα της διεπαφής. Και αυτό διότι, η ποιότητα εξαρτάται από συγκεκριμένα κριτήρια ευχρησίας όπως καθορίζονται από τον χρήστη της διεπαφής, η οποία μπορεί να περιλαμβάνει προδιαγραφές που βασίζονται σε αυτό το πρότυπο.

- Το Μέρος 171 (2008¹⁰), που αφορά σε οδηγίες για την προσβασιμότητα λογισμικού για άτομα με το ευρύτερο φάσμα σωματικών, αισθητηριακών και γνωστικών ικανοτήτων, συμπεριλαμβανομένων εκείνων που είναι προσωρινά με αναπηρία, και των ηλικιωμένων.
- Το Μέρος 110 (2006¹¹), που αφορά αρχές διαλόγου.
- Το Μέρος 112 (2017), που αφορά στην παρουσίαση πληροφορίας.
- Το Μέρος 151 (2008), που αφορά σε οδηγίες διεπαφές Ιστού.
- Το Μέρος 143 (2012¹²), που αφορά σε Φόρμες (Συμπλήρωσης).
- Το Μέρος 161 (2016), που αφορά σε οπτικά στοιχεία της διεπαφής του χρήστη.
- Το Μέρος 11 (1998¹³), που αφορά σε οδηγίες για την ευχρηστία.
- Το Μέρος 13 (1998), που αφορά στη βοήθεια / καθοδήγηση του χρήστη, με έμφαση στην υποστήριξη / βοήθεια για την αντιμετώπιση των σφαλμάτων και για χρήστες διαφόρων επιπέδων ικανοτήτων και γνώσεων.
- Το Μέρος 14 (1997), που αφορά ειδικότερα τη χρήση μενού στους διαλόγους χρήστη -συστήματος.
- Το Μέρος 16 (1999), που αφορά ειδικότερα σε διαλόγους άμεσου χειρισμού¹⁴.

Με βάση τα παραπάνω, η σχεδίαση της διεπαφής του χρήστη έγινε αρχικά μέσω της παραγωγής διαφόρων *σκελετών* (αγγλ. *wireframes*) και προσχεδίων (αγγλ. *sketches*), ως πρότυπα χαμηλής πιστότητας σε χαρτί και MS PowerPoint, τα οποία σταδιακά εξέλιξε σε σχέδια υψηλής πιστότητας σε Adobe Photoshop και Adobe XD.

10 Αυτό το πρότυπο αντικατέστησε το ISO/TS 16071:2003.

11 Αυτό το πρότυπο αναθεωρήθηκε για τελευταία φορά και επιβεβαιώθηκε το 2009. Επομένως, αυτή η έκδοση παραμένει ενεργή. Αυτό το πρότυπο είναι σε ισχύ από το 2006, αφού αντικατέστησε τα παλαιότερα και πλέον αποσυρμένα ISO 9241-10:1996 και ISO 9241-10:1994. Αναμένεται η αντικατάστασή του από το ISO/AWI 9241-110.

12 Αυτό το πρότυπο αντικατέστησε το ISO 9241-17:1998.

13 Αυτό το πρότυπο αναθεωρήθηκε για τελευταία φορά και επιβεβαιώθηκε το 2008. Επομένως, αυτή η έκδοση παραμένει ενεργή. Αυτό το πρότυπο είναι σε ισχύ από το 1998, αφού αντικατέστησε το παλαιότερο και πλέον αποσυρμένο ISO 9241, Μέρος 11 (1994β)

14 Π.χ., Σε γραφικές διεπαφές του χρήστη που βασίζονται στο μοντέλο WIMP (windows-icons-pointers-menus).

3.5.3 Σχεδίαση λογισμικού

Ο στόχος της σχεδίασης λογισμικού είναι η δημιουργία λογισμικού με συγκεκριμένες εσωτερικές ιδιότητες ποιότητας (αγγλ. *internal quality factors*), όπως:

- Ορθότητα (correctness)
- Στιβαρότητα (robustness)
- Επεκτασιμότητα (extendibility)
- Επαναχρησιμοποίηση (reusability)
- Συμβατότητα (compatibility)
- Αποδοτικότητα (efficiency)
- Μεταφεριμότητα (portability)
- Ευχρηστία (ease of use)
- Εύρος λειτουργιών (functionality)
- Παράδοση στο καθορισμένο χρονικό πλαίσιο (timeliness)
- Παράδοση στο καθορισμένο κόστος
- Ελεγχιμότητα (verifiability)
- Ασφάλεια
- Επιδιορθωσιμότητα (repairability)

Η τμηματικότητα (modularity) είναι βασική εσωτερική ιδιότητα που συνεισφέρει σε πολλά από τα παραπάνω. Η προτεινόμενη μέθοδος σχεδίασης είναι η *αντικειμενοστραφής σχεδίαση* (object-oriented design) η οποία αποβλέπει στην αύξηση της τμηματικότητας με βάση τις εξής αρχές:

- Σύνθεση (composability)
- Τμηματική κατανόηση (modular understandability)
- Συνέχεια (continuity)
- Προσταστατευόμενα τμήματα (modular protection)

Όπως αναφέρθηκε παραπάνω στην παράγραφο για την «ανάλυση απαιτήσεων», για την αποτύπωση του σχεδιασμού (μοντελοποίηση) η Ομάδα Έργου βασίζεται σε σύγχρονες διαγραμματικές τεχνικές στην γλώσσα UML οι οποίες χωρίζονται σε δύο βασικές κατηγορίες:

- Διαγραμματικές τεχνικές που χρησιμοποιούνται για την γενική περιγραφή του λογισμικού.
- Διαγραμματικές τεχνικές που χρησιμοποιούνται για τη λεπτομερειακή περιγραφή του λογισμικού.

4 Προσδιορισμός απαιτήσεων

4.1 Εισαγωγή

Ο προσδιορισμός των «απαιτήσεων» ενός συστήματος αφορά στον ακριβή προσδιορισμό του τι πρέπει να σχεδιαστεί και υλοποιηθεί και με τι ποιοτικά χαρακτηριστικά ώστε να ανταποκρίνεται ικανοποιητικά στις «επιχειρησιακές απαιτήσεις» του Φορέα Λειτουργίας και στις «απαιτήσεις χρηστών», για όλες τις κατηγορίες χρηστών του συστήματος.

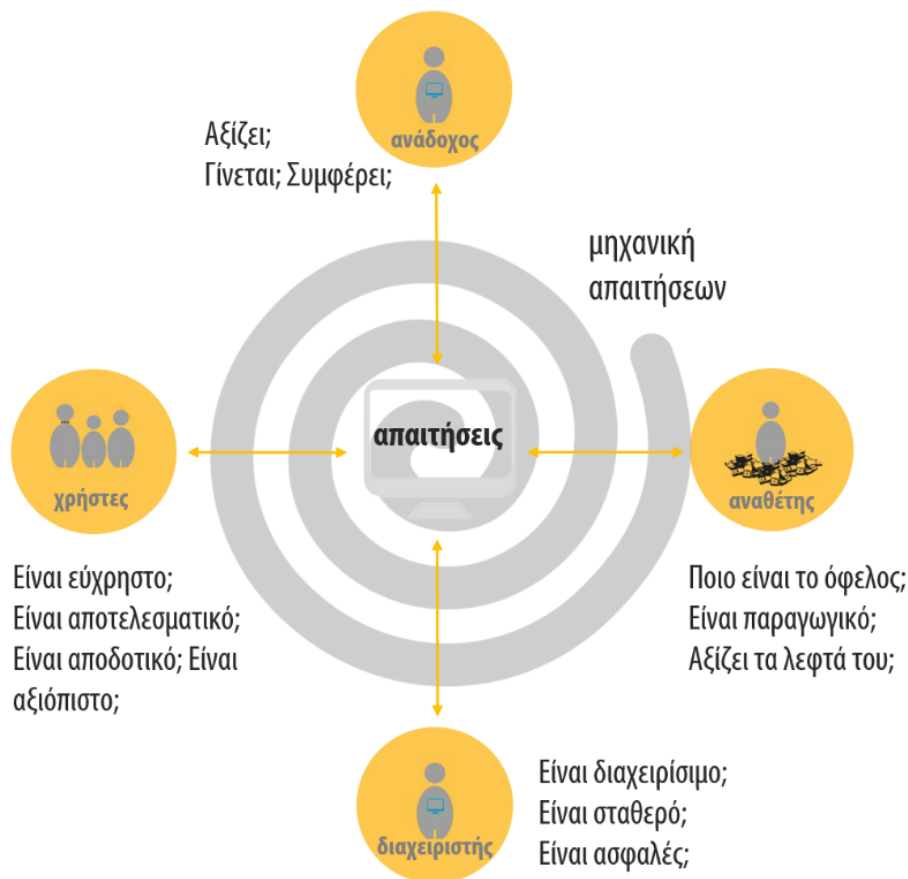
Η διατύπωση των απαιτήσεων πρέπει να είναι: πλήρης, συνεπής, ανιχνεύσιμη, ξεκάθαρη, και επαληθεύσιμη και να καλύπτει:

- «επιχειρησιακές» απαιτήσεις
- απαιτήσεις χρηστών / χειριστών / διαχειριστών
- λειτουργικές απαιτήσεις
- τεχνολογικές απαιτήσεις
- απαιτήσεις ποιότητας
- απαιτήσεις πόρων
- απαιτήσεις ελέγχου ποιότητας
- απαιτήσεις διασφάλισης ποιότητας
- απαιτήσεις τεκμηρίωσης και εκπαίδευσης
- απαιτήσεις εγκατάστασης
- απαιτήσεις συντήρησης

Αν και ως διαδικασία συνιστά το πρώτο στάδιο των εργασιών στο πλαίσιο της Ανθρωποκεντρικής Προσέγγισης εντούτοις είναι σαφές ότι το «πλαίσιο χρήσης» και οι «απαιτήσεις», γενικά, αλλά και ειδικότερα στην περίπτωση του Πληροφοριακού Συστήματος:

- Δεν είναι «στατικά», αλλά αναμένεται να παρουσιάσουν μεταβολές με την πάροδο του χρόνου. Οπότε χρειάζεται συνεχής επαλήθευση των τρεχουσών συνθηκών, και τυχόν αλλαγές να εξετάζονται ως προς τις επιπτώσεις που έχουν στα ζητούμενα ποιοτικά χαρακτηριστικά του συστήματος.
- Δεν αφορούν μόνο στη φάση της σχεδίασης, δηλαδή δεν αποτελούν εισροή μόνο για την οριστικοποίηση του σχεδιασμού (ώστε να μπορούν να αποσυρθούν μετά από τον σχεδιασμό), αλλά και μετέπειτα αποτελούν τη βάση αναφοράς για τον ποιοτικό έλεγχο του τελικού συστήματος και τον έλεγχο συμμόρφωσης με τις απαιτήσεις, ως προϋπόθεση για την οριστική παραλαβή από τον «πελάτη».

Στο παρακάτω σχήμα αποτυπώνονται ενδεικτικά οι προκλήσεις σε σχέση με τη διαχείριση των απαιτήσεων των επιμέρους εμπλεκόμενων (stakeholders), κατά τη διάρκεια του κύκλου ανάπτυξης ενός συστήματος, ώστε να διασφαλίζεται η αποδοχή του συστήματος από όλους, ως βασικό κριτήριο για τη συνολική επιτυχία του.



Εικόνα 5. Οι προκλήσεις για τη μηχανική των απαιτήσεων κατά την ανάπτυξη ενός συστήματος

Συνεπώς, η ανάλυση των απαιτήσεων, ειδικά για ένα σύστημα όπως το ΟΠΣ, είναι μια δυναμική, επαναληπτική, και συνεργατική διαδικασία η οποία απαιτεί από πλευράς αναδόχου, συστηματική προσέγγιση **μηχανικής απαιτήσεων** (αγγλ. *requirements engineering*) για τη διαχείριση των παραπάνω ζητημάτων, και η οποία στην περίπτωση αυτή έχει ξεκινήσει από πολύ νωρίς στο Έργο και θα ολοκληρωθεί με τη λήξη του.

Στο πλαίσιο αυτό, η Ομάδα Έργου έχει επιλέξει να βασιστεί σε άμεσες και αποτελεσματικές τεχνικές μηχανικής απαιτήσεων (focus groups, scenario building, rapid prototyping, κ.λπ.), ώστε καταρχήν να καθορισθεί συστηματικά, αναλυτικά και τεκμηριωμένα τι ακριβώς πρέπει να σχεδιαστεί. Η Ομάδα Έργου αρχικά επιχείρησε να εμβαθύνει στο «τι πρέπει να κάνει το σύστημα» (και όχι στο «πώς πρέπει να το κάνει», το οποίο αφορά την επόμενη φάση, δηλαδή αυτή

του σχεδιασμού¹⁵), και να προσδιορίσει πιθανά σενάρια μελλοντικών καταστάσεων και απαραίτητων λειτουργιών και συμπεριφορών από πλευράς συστήματος.

Το αποτέλεσμα αυτών των διεργασιών (βλ. κεφάλαιο 0) αποτέλεσε τη βάση του σχεδιασμού που παρουσιάζεται στα υπόλοιπα κεφάλαια της παρούσας αναφοράς, ενώ αργότερα στο Έργο θα λειτουργήσει ως βασικό στοιχείο για τον ποιοτικό έλεγχο του τελικού συστήματος.

Σημαντικό τμήμα της ανάλυσης των απαιτήσεων, είναι και η παρουσίαση και μοντελοποίηση των απαιτήσεων, δηλαδή η οργάνωση και διατύπωσή τους σε διάφορες μορφές κατάλληλες για τις μετέπειτα χρήσεις τους.

4.2 Απαιτήσεις

Οι γενικές αρχές που θα διέπουν το Πληροφοριακό Σύστημα σε λειτουργικό και τεχνολογικό επίπεδο είναι:

- Σύστημα «ανοικτής» αρχιτεκτονικής (open architecture), δηλαδή υποχρεωτική χρήση ανοικτών προτύπων που θα διασφαλίζουν:
 - την ομαλή λειτουργία και συνεργασία μεταξύ του συνόλου των επιμέρους υποσυστημάτων και των υπηρεσιών τους
 - την εύκολη επεκτασιμότητα των υποσυστημάτων χωρίς αλλαγές στη δομή και αρχιτεκτονική τους.

Οι εφαρμογές θα πρέπει να είναι κατάλληλα σχεδιασμένες ώστε να παρέχουν τη δυνατότητα εύκολης επικοινωνίας, διασύνδεσης ή και ολοκλήρωσης με τρίτες εφαρμογές ή / και υποσυστήματα.

Γι' αυτό το λόγο θα πρέπει να παρέχουν κατ' ελάχιστον τα ακόλουθα:

- Τεκμηριωμένα API (Application Programming Interface) τα οποία να επιτρέπουν την ολοκλήρωση/ διασύνδεση με τρίτες εφαρμογές, όπου αυτό είναι απαραίτητο.
- Δυνατότητα διασύνδεσης / επικοινωνίας με τρίτες εφαρμογές βάσει διεθνών standards (όπως: REST, JSON, XML, SOAP, UDDI κλπ.),
- Αρθρωτή (modular) αρχιτεκτονική του συστήματος, ώστε να επιτρέπονται μελλοντικές επεκτάσεις και αντικαταστάσεις, ενσωματώσεις, αναβαθμίσεις ή αλλαγές διακριτών τμημάτων λογισμικού ή εξοπλισμού

¹⁵ Δηλ. Η Ανάλυση δίνει έμφαση στη διερεύνηση του προβλήματος, περί τίνος πρόκειται και τι απαιτείται να κάνει ένα σύστημα ενώ ο Σχεδιασμός στη λογική της επίλυσης, στο πώς δηλαδή το σύστημα θα πληροί τις απαιτήσεις.

- Αρχιτεκτονική N-tier για την ευελιξία της κατανομής του κόστους και φορτίου μεταξύ κεντρικών συστημάτων και σταθμών εργασίας, για την αποδοτική εκμετάλλευση του δικτύου και την ευκολία στην επεκτασιμότητα, αλλά και στη συντήρησή του
- Χρήση γραφικού περιβάλλοντος λειτουργίας των χρηστών για την αποδοτική χρήση των εφαρμογών και την ευκολία εκμάθησής τους
- Διασφάλιση της πληρότητας, ποιότητας, ακεραιότητας και ασφάλειας των δεδομένων των εφαρμογών.
- Σχεδιασμός και υλοποίηση με βασική αρχή την οικονομία πόρων αλλά και τη βέλτιστη απόδοση των συστημάτων που θα προσφερθούν.

Ασφάλεια του συστήματος

Η ασφάλεια του συστήματος αποτελεί σημαντικό παράγοντα που λαμβάνεται υπόψη από τα πρώτα στάδια της ανάλυσης και της σχεδίασης του Πληροφοριακού Συστήματος και αφορά σε πολλαπλά επίπεδα. Παρακάτω αναλύονται οι πρακτικές και οι μέθοδοι που θα εφαρμοστούν στο έργο.

Χρήση SSL: Όλα τα υποσυστήματα θα εξυπηρετούνται απαραίτητως μέσω SSL (Secure Socket Layers) γνωστό και ως Ηλεκτρονικό Πιστοποιητικό, το πρωτόκολλο SSL δημιουργεί μια ασφαλή σύνδεση μεταξύ της εκάστοτε ιστοσελίδας (server) και του φυλλομετρητή (browser) του χρήστη. Τα SSL πιστοποιητικά εξασφαλίζουν την ασφαλή ανταλλαγή δεδομένων ανάμεσα στις δύο πλευρές, αποτρέποντας κακόβουλους χρήστες από την υποκλοπή δεδομένων. Επιπλέον η χρήση του SSL πλέον εξασφαλίζει καλύτερα αποτελέσματα στις μηχανές αναζήτησης. Τα πιστοποιητικά SSL που θα χρησιμοποιηθούν στο έργο θα βασίζονται στην ελεύθερη Αρχή Πιστοποίησης Let's Encrypt (letsencrypt.org) που χρησιμοποιεί το PKCS #1 SHA-256 με κρυπτογράφηση RSA Encryption ως αλγόριθμο υπογραφής του πιστοποιητικού ενώ θα γίνουν οι απαραίτητες ενέργειες σε επίπεδο server για την αυτοματοποιημένη ανανέωση τους χωρίς να απαιτούνται περαιτέρω ενέργειες από τους διαχειριστές. Αξίζει να σημειωθεί ότι όλες οι ιστοσελίδες του ΟΠΣ και της προσβάσιμης Πύλης του έργου θα εξυπηρετούνται μονίμως μέσω SSL και όχι μόνο κατά την εγγραφή και τη σύνδεση (registration, login). Με το τρόπο εξασφαλίζεται η μέγιστη δυνατή ασφάλεια. Επιπλέον θα υπάρχουν αυτοματοποιημένοι μηχανισμοί που θα διασφαλίζουν ότι όλοι οι πόροι (όπως φωτογραφίες, βίντεο, JavaScript, CSS, κ.α.) που χρησιμοποιούνται στις ιστοσελίδες βρίσκονται σε ασφαλείς διευθύνσεις SSL (https).

Ασφάλεια στην επικοινωνία μέσω REST: Στις κλήσεις που απαιτείται ταυτοποίηση χρήστη, δημιουργείται ένα αλφαριθμητικό το οποίο αποτελείται από το password του χρήστη, την τοπική IP, το περιεχόμενο του αιτήματος και

την ώρα δημιουργίας του αιτήματος και υπολογίζεται ο κατατεμαχισμός (hash) του αλφαριθμητικού με τον αλγόριθμο SHA-256. Μέσω των headers του αιτήματος αποστέλλεται το hash, το username και η ώρα (Unix Timestamp) που έγινε η κλήση. Ο server χρησιμοποιεί το username για να ανασύρει το password του χρήστη από την ενιαία βάση χρηστών του συστήματος και το χρησιμοποιεί μαζί με την ώρα που διαβάζει από το header, το περιεχόμενο του αιτήματος και την IP του αιτήματος για να υπολογίσει το αναμενόμενο hash. Συγκρίνει το hash αυτό με το hash που έχει αποσταλεί για να βεβαιωθεί ότι όντως ο χρήστης είναι αυτός που ισχυρίζεται και ότι το μήνυμα δεν έχει πειραχτεί. Επιπλέον, συγκρίνει την ώρα που βρίσκεται στα headers με την δικιά του ώρα και σε περίπτωση που το αίτημα είναι παλιότερο των 30 δευτερολέπτων τότε απορρίπτεται ώστε να ελαχιστοποιηθεί η περίπτωση ενός replay attack. Η διαδικασία αυτή είναι ένα επιπλέον στρώμα ασφάλειας πάνω από το SSL μέσω του οποίου θα πραγματοποιούνται όλες οι κλήσεις των υπηρεσιών REST.

Χρήση ομάδων χρηστών, ρόλων και ανάθεση δικαιωμάτων: Η ανάθεση των δικαιωμάτων των χρηστών (σε επίπεδο υπηρεσιών, βάσεων δεδομένων, διαχείρισης του backend, κ.λπ.) θα γίνεται στις ομάδες (groups) και κάθε χρήστης θα γίνεται μέλος μια ομάδας. Αυτό διασφαλίζει την εύκολη αλλαγή ρόλων των χρηστών αλλά και την ευκολότερη επικοινωνία (notifications) και καταγραφή (logging).

Συνθηματικά (Passwords): Θα οριστούν κανόνες (π.χ. Ελάχιστο μήκος 10 χαρακτήρες) ενώ για τις ομάδες διαχείρισης θα είναι απαραίτητη η αλλαγή του συνθηματικού ανά τακτά χρονικά διαστήματα. Μετά το registration των χρηστών θα πρέπει να γίνεται επιβεβαίωση μέσω email ώστε να διασφαλιστεί ότι ο χρήστης είναι πραγματικός.

Two-factor authentication: Ο έλεγχος ταυτότητας δύο παραγόντων, μέσω κινητού ή και PC, παρέχει ένα επιπλέον επίπεδο ασφαλείας πέραν του κλασσικού τρόπου σύνδεσης με το όνομα χρήστη και συνθηματικού για την επιβεβαίωση της ταυτότητας του χρήστη κατά την είσοδο στον λογαριασμό του. Ο τρόπος αυτός χρησιμοποιείται σε συνδυασμό με ένα κινητό, ένα tablet ή κάποια άλλη συσκευή παραγωγής ηλεκτρονικών κωδικών, σαν αυτόν που χρησιμοποιούν συνήθως οι τράπεζες. Ο τρόπος αυτός ουσιαστικά προστατεύει από τυχόν κλοπή του username και του password αφού δεν αρκούν πλέον μόνο αυτά για να εισέλθει κάποιος στον λογαριασμό του. Ακόμα κι αν κλαπούν τα στοιχεία πρόσβασης, η σύνδεση στο λογαριασμό του χρήστη, χωρίς τη χρήση του επιπλέον κωδικού two-factor authentication είναι αδύνατη.

Καταγραφή των ενεργειών: Κάθε ενέργεια που θα γίνεται στο ΟΠΣ, θα καταγράφεται λεπτομερώς σε log files Με το τρόπο αυτό θα μπορεί να γίνεται

ανάλυση των δεδομένων ώστε να μειώνονται οι προσπάθειες διαπέρασης του συστήματος από κακόβουλους χρήστες.

Εφαρμογή καλών πρακτικών ασφάλειας: Σε επίπεδο server θα γίνει χρήση των fail2ban για την αποκοπή προσπαθειών διάσπασης των συνθηματικών, no root password ώστε να απαιτείται πιστοποιητικό για τη σύνδεση (SSH), amavis για τον έλεγχο ιών και trojans, χρήση firewall σε επίπεδο λειτουργικού συστήματος. Σε επίπεδο διασφάλισης δεδομένων που βρίσκονται σε αποθήκευση (data-at-rest) θα ενεργοποιηθούν με την παράδοση του έργου τεχνικές κρυπτογράφησης (αν το επιθυμεί η Αναθέτουσα καθώς η κρυπτογράφηση μειώνει ελαφρώς την ταχύτητα των εφαρμογών) στη μεριά του server όπου και θα απαιτείται η χρήση ενός κλειδιού KMS για την αποκρυπτογράφηση τους. Σε επίπεδο frontend, θα χρησιμοποιούνται σε όλες τις φόρμες, μέσω των οποίων θα μεταφέρονται δεδομένα, τα session tokens ώστε να αποφεύγονται επιθέσεις τύπου Cross-Site Request Forgery (CSRF).

4.3 Εγγραφή και αυθεντικοποίηση χρηστών / φορέων

Η Πλατφόρμα υποστηρίζει τόσο τους ανώνυμους χρήστες/επισκέπτες οι οποίοι έχουν πρόσβαση σε όλα τα δημόσια δεδομένα της Πλατφόρμας, όσο και στους εγγεγραμμένους χρήστες οι οποίοι μπορούν να έχουν μεγαλύτερη αλληλεπίδραση με το γενικότερο Πληροφοριακό Σύστημα. Έχει γίνει επίσης ανάλυση του τρόπου που λειτουργεί ο μηχανισμός single sign on μέσω OAuth2.0 (που είναι το de-facto standard) και επιτρέπει στους χρήστες την είσοδο τους στα διάφορα υποσυστήματα και λειτουργίες της Πλατφόρμας χωρίς να χρειάζεται εγγραφή για κάθε υποσύστημα ξεχωριστά.

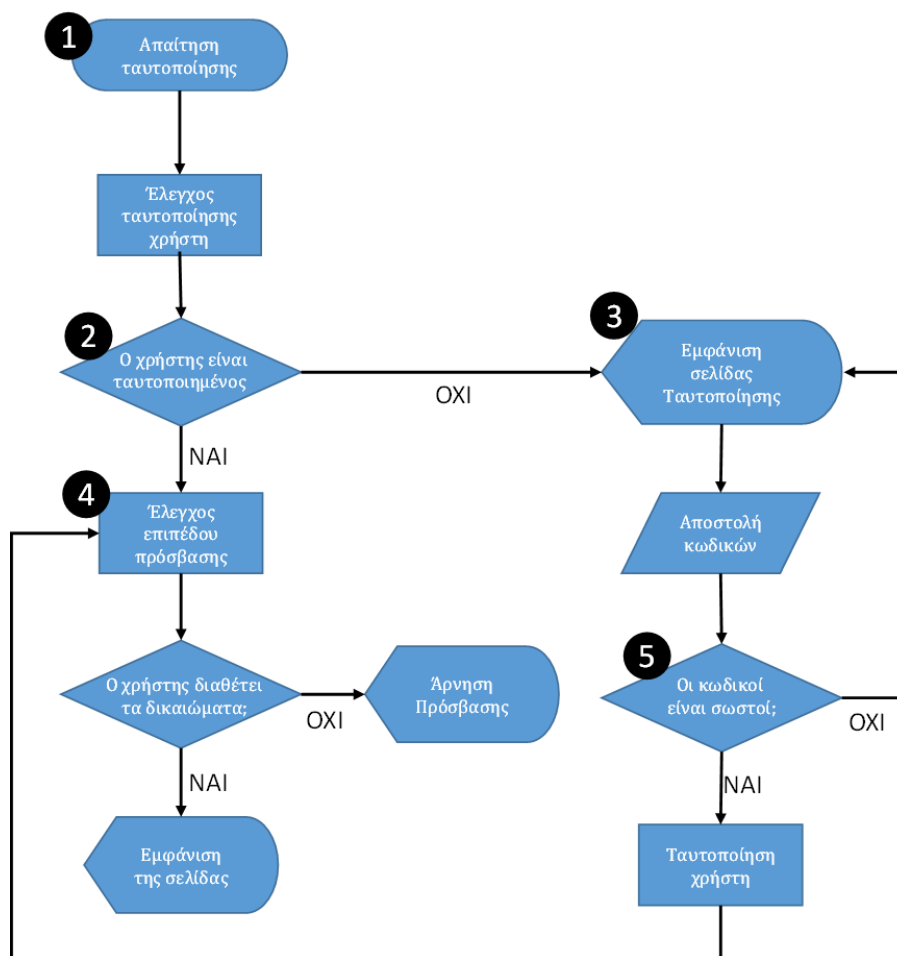
Οι εγγεγραμμένοι χρήστες την πρώτη φορά θα πρέπει να περνούν από μια διαδικασία αρχικής καταγραφής (registration) έτσι ώστε να αποκτήσουν λογαριασμό στο σύστημα και να μπορούν να συνδεθούν στις ειδικές υπηρεσίες – εφαρμογές. Με τον τρόπο αυτό είναι δυνατή η δημιουργία προσωπικού χώρου δεδομένων ο οποίος θα χρησιμοποιηθεί για την αποθήκευση δεδομένων που σχετίζονται με τις παρεχόμενες υπηρεσίες.

Η διαδικασία ταυτοποίησης θα ακολουθεί τα βήματα που παρουσιάζονται στη συνέχεια:

- Βήμα 1^ο: Έλεγχος απαιτήσεων εξουσιοδότησης της συγκεκριμένης σελίδας που ζητήθηκε.
- Βήμα 2^ο: Έλεγχος της κατάστασης ταυτοποίησης του χρήστη.
- Βήμα 3^ο: Αν ο χρήστης δεν έχει ταυτοποιηθεί (ανώνυμος χρήστης) παρουσίαση της "σελίδας ταυτοποίησης" (Login) (Συνέχεια στο Βήμα 5).

- Βήμα 4^ο: Αν ο χρήστης έχει ταυτοποιηθεί αλλά τα δικαιώματα του δεν του επιτρέπουν την πρόσβαση στη συγκεκριμένη σελίδα παρουσίαση της "σελίδας άρνησης πρόσβασης" (Access denied). Αν ο χρήστης έχει τα κατάλληλα δικαιώματα, παρουσίαση της σελίδας. Τέλος διαδικασίας
- Βήμα 5^ο: Έλεγχος των κωδικών πρόσβασης. Αν οι κωδικοί είναι λανθασμένοι παρουσίαση της σελίδας ταυτοποίησης με το μήνυμα "λανθασμένος κωδικός" (login incorrect). Αν οι κωδικοί είναι σωστοί ταυτοποίηση χρήστη και επιστροφή στο 4^ο βήμα.

Η διαδικασία ταυτοποίησης του χρήστη παρουσιάζεται στην παρακάτω εικόνα.



Εικόνα 6. Διάγραμμα ροής της διαδικασίας ταυτοποίησης χρηστών

Προφανώς, το ότι κάποιος χρήστης είναι εγγεγραμμένος δε σημαίνει ότι μπορεί να προσπελάζει όλα τα υποσυστήματα, καθώς υπάρχουν περιορισμοί και κανόνες ανάλογα με τις ομάδες χρηστών που ανήκει ο κάθε χρήστης (κάτι που ελέγχει και ορίζει ο υπερδιαχειριστής).

Η εγγραφή των χρηστών γίνεται μέσω συνδέσμου πρώτης προβολής στην αρχική σελίδα, ("εγγραφή") ο οποίος οδηγεί στη φόρμα της εγγραφής. Μετά την

επιτυχημένη εγγραφή, οι χρήστες θα λαμβάνουν email που θα περιέχει ένα σύνδεσμο που ενεργοποιεί το λογαριασμό τους. Αν ο Φορέας το επιθυμεί μπορεί να παρεμβάλλεται και ενδιάμεσος έλεγχος από τους διαχειριστές για την ενεργοποίηση του λογαριασμού (moderation). Από τη στιγμή που κάποιος χρήστης είναι εγγεγραμμένος, μπορεί να κάνει εισαγωγή (login) και να δημιουργήσει το «κατάστημα» του. Υπάρχει κατάλληλος μηχανισμός μέσω cookies και sessions (secure) που μπορεί να κρατάει συνδεδεμένο τον χρήστη ακόμα και αν κλείσει τον browser (όπως δηλαδή γίνεται με τις υπηρεσίες της Google).

Οι κωδικοί ασφαλείας και ότι άλλο στοιχείο αφορά στον χρήστη θα μπορεί να επεξεργαστεί μέσω της προσωπικής σελίδας που έχει ο κάθε χρήστης ("My Account"). Μέσω της σελίδας του λογαριασμού του χρήστη, όπως ορίζουν οι πιο πρόσφατες καλές πρακτικές, θα υπάρχει η δυνατότητα ακόμα και διαγραφής του λογαριασμού (προφανώς θα υπάρχει διπλή δικλείδα ασφαλείας ώστε να μην γίνει διαγραφή κατά λάθος).

5 Σχεδιασμός της εμπειρίας του χρήστη

5.1 Περιβάλλον της Πλατφόρμας

Ως γενική αρχή, το Περιβάλλον της Πλατφόρμας με πληροφορίες τουριστικού ενδιαφέροντος και προσβασιμότητας για τουρίστες και επισκέπτες της διασυνοριακής περιοχής του προγράμματος, με απώτερο σκοπό την προβολή, την προώθηση και την ανάπτυξη του προσβάσιμου τουρισμού κληρονομιάς στη διασυνοριακή περιοχή θα έχει μια πολύ απλή λογική πλοήγησης θα είναι προσβάσιμο σύμφωνα με τις οδηγίες WCAG 2.0 σε επίπεδο "A" και με τις οδηγίες ATAG 2.0, όπως προβλέπεται από το W3C για τη σχεδίαση Web περιβαλλόντων / εφαρμογών δημιουργίας περιεχομένου ιστού, ώστε:

- να είναι πιο προσβάσιμα σε συγγραφείς / διαχειριστές με αναπηρία (Μέρος A) και
- να επιτρέπουν, να υποστηρίζουν και να διασφαλίζουν την παραγωγή πιο προσβάσιμου περιεχομένου Ιστού (Μέρος B).

Επομένως, ως προς το Μέρος (A), απαιτείται η διεπαφή της Πλατφόρμας να είναι προσβάσιμη (WCAG 2.0 επίπεδο A, και ATAG 2.0 - PART A), και ως προς το Μέρος (B), απαιτείται να προσφέρονται στους χρήστες εργαλεία και διαδικασίες που διασφαλίζουν ότι το περιεχόμενο που δημιουργείται είναι προσβάσιμο.

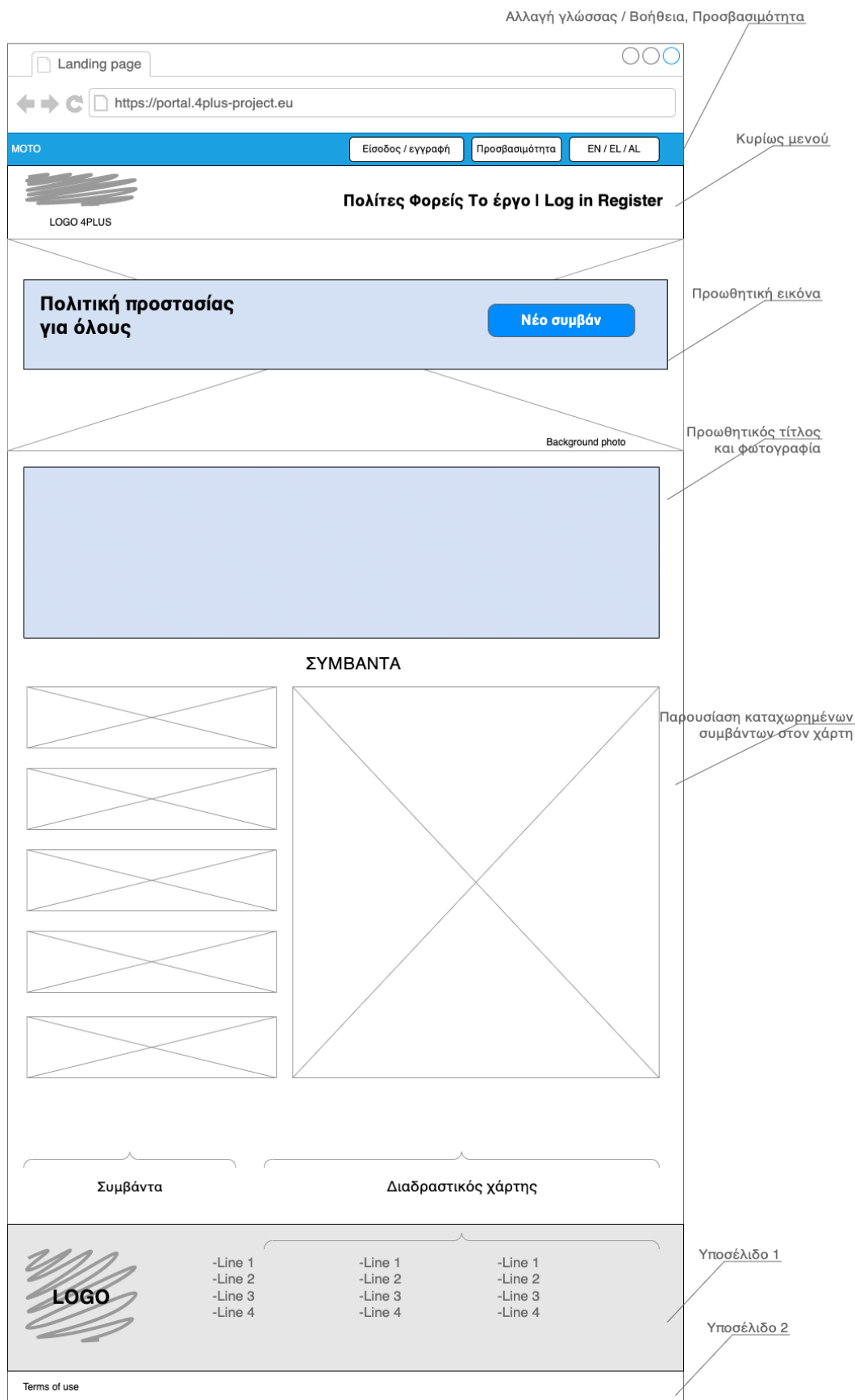
Επιπλέον, βασική πρόκληση για τον σχεδιασμό του Πληροφοριακού Συστήματος είναι να ακολουθεί ένα **μοντέλο πλοήγησης που να διαισθητικό**, δηλαδή να είναι κατανοητό και να ταιριάζει στον τρόπο σκέψης (conceptual model) των χρηστών, και όχι να βασίζεται για παράδειγμα στη λογική της αρχιτεκτονικής των επιμέρους υποσυστημάτων.

5.2 Σχεδίαση της Διεπαφής του χρήστη

Με βάση την παραπάνω βασική δομή και η φιλοσοφία της αλληλεπίδρασης για τους χρήστες, καθορίστηκαν τα χαρακτηριστικά που πρέπει να έχει η διεπαφή του χρήστη, ξεκινώντας από της σχεδίαση της «αρχικής σελίδας» στην οποία θα «προσγειώνονται» (landing page) οι χρήστες που το επισκέπτονται και η οποία θα είναι πλήρως προσαρμοσμένη στο προφίλ του και τις εργασίες στις οποίες έχει δικαίωμα πρόσβασης ο κάθε χρήστης ξεχωριστά.

Στις παρακάτω εικόνες, παρουσιάζεται ως προσχέδιο (mockup) η δομή της διεπαφής και τα βασικά χαρακτηριστικά της Πλατφόρμας.

Το Έργο συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση και από εθνικούς πόρους των κρατών που συμμετέχουν στο Πρόγραμμα



Εικόνα 7. Προσχέδιο της αρχικής σελίδας όπως θα παρουσιάζεται στους «Επισκέπτες»

5.3 Χρήστες

A. Εξωτερικοί Χρήστες (δηλ. χωρίς εγγραφή, «Επισκέπτες»):

- Πολίτες, το οποίο θα έχει πρόσβαση στην πρώτη περιοχή (βλ. παραπάνω).
- Φορείς από τη διασυνοριακή περιοχή. Για τους φορείς θα υπάρχει δυνατότητα δημιουργίας λογαριασμού. Σε διαφορετική περίπτωση, όταν δηλ. δε θα έχει δημιουργηθεί λογαριασμός, η πρόσβασή τους θα περιορίζεται στα δημόσια στοιχεία.

B. Εσωτερικοί Χρήστες (δηλ. με εγγραφή):

- Διαχειριστές Συστήματος. Οι διαχειριστές θα είναι στελέχη ή συνεργάτες των εταίρων, και θα αναλάβουν μετά την οριστική παραλαβή του έργου τη λειτουργία και διαχείριση της Κεντρικής Βάσης Δεδομένων, των υποσυστημάτων της πλατφόρμας καθώς και των χρηστών.
- Διαχειριστές Περιεχομένου. Συνιστούν ένα ικανό και ενιαίο πυρήνα στελεχών/συνεργατών του φορέα λειτουργίας με εξειδικευμένη γνώση σε λειτουργικό, τεχνικό και επιχειρησιακό επίπεδο. Τα στελέχη αυτά σταδιακά θα καταστούν ικανά να αναλάβουν, μετά το πέρας του παρόντος Έργου, τη διαχείριση και λειτουργία των συστημάτων. Τα εκπαιδευμένα στελέχη του μηχανισμού θα έχουν πλήρη πρόσβαση στο σύστημα και θα πραγματοποιούν τις απαραίτητες αλλαγές/ενημερώσεις. Συνεπώς, θα προβλεφθεί διαδικασία ταυτοποίησης της κάθε ενημέρωσης.
- Φορείς από τη διασυνοριακή περιοχή. Για τους φορείς θα υπάρχει δυνατότητα δημιουργίας λογαριασμού, ώστε να λειτουργούν ως Διαχειριστές συγκεκριμένου/προβλεπόμενου είδους περιεχομένου.

5.4 Δομή της Διεπαφής



Εικόνα 8. Προσχέδιο της δομής (περιοχές – λειτουργίες) της Πύλης όπως θα παρουσιάζεται στους διάφορους τύπους χρηστών

5.5 Υλοποίηση της Διεπαφής

Η υλοποίηση της διεπαφής του Πληροφοριακού Συστήματος, εφαρμόζοντας βέλτιστες πρακτικές και μεθοδολογίες, εξασφαλίζει τη βέλτιστη εμπειρία αλληλεπίδρασης:

5.5.1 Ταχύτητα απόκρισης

Λαμβάνονται υπόψη οι παρακάτω ενδεικτικές σύγχρονες πρακτικές βελτιστοποίησης και καλού σχεδιασμού των web-based πληροφοριακών συστημάτων:

- Βελτιστοποίηση μεγέθους εικόνων (πχ με χρήση του tinypng)
- Ιεράρχηση του εμφανούς περιεχομένου μέσω εξάλειψης της JavaScript και του CSS από το render-blocking κομμάτι στο περιεχόμενο που βρίσκεται πιο ψηλά (τεχνική above-the-fold-content)
- Χρήση τεχνικών συμπίεσης του περιεχομένου
- Αξιοποίηση της προσωρινής αποθήκευσης του φυλλομετρητή
- Μείωση του μεγέθους του CSS
- Μείωση του μεγέθους της Javascript
- Μείωση του μεγέθους της HTML
- Μείωση του χρόνου απόκρισης του διακομιστή

5.5.2 Ανταποκρίσιμη διάταξη

Τα υποσυστήματα του ΟΠΣ σχεδιάζονται με βάση τον ανταποκρίσιμο σχεδιασμό που σημαίνει ότι η διάταξη των ιστοσελίδων προσαρμόζονται ανάλογα με το μέγεθος της οθόνης της συσκευής που τα προσπελάζει. Αυτό σημαίνει ότι χωρίς αλλαγή στο markup, με χρήση CSS media queries, η ιστοσελίδα προσαρμόζεται κατάλληλα. Για παράδειγμα το μενού επεκτείνεται όταν η προσπέλαση γίνεται από υπολογιστή αλλά συρρικνώνεται όταν η προσπέλαση γίνεται για παράδειγμα μέσω κινητού τηλεφώνου. Επιπλέον της διάταξης όμως, θα ενσωματωθούν μηχανισμοί προσαρμοζόμενου σχεδιασμού (πλέον του ευέλικτου) που σημαίνει ότι και το περιεχόμενο μπορεί να διαφέρει ανάλογα της συσκευής. Για παράδειγμα χρήση μικρότερους μεγέθους (σε Kb, δηλαδή διαφορετική εικόνα) φωτογραφιών όταν η προσπέλαση γίνεται από κινητά τηλέφωνα για να βελτιωθεί η ταχύτητα.

Εκτός του προσαρμοζόμενου και ευέλικτου σχεδιασμού ως απαραίτητων συστατικών του ανταποκρίσιμου σχεδιασμού και λαμβάνοντας υπόψη τις ιδιαίτερες απαιτήσεις των χρηστών της Πλατφόρμας, η προτεινόμενη λύση

εισάγει επιπλέον την έννοια του προσαρμοσμένου προφίλ χρηστών που δίνει έμφαση στις ρυθμίσεις προσβασιμότητας, εξατομικευμένα για κάθε χρήστη. Οι χρήστες θα μπορούν μέσω των ρυθμίσεων προσβασιμότητας να επιλέγουν τα χρώματα (πχ υψηλή αντίθεση για άτομα με προβλήματα όρασης), το πλάτος του περιεχομένου της σελίδας (πχ μέγιστο-μεσαίο-στενό για άτομα με γλαύκωμα), εμφάνιση με μονή σελιδοποίηση και εμφάνιση του ανεπτυγμένου μενού για άτομα με άνοια, μεγαλύτερα κενά στις γραμμές και αποφυγή συντομεύσεων για ηλικιωμένα άτομα, κλπ. Όλες οι ρυθμίσεις θα αποθηκεύονται σε cookies για τους επισκέπτες, ενώ οι εγγεγραμμένοι χρήστες θα έχουν το δικαίωμα αποθήκευσης των προτιμήσεων τους ώστε ακόμα και αν επισκεφθούν το Παρατηρητήριο από οποιοδήποτε υπολογιστή ή κινητό τηλέφωνο να μπορεί να γίνει η προσαρμογή αυτόματα.

5.5.3 Καλές πρακτικές

- Κατά τη διάρκεια καταχώρησης μέσω φόρμας, τα δεδομένα παραμένουν ακόμα και αν ο χρήστης κατά λάθος κλείσει τον browser ή ανανεώσει τη σελίδα. Ο μηχανισμός αυτός κάνει χρήση του local storage δυναμικά και βελτιώνει πολύ την εμπειρία χρήσης (user experience) των χρηστών. Επιπλέον, η καταχώρηση των δεδομένων γίνεται ασύγχρονα χωρίς να χρειάζεται να φορτωθεί εξ' αρχής το σύνολο της σελίδας.
- Όλα τα υποσυστήματα του Παρατηρητηρίου υποστηρίζουν Search Engine Friendly (SEF) URLs τα οποία αν θέλουν οι διαχειριστές μπορούν να αλλάζουν κατά τη διάρκεια εισαγωγής/επεξεργασίας του περιεχομένου εξασφαλίζοντας πλήρη ελευθερία. Αυτό επιτυγχάνεται μέσω εξελιγμένων τεχνικών Routing αλλά και των κατάλληλων ρυθμίσεων σε επίπεδο διακομιστών.
- Οι ιστοσελίδες ακολουθούν τις οδηγίες του WCAG2.0 σε επίπεδο AAA.
- Κατά την εισαγωγή δεδομένων γίνεται πάντα έλεγχος της ορθότητας τους τόσο στη μεριά του client μέσω JavaScript όσο και στη μεριά του server. Ο έλεγχος γίνεται όλους τους τύπους των δεδομένων ενώ οι κανόνες (πχ για τη μορφή της ημερομηνίας) μπορούν να οριστούν από τους υπερδιαχειριστές. Οι κανόνες επικύρωσης εφαρμόζονται επίσης και σε όλες τις φόρμες εισαγωγής δεδομένων ενώ υπάρχουν και μηχανισμοί ασφαλείας που δεν επιτρέπουν τη μαζική εισαγωγή μέσω bots στην οριακή περίπτωση που υπάρξει κακόβουλος εγγεγραμμένος χρήστης.
- Ενοποιημένος μηχανισμός εισόδου στα υποσυστήματα με μια εγγραφή. Η αυθεντικοποίηση γίνεται μέσω του υποσυστήματος διαχείρισης χρηστών και του OAuth2.0 server που χρησιμοποιείται.

- Όλα τα υποσυστήματα, υποστηρίζουν εξ' ορισμού τη κωδικοποίηση UTF-8.
- Υποστηρίζεται η αυτόματη online βοήθεια στους χρήστες με οδηγίες χρήσης διαθέσιμη και σε διαφορετικές προσβάσιμες μορφές.
- Εκτεταμένη χρήση ARIA. ARIA είναι μια μικρή ποσότητα σήμανσης (κώδικας HTML) που χρησιμοποιείται για το καθορισμό κανόνων ή λογικής για τους αναγνώστες οθόνης και άλλες βοηθητικές τεχνολογίες. Για παράδειγμα υπάρχει η δυνατότητα να ορίζεις ρόλους σε συγκεκριμένα html tags ώστε να αποκτούν σημασία και νόημα (πχ `<header role="banner">banner</header>`).
- Χρήση σχετικών (em, rem, vs, hs) και όχι απόλυτων μέτρων (px) στο CSS για καλύτερη προσαρμοστικότητα του περιεχομένου.
- Χρήση accesskeys (πχ με το πλήκτρο M να εμφανίζεται το μενού).
- Χρήση tabindex (ώστε να αποφεύγονται η περιττή χρήση του tab για τα άτομα που χρησιμοποιούν μόνο το πληκτρολόγιο).

6 Αρχιτεκτονική συστήματος και εφαρμογών

6.1 Γενικά

Το πληροφοριακό σύστημα και τα επιμέρους υποσυστήματα που το απαρτίζουν αφορά σε μία σύνθετη πλατφόρμα που συνδυάζει πολλαπλές τεχνολογίες, τόσο σε επίπεδο λογισμικού όσο και σε επίπεδο επιλογής κατάλληλων τεχνικών και βέλτιστων πρακτικών λύσεων.

Οι γενικές αρχές που διέπουν τον προτεινόμενο σχεδιασμό, απορρέουν από τις προσεγγίσεις και τεχνικές που παρουσιάστηκαν στα προηγούμενα κεφάλαια, με σαφή έμφαση στην διασφάλιση της ποιότητας, στην προσβασιμότητα των εφαρμογών, στην επεκτασιμότητα (extendability) των συστημάτων, και στην ασφάλεια (security). Στις παρακάτω ενότητες αναλύονται τα χαρακτηριστικά του προτεινόμενου σχεδιασμού των εφαρμογών του Έργου.

6.2 Ανοιχτό λογισμικό και τεχνολογίες

Η ομάδα υλοποίησης διαθέτει μακροχρόνια εμπειρία στην ανάπτυξη διαδικτυακών εφαρμογών που στηρίζονται εξ' ολοκλήρου σε ελεύθερο λογισμικό ανοιχτού κώδικα και στο Έργο όλα τα υποσυστήματα που θα απαρτίζουν την πλατφόρμα λογισμικού του θα βασίζονται σε αυτό. Η σωστή επιλογή και ο συνδυασμός των κατάλληλων εργαλείων ανοιχτού λογισμικού αποτελεί σύνθετη διαδικασία που διασφαλίζεται από την πολυετή ενασχόληση της ομάδας ανάπτυξης με αποκλειστικά τέτοιου είδους τεχνολογίες. Τα πλεονεκτήματα της χρήσης ανοιχτού λογισμικού είναι:

Ευελιξία. Ο ανοιχτός κώδικας επιτρέπει την τεχνολογική ευελιξία, προσφέροντας πολλαπλούς τρόπους για την επίλυση προβλημάτων και βοηθά την ομάδα ανάπτυξης να μην αντιμετωπίζει εμπόδια επειδή κάποια λύση ή τεχνολογική ανάγκη της δεν παρέχεται από κάποιον προμηθευτή.

Ταχύτητα. Ο ανοιχτός κώδικας διευκολύνει την ταχύτητα. Ένα μεγάλο πλεονέκτημα του ανοιχτού κώδικα είναι η ικανότητα άμεσης εγκατάστασης του λογισμικού και συνεπώς άμεσης παραγόμενης αξίας. Η επαγγελματική υποστήριξη για τα προϊόντα ανοιχτού κώδικα από εκατοντάδες εξειδικευμένες εταιρείες και επαγγελματίες συμβούλους θεωρείται δεδομένη.

Αποτελεσματικότητα ως προς το κόστος. Ο ανοιχτός και ελεύθερος κώδικας είναι περισσότερο αποτελεσματικός ως προς το κόστος σε σχέση με τα κόστη αντίστοιχων ιδιόκτητων λογισμικών. Οι λύσεις ανοιχτού κώδικα όχι μόνο είναι

πιο φτηνές σε σχέση με τις ισοδύναμες κλειστού κώδικα, αλλά τις περισσότερες φορές παρέχουν αυξημένες δυνατότητες.

Ασφάλεια πληροφοριών. Το εμπορικό ελεύθερο και ανοιχτό λογισμικό κυριαρχεί στο τομέα της Ασφάλεια Συστημάτων Πληροφορικής. Η ανταπόκριση και η παροχή λύσεων της κοινότητας ανοιχτού κώδικα στα προβλήματα ασφάλειας των λογισμικών της είναι αρκετά υψηλή. Το ότι μπορεί να αναγνωρίζει και να επιλύει προβλήματα μόλις αυτά ανακύψουν είναι το πιο βασικό πλεονέκτημα του ανοιχτού κώδικα, σε σχέση με τους περιορισμούς του κλειστού κώδικα σε ένα ιδιόκτητο περιβάλλον, όπου λίγοι το ξέρουν όσο η εταιρεία που το ανέπτυξε και το προμηθεύει.

Προσέλκυση καλύτερου ανθρώπινου δυναμικού. Το λογισμικό ελεύθερου και ανοιχτού κώδικα, προσελκύει το καλύτερο ανθρώπινο δυναμικό με γνώσεις στις Τεχνολογίες Πληροφορικής και Επικοινωνιών (ΤΠΕ) και τους πιο ικανούς μηχανικούς λογισμικού.

Κόστος συντήρησης. Ένα από τα βασικά πλεονεκτήματα του ανοιχτού κώδικα είναι η συμμετοχή της κοινότητας στη διαρκή συντήρηση και βελτίωση του υπάρχοντος κώδικα. Αντί κανείς να στηρίζεται τεχνολογικά και να επενδύει αναπτύσσοντας μια εφαρμογή που πρέπει να την διατηρήσει μόνος, μπορεί να μοιράζεται με τη κοινότητα τα πολλαπλά κόστη της συντήρησης και ανάπτυξης των εφαρμογών.

Ο ανοιχτός κώδικας αποτελεί το μέλλον στις ΤΠΕ και για αυτό η ανάπτυξη του Έργου θα γίνει αποκλειστικά με χρήση ανοιχτού λογισμικού και ανοιχτών τεχνολογιών. Επιπλέον όλες οι διεπαφές θα είναι αποκλειστικά και εξ' ολοκλήρου διαδικτυακές (web based) που σημαίνει ότι για τη χρήση όλων των συστημάτων που θα αναπτυχθούν θα αρκεί ένας οποιοσδήποτε φυλλομετρητής (όπως Chrome, Safari, Firefox, IE, κ.α.) που συνοδεύεται σε όλα τα σύγχρονα λειτουργικά συστήματα χωρίς να απαιτείται καμία επιπλέον προσθήκη (π.χ. flash) για να λειτουργήσουν. Επιπλέον θα διασφαλιστεί, βάσει των τεχνικών ανάπτυξης που θα χρησιμοποιηθούν (όπως JS fallbacks, χρήση μόνο πιστοποιημένου HTML, CSS, κ.λπ.), το τελικό σύστημα να είναι πλήρως λειτουργικό και από παλαιότερους φυλλομετρητές, τουλάχιστον δύο (2) προγενέστερων εκδόσεων από την πιο πρόσφατη. Αυτό επίσης σημαίνει ότι το σύνολο των υπηρεσιών **θα λειτουργεί σε οποιοδήποτε λειτουργικό σύστημα (όπως Windows, MacOS, Android, Linux, κ.ά.)** αρκεί να υπάρχει διαθέσιμος φυλλομετρητής με σύνδεση στο διαδίκτυο. Σε ότι αφορά στους εξυπηρετητές (servers) θα βασίζονται στο λειτουργικό Linux για να εξασφαλισθεί η μέγιστη ασφάλεια, ταχύτητα και συμβατότητα.

Με τους παραπάνω τρόπους γίνεται ξεκάθαρο ότι το λογισμικό που θα παραχθεί και παραδοθεί στο τέλος του Έργου θα είναι **πλήρως συμβατό με όλες τις γνωστές πλατφόρμες υλικού και λογισμικού.** Θα συνοδεύεται δε από **άδειες**

χρήσης ανοιχτού λογισμικού (όπως MIT, Apache, GPL, κ.λπ.) που επιτρέπει την μελλοντική επέκταση, παραμετροποίηση και τροποποίηση των υποσυστημάτων που θα παραδοθούν καθώς και την ενσωμάτωση νέων χαρακτηριστικών και νέων εφαρμογών όταν αυτό κριθεί απαραίτητο.

6.3 Ανοιχτή Αρχιτεκτονική (open architecture)

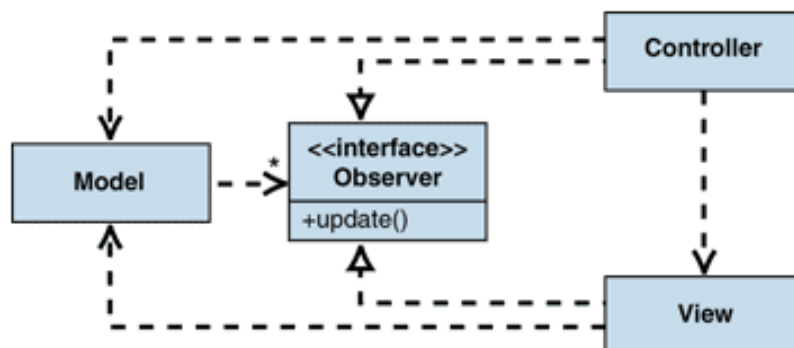
Η σχεδίαση της συνολικής αρχιτεκτονικής του Παρατηρητηρίου έχει γίνει από την αρχή με γνώμονα:

- τη διασφάλιση της συμβατότητας με την προτεινόμενη λογική αρχιτεκτονική,
- τη διασφάλιση της αξιόπιστης επεκτασιμότητας του συστήματος,
- τη διασφάλιση της διασύνδεσης των υποσυστημάτων,
- τη διασφάλιση της πολυεπίπεδης ασφάλειας του συστήματος και
- τη διασφάλιση της εύκολης και απρόσκοπτης αναβάθμισης των πόρων όταν απαιτηθεί με σκοπό να καλυφθεί η αυξανόμενη επισκεψιμότητα χωρίς μεταπτώσεις του συστήματος.

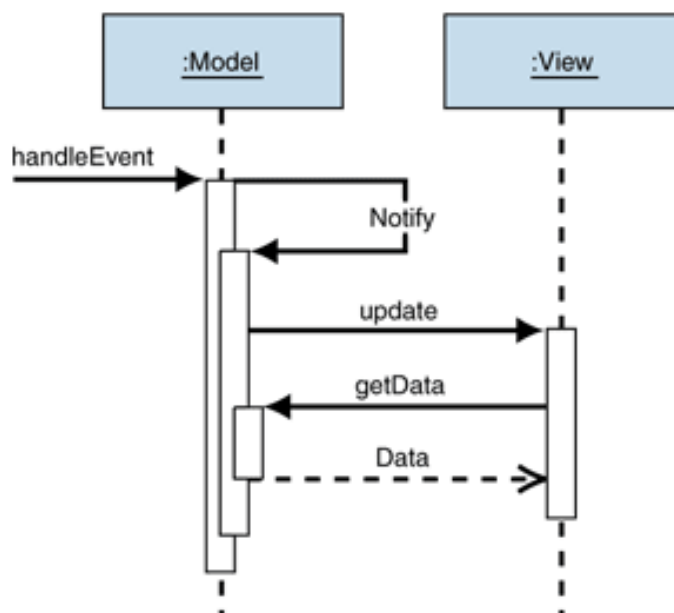
6.3.1 Διασφάλιση της επεκτασιμότητας

Η ομάδα ανάπτυξης της Ένωσης έχει συμμετάσχει σε έργα μεγάλης κλίμακας με πολύ μεγάλες ομάδες προγραμματιστών ενώ μέλη της ομάδας συμμετέχουν ενεργά στην ανάπτυξη εφαρμογών ανοιχτού λογισμικού με εκατοντάδες συνεισφερόμενους σε διεθνές επίπεδο. Συνεπώς, λόγω της μεγάλης εμπειρίας της ομάδας στην υλοποίηση λογισμικού το οποίο επεκτείνεται από πολλούς διαφορετικούς developers, στο προτεινόμενο έργο θα εφαρμοστούν όλες οι καλές πρακτικές από την αρχή της σχεδίασης του και που αφορούν στην αρχιτεκτονική και τους μηχανισμούς επέκτασης των πληροφοριακών συστημάτων όπως είναι η χρήση plugins (ενθεμάτων) και modules (αρθρωμάτων) για κάθε λειτουργία της εφαρμογής. Με το τρόπο αυτό, πέραν της διασφάλισης της επεκτασιμότητας για μελλοντικές αναβαθμίσεις και προσθήκες νέων χαρακτηριστικών, είναι δυνατή και η **παράλληλη υλοποίηση** των υποσυστημάτων, κάτι που διασφαλίζει την τήρηση του χρονοδιαγράμματος κατά την διάρκεια της ανάπτυξης της συνολικής εφαρμογής. Επιπλέον, βασικό πλεονέκτημα της χρήσης ενθεμάτων για τις διάφορες λειτουργίες της εφαρμογής είναι η **επαναχρησιμοποίηση** τους στα διαφορετικά υποσυστήματα. Τα υποσυστήματα που διαθέτουν γραφική διεπαφή (τόσο για τους τελικούς χρήστες όσο και για τους διαχειριστές) και που θα υλοποιηθούν στα πλαίσια του έργου βασίζονται επί της αρχής στην **σχεδίαση MVC ενεργού μοντέλου** (Model-View-

Controller / Active model) σε συνδυασμό με τη χρήση του **πρότυπου Observer pattern** για την επικοινωνία μεταξύ των ενθεμάτων.

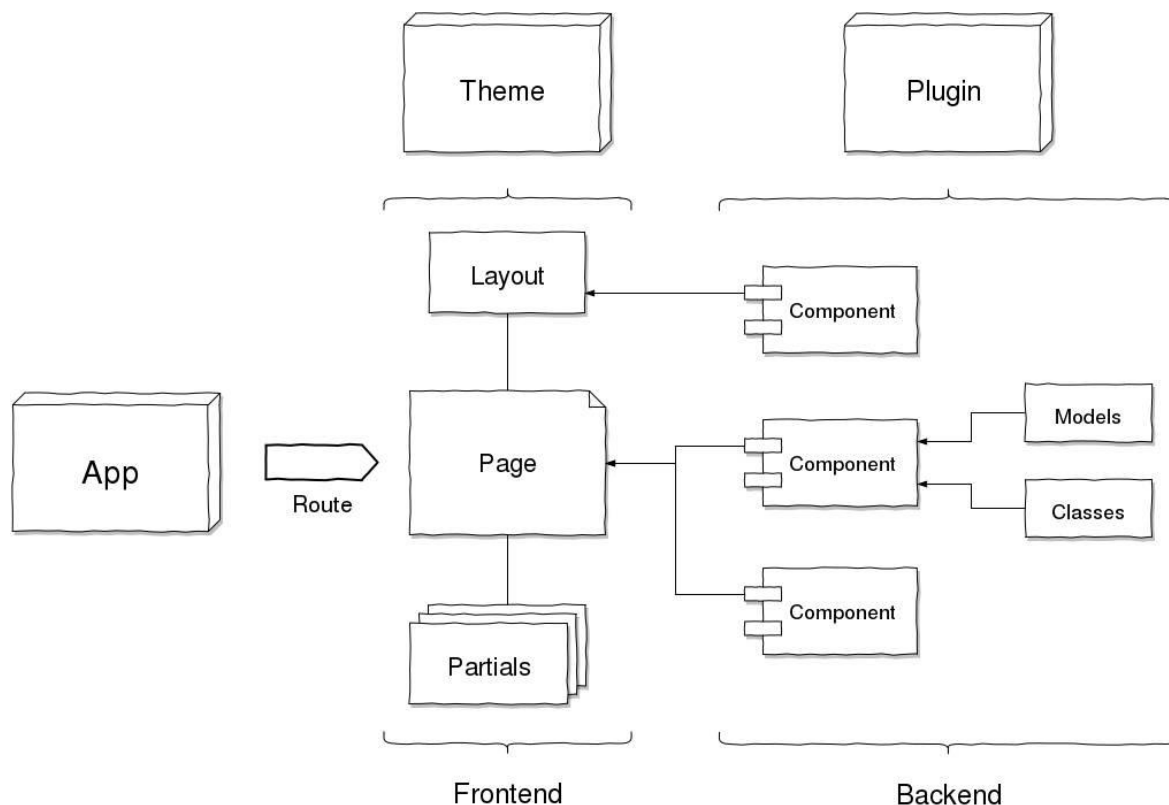


Εικόνα 9. MVC Ενεργό Μοντέλο (Active Model)



Εικόνα 10. Η λειτουργία του Observer (παρατηρητή) στην αρχιτεκτονική MVC

Επειδή η προσβασιμότητα έχει σημαντικό και κύριο ρόλο στην σχεδίαση της αρχιτεκτονικής, η αρχιτεκτονική MVC θα αποτελεί μέρος της σχεδίασης με θέματα (themes) και διατάξεις (layouts) όπως φαίνεται στο παρακάτω διάγραμμα. Αυτή η διαφοροποίηση επιτρέπει την κατασκευή του HTML mark-up, CSS, JS κεντρικά, ενώ τα plugins υλοποιούν μόνο την επιχειρησιακή λογική (business logic). Με το τρόπο αυτό διασφαλίζεται ότι όλα τα ενθέματα θα είναι προσβάσιμα σύμφωνα με τις οδηγίες του WCAG2.0 το οποίο θα εφαρμόζεται στα layouts.



Εικόνα 11. Modular αρχιτεκτονική με χρήση ενθεμάτων (plugins) και θεμάτων (themes) για την διασφάλιση της προσβασιμότητας κατά την ανάπτυξη και επέκταση του συστήματος

6.4 Διασφάλιση της επικοινωνίας και της διασύνδεσης των υποσυστημάτων

Απαραίτητη προϋπόθεση για την επιτυχία της υλοποίησης είναι η διασφάλιση της επικοινωνίας τόσο των υποσυστημάτων του Πληροφοριακού Συστήματος μεταξύ τους (καθώς θα μπορούν να λειτουργούν και αυτόνομα) όσο και με τρίτα συστήματα για την άντληση και το διαμοιρασμό δεδομένων και υπηρεσιών. Συνεπώς μεγάλη έμφαση δίνεται στην υλοποίηση κατάλληλων μηχανισμών που επιτρέπουν την επικοινωνία των υποσυστημάτων βάσει βέλτιστων πρακτικών διασύνδεσης που διασφαλίζουν μεγαλύτερη συμβατότητα. Κάθε υποσύστημα που θα υλοποιηθεί στα πλαίσια του έργου, απαραίτητως θα υποστηρίζει επικοινωνία μέσω **δικτυακών υπηρεσιών** (web services).

Τα web services είναι εφαρμογές client / server που επικοινωνούν μέσω του Παγκοσμίου Ιστού και του πρωτόκολλου μεταφοράς HTTP (Hypertext transfer protocol). Όπως περιγράφεται από την Κοινοπραξία του Παγκόσμιου Ιστού (W3C), οι υπηρεσίες αυτές παρέχουν ένα πρότυπο τρόπο διαλειτουργικότητας των εφαρμογών λογισμικού ανεξάρτητα από πλατφόρμες και πλαίσια στα οποία αυτές έχουν δημιουργηθεί. Όπως κάθε εφαρμογή έτσι και αυτές που βασίζονται σε web services μπορούν να εκτελέσουν πλήθος λειτουργιών. Οι υπηρεσίες αυτές

μπορεί να συνδυαστούν για να επιτευχθούν πολύπλοκες λειτουργίες καθώς επίσης μπορούν και να αλληλοεπιδρούν μεταξύ τους για την παροχή εξελιγμένων υπηρεσιών. Ενώ οι browser-based εφαρμογές του διαδικτύου λειτουργούν με βάση την αναπαράσταση των δεδομένων στους τελικούς χρήστες τα web services εκτός από την παρουσίαση των δεδομένων επιτρέπουν να χρησιμοποιούν οι χρήστες και **προγραμματιστικά** το δίκτυο για να προσπελάσουν τις υπηρεσίες αυτές και να χρησιμοποιήσουν την λειτουργικότητα τους προς όφελος τους.

Όλα τα υποσυστήματα του έργου θα βασίζονται στην αρχιτεκτονική δικτυακών υπηρεσιών **REST** (Representational State Transfer) που αποτελεί ένα σύνολο από αρχές σχεδίασης μιας δικτυακής υπηρεσίας που επικεντρώνεται στους πόρους (π.χ., δεδομένα) ενός συστήματος. Η αρχιτεκτονική REST σήμερα έχει ξεπεράσει τις παλαιότερες μεθόδους (π.χ., SOAP) και αποτελεί βέλτιστη πρακτική καθώς η βασική αρχή σχεδίασης του είναι η ένα-προς-ένα αντιστοίχιση μεταξύ λειτουργιών CRUD (create, read, update, delete) και HTTP μεθόδων.

Οι κλήσεις στο REST API θα περιλαμβάνουν εντός του HTTP αιτήματος (header και body) όλη την πληροφορία (action, parameters, context) που είναι απαραίτητη από τον server για την παραγωγή μιας απόκρισης στο αίτημα αυτό. Τα αιτήματα πρέπει να είναι ολοκληρωμένα και αυτόνομα έτσι ώστε κατά την επεξεργασία τους από τον server να μην απαιτούν την ανάσυρση κάποιας άλλης πληροφορίας η οποία να επηρεάζει την απόκριση στο ερώτημα. Αυτή η «έλλειψη κατάστασης» (stateless) σε μια υπηρεσία REST απαλλάσσει τον server από την ανάγκη συγχρονισμού των δεδομένων μιας συνόδου με μια εξωτερική εφαρμογή. Επιπλέον, επιτρέπει στις υπηρεσίες REST να προσαρμόζονται σε περιπτώσεις όπου απαιτούνται υψηλές επιδόσεις χρησιμοποιώντας clusters of servers με διαμοιρασμό φορτίου (load-balancing) και δυνατότητες για failover, proxies και άλλες τοπολογίες, οι οποίες θα προωθούν τα αιτήματα από τον ένα server στον άλλο με σκοπό την ελάττωση του συνολικού χρόνου απόκρισης στην κλήση μιας δικτυακής υπηρεσίας.

Με βάση το REST, το URI δεν χρησιμοποιείται πλέον για την περιγραφή της ενέργειας που θέλουμε να εκτελέσουμε αλλά μόνο τον εντοπισμό του πόρου επί του οποίου θα ασκηθεί η ενέργεια απλοποιώντας πολύ τη διαδικασία. Στο προτεινόμενο έργο, τα δεδομένα που θα μεταφέρονται μεταξύ των υποσυστημάτων θα είναι βασισμένα στη σημειογραφία **JSON** που δεν περιέχει πλεόνασμα περιεχομένου (content overhead) όπως το XML, κάτι που συμβάλει τόσο στην ταχύτητα όσο και στην μείωση του όγκου δεδομένων (data transfer) που μεταφέρονται από τον server στον client.

Κάθε υποσύστημα για το σύνολο των μεθόδων διασύνδεσης που θα υποστηρίζει θα συνοδεύεται από **δυναμική τεκμηρίωση** του API (προγραμματιστική διεπαφή) που θα καλείται μέσω των web services και συγκεκριμένα με τη χρήση

του **Swagger** που είναι το δημοφιλέστερο πλαίσιο εργαλείων ανάπτυξης API στον κόσμο για την προδιαγραφή **OpenAPI** (OAS) που επιτρέπει την ανάπτυξη ολόκληρου του κύκλου ζωής του API, από το σχεδιασμό και την τεκμηρίωση έως τη δοκιμή και την ανάπτυξη. Με το τρόπο αυτό διασφαλίζεται ότι η πρόσβαση των ανοιχτών δεδομένων του Παρατηρητηρίου σε εξωτερικούς συνεργάτες και ωφελούμενους θα είναι πλήρως τεκμηριωμένη ενώ θα μπορούν να γίνονται και **πραγματικές κλήσεις στις υπηρεσίες REST API ανάλογα με τη διαβάθμιση των χρηστών.**

6.5 Ασφάλεια του συστήματος

Η ασφάλεια του συστήματος αποτελεί σημαντικό παράγοντα που λαμβάνεται υπόψη από τα πρώτα στάδια της ανάλυσης και της σχεδίασης του ΟΠΣ και αφορά σε πολλαπλά επίπεδα. Παρακάτω αναλύονται οι πρακτικές και οι μέθοδοι που θα εφαρμοστούν στο έργο.

- **Χρήση SSL:** Όλα τα υποσυστήματα θα εξυπηρετούνται απαραίτητως μέσω SSL (Secure Socket Layers) γνωστό και ως Ηλεκτρονικό Πιστοποιητικό, το πρωτόκολλο SSL δημιουργεί μια ασφαλή σύνδεση μεταξύ της εκάστοτε ιστοσελίδας (server) και του φυλλομετρητή (browser) του χρήστη. Τα SSL πιστοποιητικά εξασφαλίζουν την ασφαλή ανταλλαγή δεδομένων ανάμεσα στις δύο πλευρές, αποτρέποντας κακόβουλους χρήστες από την υποκλοπή δεδομένων. Επιπλέον η χρήση του SSL πλέον εξασφαλίζει **καλύτερα αποτελέσματα στις μηχανές αναζήτησης**. Τα πιστοποιητικά SSL που θα χρησιμοποιηθούν στο έργο θα βασίζονται στην ελεύθερη Αρχή Πιστοποίησης Let's Encrypt (letsencrypt.org) που χρησιμοποιεί το **PKCS #1 SHA-256 με κρυπτογράφηση RSA Encryption ως αλγόριθμο υπογραφής του πιστοποιητικού** ενώ θα γίνουν οι απαραίτητες ενέργειες σε επίπεδο server για την αυτοματοποιημένη ανανέωση τους χωρίς να απαιτούνται περαιτέρω ενέργειες από τους διαχειριστές. Αξίζει να σημειωθεί ότι όλες οι ιστοσελίδες του ΟΠΣ και της προσβάσιμης Πύλης του έργου θα **εξυπηρετούνται μονίμως μέσω SSL** και όχι μόνο κατά την εγγραφή και τη σύνδεση (registration, login). Με το τρόπο εξασφαλίζεται η μέγιστη δυνατή ασφάλεια. Επιπλέον θα υπάρχουν αυτοματοποιημένοι μηχανισμοί που θα διασφαλίζουν ότι όλοι οι πόροι (όπως φωτογραφίες, βίντεο, javascript, CSS, κ.α.) που χρησιμοποιούνται στις ιστοσελίδες βρίσκονται σε ασφαλείς διευθύνσεις SSL (https).
- **Ασφάλεια στην επικοινωνία μέσω REST:** Στις κλήσεις που απαιτείται ταυτοποίηση χρήστη, δημιουργείται ένα αλφαριθμητικό το οποίο αποτελείται από το password του χρήστη, την τοπική IP, το περιεχόμενο του αιτήματος και την ώρα δημιουργίας του αιτήματος και υπολογίζεται

ο κατατεμαχισμός (hash) του αλφαριθμητικού με τον αλγόριθμο **SHA-256**. Μέσω των headers του αιτήματος αποστέλλεται το hash, το username και η ώρα (Unix Timestamp) που έγινε η κλήση. Ο server χρησιμοποιεί το username για να ανασύρει το password του χρήστη από την ενιαία βάση χρηστών του συστήματος και το χρησιμοποιεί μαζί με την ώρα που διαβάζει από το header, το περιεχόμενο του αιτήματος και την IP του αιτήματος για να υπολογίσει το αναμενόμενο hash. Συγκρίνει το hash αυτό με το hash που έχει αποσταλεί για να βεβαιωθεί ότι όντως ο χρήστης είναι αυτός που ισχυρίζεται και ότι το μήνυμα δεν έχει πειραχθεί. Επιπλέον, συγκρίνει την ώρα που βρίσκεται στα headers με την δικιά του ώρα και σε περίπτωση που το αίτημα είναι παλιότερο των 30 δευτερολέπτων τότε απορρίπτεται ώστε να ελαχιστοποιηθεί η περίπτωση ενός replay attack. Η διαδικασία αυτή είναι ένα **επιπλέον στρώμα ασφάλειας** πάνω από το SSL μέσω του οποίου θα πραγματοποιούνται όλες οι κλήσεις των υπηρεσιών REST.

- **Χρήση ομάδων χρηστών, ρόλων και ανάθεση δικαιωμάτων:** Η ανάθεση των δικαιωμάτων των χρηστών (σε επίπεδο υπηρεσιών, βάσεων δεδομένων, διαχείρισης του backend, κ.λπ.) θα γίνεται στις ομάδες (groups) και κάθε χρήστης θα γίνεται μέλος μια ομάδας. Αυτό διασφαλίζει την εύκολη αλλαγή ρόλων των χρηστών αλλά και την ευκολότερη επικοινωνία (notifications) και καταγραφή (logging).
- **Συνθηματικά (Passwords):** Θα οριστούν κανόνες (π.χ. Ελάχιστο μήκος 10 χαρακτήρες) ενώ για τις ομάδες διαχείρισης θα είναι απαραίτητη η αλλαγή του συνθηματικού ανά τακτά χρονικά διαστήματα. Μετά το registration των χρηστών θα πρέπει να γίνεται επιβεβαίωση μέσω email ώστε να διασφαλιστεί ότι ο χρήστης είναι πραγματικός.
- **Two-factor authentication:** Ο έλεγχος ταυτότητας δύο παραγόντων, μέσω κινητού ή και PC, παρέχει ένα επιπλέον επίπεδο ασφαλείας πέραν του κλασσικού τρόπου σύνδεσης με το όνομα χρήστη και συνθηματικού για την επιβεβαίωση της ταυτότητας του χρήστη κατά την είσοδο στον λογαριασμό του. Ο τρόπος αυτός χρησιμοποιείται σε συνδυασμό με ένα κινητό, ένα tablet ή κάποια άλλη συσκευή παραγωγής ηλεκτρονικών κωδικών, σαν αυτόν που χρησιμοποιούν συνήθως οι τράπεζες. Ο τρόπος αυτός ουσιαστικά προστατεύει από τυχόν κλοπή του username και του password αφού δεν αρκούν πλέον μόνο αυτά για να εισέλθει κάποιος στον λογαριασμό του. Ακόμα κι αν κλαπούν τα στοιχεία πρόσβασης, η σύνδεση στο λογαριασμό του χρήστη, χωρίς τη χρήση του επιπλέον κωδικού two-factor authentication είναι αδύνατη.
- **Καταγραφή των ενεργειών:** Κάθε ενέργεια που θα γίνεται στο ΟΠΣ, θα καταγράφεται λεπτομερώς σε log files Με το τρόπο αυτό θα μπορεί να

γίνεται ανάλυση των δεδομένων ώστε να μειώνονται οι προσπάθειες διαπέρασης του συστήματος από κακόβουλους χρήστες.

- **Εφαρμογή καλών πρακτικών ασφάλειας:** Σε επίπεδο server θα γίνει χρήση των fail2ban για την αποκοπή προσπαθειών διάσπασης των συνθηματικών, no root password ώστε να απαιτείται πιστοποιητικό για τη σύνδεση (SSH), amavis για τον έλεγχο ιών και trojans, χρήση firewall σε επίπεδο λειτουργικού συστήματος. Σε επίπεδο διασφάλισης δεδομένων που βρίσκονται σε αποθήκευση (data-at-rest) θα ενεργοποιηθούν με την παράδοση του έργου τεχνικές κρυπτογράφησης (αν το επιθυμεί η Αναθέτουσα καθώς η κρυπτογράφηση μειώνει ελαφρώς την ταχύτητα των εφαρμογών) στη μεριά του server όπου και θα απαιτείται η χρήση ενός κλειδιού KMS για την αποκρυπτογράφηση τους. Σε επίπεδο frontend, θα χρησιμοποιούνται σε όλες τις φόρμες, μέσω των οποίων θα μεταφέρονται δεδομένα, τα session tokens ώστε να αποφεύγονται επιθέσεις τύπου Cross-Site Request Forgery (CSRF).

6.6 Απρόσκοπτη αναβάθμιση του συστήματος (scaling)

Ο καλός σχεδιασμός ενός διαδικτυακού πληροφοριακού συστήματος πρέπει να λαμβάνει υπόψη, από το στάδιο της σχεδίασης ακόμη, τη δυνατότητα αναβάθμισης των πόρων του συστήματος χωρίς να επηρεάζονται τα επιμέρους υποσυστήματα και χωρίς να γίνεται αντιληπτό από τους τελικούς χρήστες. Για να εξασφαλισθεί ότι το τελικό σύστημα θα μπορέσει να ανταποκριθεί σε αυξημένη ζήτηση πόρων και αυξημένη επισκεψιμότητα πολλαπλών χρηστών ταυτόχρονα αλλά και την αυξημένη ροή δεδομένων, εξ' αρχής όλα τα υποσυστήματα συμπεριλαμβανομένων και των Βάσεων Δεδομένων και των εξυπηρετητών εφαρμογών θα σχεδιαστούν ως web-scale applications. Αυτό σημαίνει ότι θα μπορούν αναλόγως τη ζήτηση, να προσθέσουν πόρους (πχ μνήμη) και αν αυτό δεν αρκεί να επεκταθούν σε πολλαπλούς εξυπηρετητές μέσω διαμοιρασμού του φόρτου. Για να εξασφαλιστούν τα παραπάνω και να μπορεί να είναι δυνατή η εξισορρόπηση φόρτου (load balancing), το πληροφοριακό σύστημα που θα αναπτυχθεί θα είναι βασισμένο στην αρχιτεκτονική 3-βαθμίδων.

6.7 Αρχιτεκτονική 3-βαθμίδων (3-tier)

Η 3-tier αρχιτεκτονική είναι μια client-server αρχιτεκτονική στην οποία η απεικόνιση (presentation), η διαχείριση των δεδομένων (data management) και η επεξεργασία (processing) είναι διακριτές διαδικασίες (Eckerson, 1995). Για παράδειγμα, μια εφαρμογή η οποία χρησιμοποιεί μια ενδιάμεση βαθμίδα (tier)

για να ικανοποιήσει αιτήματα για δεδομένα (data requests) ανάμεσα σε ένα χρήστη και μια βάση δεδομένων μπορεί να θεωρηθεί πως έχει σχεδιαστεί με βάση μια πολυεπίπεδη (multitier) αρχιτεκτονική. Σε ότι αφορά την ορολογία, συχνά ο όρος multitier αρχιτεκτονική αναφέρεται αποκλειστικά στη 3-tier αρχιτεκτονική, ενώ αντί για τον όρο tier χρησιμοποιείται συχνά ο όρος layer με την ίδια σημασία, παρότι υπάρχει μια σημαντική διαφορά μεταξύ τους. Ένα layer είναι ένας λογικά δομημένος μηχανισμός από στοιχεία που απαρτίζουν μια οντότητα λογισμικού (software solution), ενώ ένα tier είναι ένας φυσικά δομημένος μηχανισμός για την υποδομή του συστήματος. Σε κάθε περίπτωση, η 3-tier είναι μια client-server αρχιτεκτονική στην οποία η διεπαφή χρήστη, οι διεργασίες που αποτελούν τη “λογική” της εφαρμογής και η αποθήκη δεδομένων αναπτύσσονται και διατηρούνται σαν ανεξάρτητα τμήματα (modules), τις περισσότερες φορές σε διαφορετικές πλατφόρμες. Η 3-tier αρχιτεκτονική αποτελείται από τις ακόλουθες τρεις βαθμίδες (tiers):

- **Presentation tier:** αποτελεί την “υψηλότερη” βαθμίδα στην εφαρμογή. Από άποψη ευχρηστίας είναι η πιο σημαντική βαθμίδα της εφαρμογής επειδή είναι αυτή την οποία βλέπουν και χρησιμοποιούν οι χρήστες. Ακόμη και με καλά δομημένα τις άλλες βαθμίδες, αν η Presentation βαθμίδα δεν έχει σχεδιαστεί σωστά, θα δώσει στους χρήστες περιορισμένες δυνατότητες στο σύστημα. Είναι η βαθμίδα που εφαρμόζονται όλες οι καλές πρακτικές που διασφαλίζουν τη “σχεδίαση για όλους” προσέγγιση.
- **Application Logic tier:** είναι διαχωρισμένη από την Presentation βαθμίδα και λειτουργεί ως διακριτό κομμάτι της εφαρμογής. Παρόλο που μια εφαρμογή Παγκόσμιου Ιστού θα μπορούσε να επικοινωνήσει άμεσα με τα δεδομένα της, προτείνεται να γίνεται η επικοινωνία μέσω της συγκεκριμένης βαθμίδας, με αποτέλεσμα να ανάγεται σε ζωτικής σημασίας βαθμίδα, καθώς κάνει έλεγχο στα δεδομένα που καταχωρούνται πριν καλέσει μια μέθοδο από το Data tier. Επιπλέον, διασφαλίζει ότι τα δεδομένα που θα εισαχθούν στο σύστημα είναι σωστά, ενώ μπορεί πολύ συχνά να διασφαλίσει ότι και η “έξοδος” του συστήματος είναι σωστή. Αυτή η επαλήθευση (validation) στα δεδομένα είναι μέρος της “λογικής” της εφαρμογής, δηλαδή κάποιων κανόνων που εφαρμόζονται σε υπολογισμούς ή ενέργειες που γίνονται στη συγκεκριμένη βαθμίδα, για να παρθούν αποφάσεις για τα δεδομένα. Επίσης, η ενσωμάτωση της “λογικής” της εφαρμογής στη συγκεκριμένη βαθμίδα, της δίνει την δυνατότητα να είναι επαναχρησιμοποιήσιμη (reusable) μέσα στην εφαρμογή. Η μεσαία βαθμίδα μπορεί να είναι η ίδια πολυεπίπεδη (multi-tier), οπότε στην περίπτωση αυτή η αρχιτεκτονική καλείται n-tier.
- **Data tier:** συνήθως αποτελείται από database servers, στους οποίους αποθηκεύονται τα δεδομένα της εφαρμογής, ανεξάρτητα από τους application servers και τη “λογική” της εφαρμογής. Καθώς τα δεδομένα είναι το βασικό στοιχείο των περισσότερων εφαρμογών, η Data βαθμίδα είναι ένα διακριτό

τμήμα της εφαρμογής, μοναδικός σκοπός του οποίου είναι να εξάγει αποθηκευμένα δεδομένα από τη βάση δεδομένων και να εισάγει νέα σε αυτήν. Μέσω αυτής της προσέγγισης τα δεδομένα μπορούν εύκολα να επαναχρησιμοποιηθούν, κάνοντας την εφαρμογή ευκολότερη στην συντήρηση (maintenance). Επιπλέον η ύπαρξη μιας διακριτής βαθμίδας για τα δεδομένα βελτιώνει την απόδοση και το scalability.

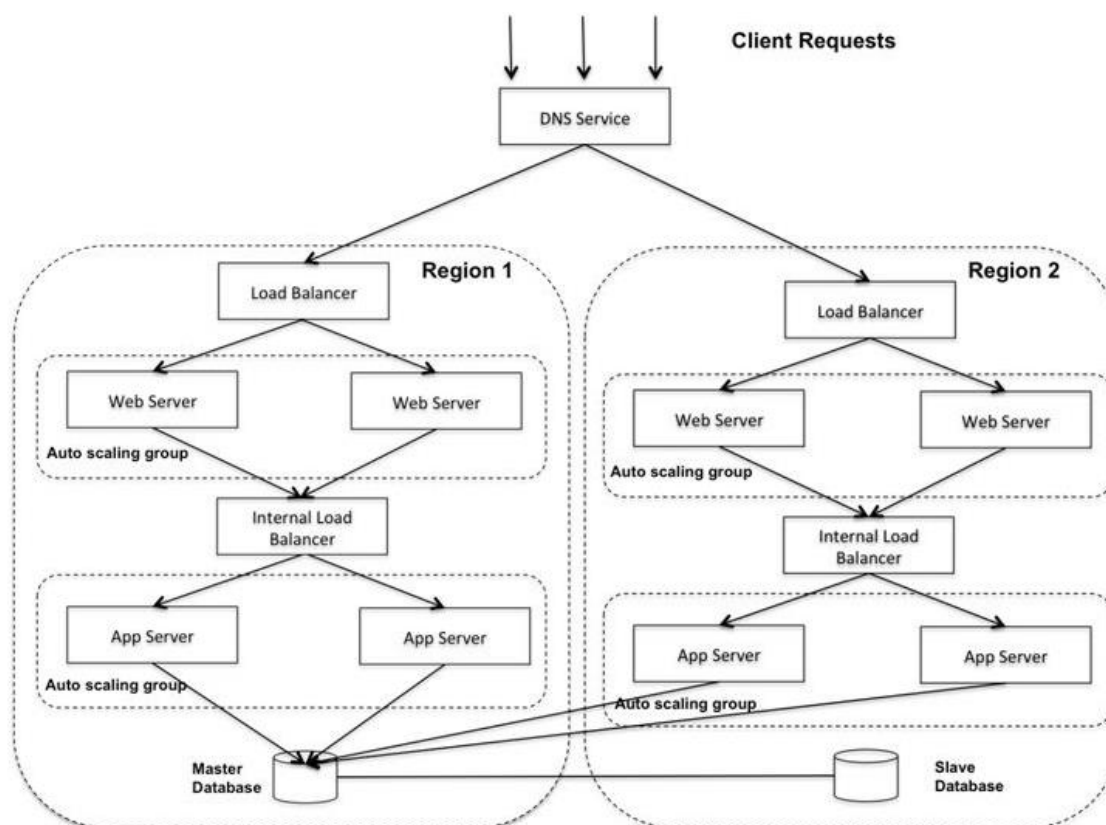
Συνεπώς, η Πλατφόρμα θα αναπτυχθεί και θα λειτουργήσει χρησιμοποιώντας τρεις διακριτές βαθμίδες, ενώ θα χρησιμοποιηθούν υπηρεσίες νέφους ώστε να εξασφαλιστεί η δυνατότητα μαζικής κλιμάκωσης (scaling) τόσο στο εύρος ζώνης (bandwidth) όσο και στον παρεχόμενο αποθηκευτικό χώρο αλλά και η ελαστικότητα (elasticity) του συστήματος που αφορά στην αυξομείωση των υπολογιστικών πόρων ανάλογα με τις ανάγκες του πληροφοριακού συστήματος για τη συγκεκριμένη χρονική περίοδο (για παράδειγμα μέσω του load balancing).

Οι λόγοι που μας οδήγησαν στην υιοθέτηση της αρχιτεκτονικής 3-βαθμίδων είναι:

- Ο τελικός κώδικας κάθε βαθμίδας μπορεί να επαναχρησιμοποιηθεί και σε άλλες βαθμίδες ή τελείως διαφορετικές εφαρμογές που εκτελούν ίδιες λειτουργίες.
- Κάθε βαθμίδα είναι σχετικά εύκολο να υλοποιηθεί, οπότε είναι καλύτερο να υπάρχουν τρία απλά τμήματα σε μια εφαρμογή παρά ένα πολύπλοκο.
- Σε κάθε βαθμίδα μπορεί να εμπλακεί διαφορετική ομάδα ειδικών που θα συμβάλλουν με τις εξειδικευμένες γνώσεις τους στην ανάπτυξή της με βέλτιστο τρόπο.
- Εκτός από τα συνήθη πλεονεκτήματα του “σπονδυλωτού” λογισμικού (modular software), η 3-tier αρχιτεκτονική έχει την τάση να επιτρέπει σε οποιοδήποτε από τις τρεις βαθμίδες να αναβαθμιστεί ή να αντικατασταθεί ανεξάρτητα από τις υπόλοιπες καθώς οι απαιτήσεις και η τεχνολογία αλλάζουν. Για παράδειγμα, μια αλλαγή σε μία βιβλιοθήκη στην Presentation βαθμίδα θα επηρέαζε μόνο τον κώδικα της διεπαφής χρήστη.

Για να εξασφαλιστεί η διασφάλιση της ακεραιότητας των δεδομένων και η απρόσκοπτη λειτουργία του πληροφοριακού συστήματος ακόμα και στην οριακή περίπτωση που ο αριθμός των χρηστών αυξηθεί κατακόρυφα (πχ >10000 χρήστες να προσπελάζουν παράλληλα τους πόρους του συστήματος), η 3-tier αρχιτεκτονική θα στηθεί στο cloud με τέτοιο τρόπο εξ' αρχής ώστε να είναι δυνατή η επέκταση της σε n-tier αρχιτεκτονική όποτε χρειαστεί χωρίς να επηρεαστεί στο ελάχιστο η λειτουργικότητά της.

Το σύστημα συνεπώς θα μπορεί να παραταχθεί (deploy) σε πολλές τοποθεσίες. Συνήθως, αυτές οι τοποθεσίες είναι ξεχωριστές γεωγραφικές περιοχές ή ζώνες. Στο έργο προτείνεται η χρήση της υπηρεσίας Linode ή AWS. Η κλιμακωτή αρχιτεκτονική στο νέφος και η επιλογή της συγκεκριμένης υπηρεσίας, υποστηρίζει την αυτόματη κλιμάκωση και την εξισορρόπηση φορτίου στους διακομιστές web και τους διακομιστές εφαρμογών ενώ μπορεί να υλοποιεί master-slave μοντέλο βάσης δεδομένων σε δύο διαφορετικές ζώνες ή κέντρα δεδομένων (συνδεδεμένα με υψηλές ταχύτητες). Η βασική βάση δεδομένων αντιγράφεται συγχρόνως (και αυτομάτως) στη βοηθητική slave. Συνολικά, η αρχιτεκτονική που επιλέχθηκε αντιπροσωπεύει έναν απλό τρόπο επίτευξης μιας εξαιρετικά επεκτάσιμης και υψηλής ποιότητας εφαρμογής σε περιβάλλον νέφους. Τα παραπάνω απεικονίζονται στην **Εικόνα 12**.

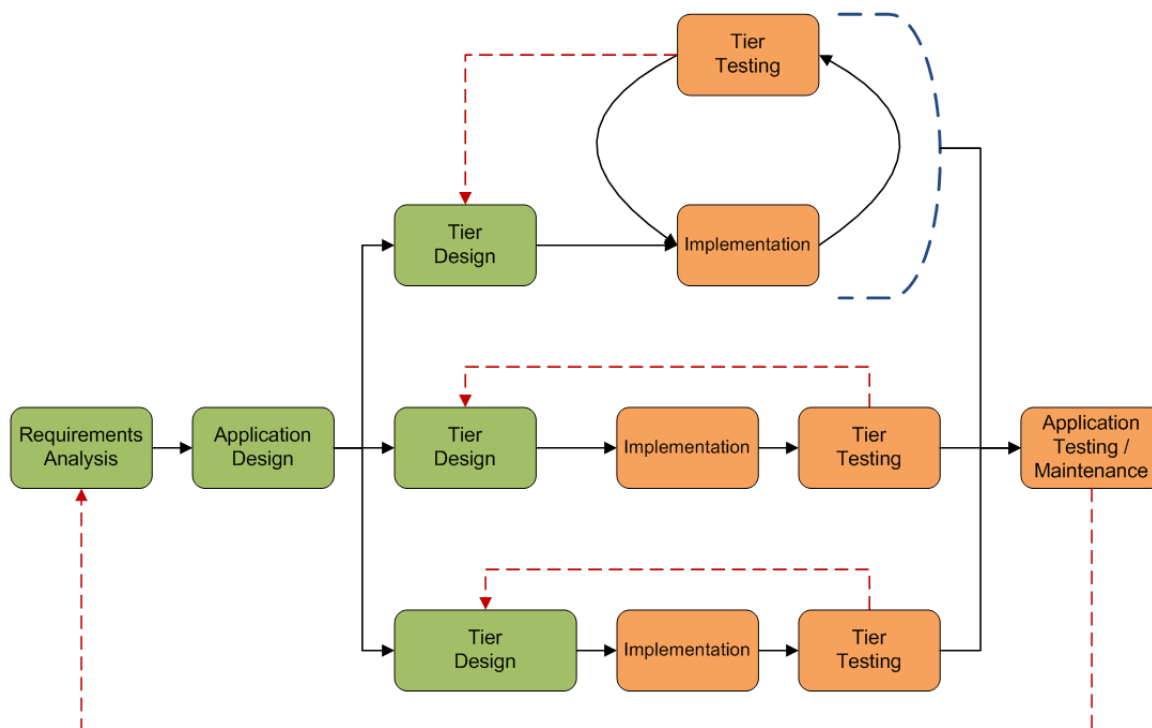


Εικόνα 12. Η Πλατφόρμα σχεδιάζεται με γνώμονα την απρόσκοπτη επέκταση του και την υψηλή διαθεσιμότητα ανεξαρτήτως της αύξησης του φόρτου εργασιών

7 Μεθοδολογία, μέσα και εργαλεία υλοποίησης

7.1 Γενικά

Το Πληροφοριακό Σύστημα που θα αναπτυχθεί στα πλαίσια του έργου, θα είναι βασισμένο σε αρχιτεκτονική 3-βαθμίδων (3-tier architecture) και για το λόγο αυτό, όλες οι ενέργειες ανάπτυξης θα πρέπει να είναι συμβατές με αυτή. Συνεπώς, βασιζόμενοι στο υψηλό επίπεδο ανεξαρτησίας των βαθμίδων της 3-Tier αρχιτεκτονικής, αντιμετωπίζουμε κάθε συστατικό/βαθμίδα σαν μία διακριτή εφαρμογή, σε ότι αφορά στη διαδικασία ανάπτυξης τους, όπως φαίνεται και στο παρακάτω διάγραμμα.



Εικόνα 13. Διαδικασία Ανάπτυξης της πλατφόρμας βασισμένη στην αρχιτεκτονική 3-Βαθμίδων

Η πρώτη φάση αυτής της προσέγγισης είναι μια κοινή φάση “Ανάλυση Απαιτήσεων”, όπου καθορίζονται οι συγκεκριμένοι στόχοι και τα χαρακτηριστικά της τελικής εφαρμογής. Στη συνέχεια, στη φάση “Σχεδιασμού Εφαρμογής”, πραγματοποιείται ο αρχικός και θεμελιώδης σχεδιασμός της πλήρους εφαρμογής, όπου λαμβάνονται οι αποφάσεις για τη φυσική κατανομή των σχετικών τριών βαθμίδων της εφαρμογής, αλλά κυρίως για τα συγκεκριμένα πρωτόκολλα και εργαλεία που θα χρησιμοποιηθούν για τη διασύνδεσή τους.

Ακολούθως, η διαδικασία χωρίζεται σε τρεις κλάδους που εξυπηρετούν το σκοπό της ανάπτυξης των διακριτών τμημάτων της εφαρμογής που προορίζονται για κάθε βαθμίδα της, και μάλιστα, εξαιτίας αυτής της διακριτότητας, μπορούν να διεκπεραιωθούν σχεδόν παράλληλα, κάτι που εξασφαλίζει τη τήρηση των χρονοδιαγραμμάτων:

Για την ανάπτυξη του τμήματος της διεπαφής χρήστη:

- ανάλογα με τα αποτελέσματα της φάσης “Σχεδιασμού Εφαρμογής”, ο “Σχεδιασμός Βαθμίδας” περιλαμβάνει τις αποφάσεις για τα χαρακτηριστικά της διεπαφής χρήστη της εφαρμογής, όπως για παράδειγμα τη χρήση συγκεκριμένων προγραμματιστικών τεχνικών ή την ανάγκη για ανάπτυξη ειδικής διεπαφής με έμφαση στην διατήρηση της προσβασιμότητας ακολουθώντας καλές πρακτικές και πρότυπα (WCAG2.0) προκειμένου να υλοποιηθεί ο επιθυμητός βαθμός αλληλεπίδρασης των τελικών χρηστών με την εφαρμογή,
- κατά τη φάση “Υλοποίησης” πραγματοποιείται η ανάπτυξη μιας πλούσιας διεπαφής χρήστη (με χρήση σύγχρονων μεθόδων όπως είναι οι ασύγχρονες κλήσεις AJAX) για την εφαρμογή στο frontend, και με χρήση PHP (WordPress) για το backend και
- η επαναληπτική διαδικασία μεταξύ της τελευταίας φάσης και της φάσης “Ελέγχου Βαθμίδας” εξασφαλίζει πως έχουν εκπληρωθεί όλα τα προαπαιτούμενα για το συγκεκριμένο τμήμα της εφαρμογής, καθώς και ότι όλη η επικοινωνία με τη μεσαία βαθμίδα θα πραγματοποιείται μέσω του προαποφασισμένου πρωτοκόλλου διασύνδεσης τους που στη περίπτωση του έργου είναι η χρήση web-services και συγκεκριμένα REST.

Προκειμένου να σχεδιαστεί και να υλοποιηθεί το σύστημα διαχείρισης της βάσης γνώσης της εφαρμογής:

- Σχεδιάζεται το κατάλληλο σχήμα για την εφαρμογή και το περιεχόμενό της σχεσιακής βάσης δεδομένων
- η φάση “Υλοποίησης” ουσιαστικά περιλαμβάνει τη δόμηση και το “γέμισμα” των ΒΔ.
- κατά τη φάση “Ελέγχου Βαθμίδας” βεβαιώνεται πως μπορεί να υπάρχει απρόσκοπτη συνεργασία με τους μηχανισμούς διασύνδεσης που ορίστηκαν κατά τη φάση του “Σχεδιασμού Εφαρμογής” (REST API).

Όσο για τη μεσαία βαθμίδα, οι διαδοχικές φάσεις “Σχεδιασμού Βαθμίδας”, “Υλοποίησης” και “Ελέγχου Βαθμίδας” ασχολούνται με την ανάπτυξη των συγκεκριμένων προγραμματιστικών τμημάτων που θα υποστηρίζουν τα χαρακτηριστικά της “λογικής” της εφαρμογής (business and application logic), προσέχοντας να ικανοποιούν τις απαιτήσεις διασύνδεσης των βαθμίδων.

Τέλος, η ολοκληρωμένη τελική εφαρμογή δοκιμάζεται στη φάση “Ελέγχου Εφαρμογής/Συντήρησης” με βάση συγκεκριμένα σενάρια ελέγχου και ενδεικτικά παραδείγματα χρήσης, κάνοντας πιθανές προσαρμογές όπου απαιτείται, προκειμένου να παραδοθεί στους τελικούς χρήστες ένα πιο ολοκληρωμένο και αξιόπιστο τελικό προϊόν. Πρέπει να σημειωθεί πως στην περίπτωση που εντοπιστεί κάποια κρίσιμη έλλειψη ή σφάλμα της εφαρμογής κατά τη διάρκεια των φάσεων ελέγχου (ή κατά τη διάρκεια της πιλοτικής λειτουργίας της εφαρμογής), η διαδικασία είτε επιστρέφει στην αντίστοιχη φάση “Σχεδιασμού Βαθμίδας”, αν το πρόβλημα παρατηρηθεί σε κάποια φάση “Ελέγχου Βαθμίδας”, είτε οδηγείται σε επανεκκίνηση, αν το πρόβλημα παρατηρηθεί στην τελική φάση “Ελέγχου Εφαρμογής/Συντήρησης”.

7.2 Προσέγγιση υλοποίησης

Η *ευελιξία* είναι απαραίτητη σε όλες τις φάσεις ανάπτυξης των εφαρμογών Ιστού. Ευελιξία είναι η ικανότητα να προλαβαίνουμε και να μπορούμε να ανταποκριθούμε στις εξελίξεις οι οποίες μπορεί να έχουν δυσμενείς συνέπειες για την εφαρμογή που αναπτύσσεται, δηλαδή η ανάπτυξη της εφαρμογής να μπορεί να προσαρμοστεί και να συμβαδίζει με σχετικές αλλαγές των απαιτήσεων, των γνώσεων του συστήματος και του περιβάλλοντος στο οποίο αυτό αναπτύσσεται. Η **ευέλικτη υλοποίηση** (agile development) που θα εφαρμοστεί στο παρόν Έργο, βασίζεται σε ένα σύνολο ευέλικτων αρχών και πρακτικών, που σκοπό τους έχουν την ενθάρρυνση των αλλαγών των απαιτήσεων σύμφωνα με τις προκύπτουσες ανάγκες στη διάρκεια του Έργου.

Η πρόοδος μετριέται σε σχέση με το παραγόμενο προϊόν που παραδίδεται, ελαχιστοποιώντας τη σπατάλη χρόνου για τις αλλαγές που μπορεί να προκύψουν. Η πρόοδος των εργασιών που ολοκληρώθηκαν χρησιμοποιούνται για την πρόβλεψη του χρόνου ολοκλήρωσης των μελλοντικών εργασιών, και την αναλόγως δέσμευση επιπλέον πόρων εφόσον αυτό είναι αναγκαίο.

Πιο συγκεκριμένα, η πρόοδος των εργασιών βασίζεται στη χρήση χρονοκύβων (time boxes) και γίνεται ξεκινώντας έναν αριθμό επαναλήψεων (iterations), οι οποίες έχουν προκαθορισμένη χρονική διάρκεια. Η ομάδα ανάπτυξης επικεντρώνεται στο σκοπό της κάθε επανάληψης που είναι σχετικός με τις απαιτήσεις των χρηστών, ενώ ταυτόχρονα παρατηρεί την πρόοδο που έχει

συντελεστεί με βάση το σύνολο των απαιτήσεων. Η χρήση των time boxes μας επιτρέπει να υπολογίσουμε το χρόνο που απαιτείται για να τελειώσουν οι μη ολοκληρωμένες εργασίες μέσα σε κάθε επανάληψη. Ολοκλήρωση μιας επανάληψης σημαίνει ενσωμάτωση με τις προηγούμενες διαδικασίες, δοκιμή και τεκμηρίωση, δηλαδή κάθε εργασία θα πρέπει να τελειώνει με ένα ενδιάμεσο ή πλήρως λειτουργικό τμήμα λογισμικού.

Ο σχεδιασμός μιας επανάληψης απεικονίζεται στο παρακάτω διάγραμμα.



Εικόνα 14. Σχεδιασμός επαναλήψεων για την πρόοδο των εργασιών ανάπτυξης εφαρμογών Ιστού στο Έργο

Πιο συγκεκριμένα, η μεθοδολογία ανάπτυξης που θα εφαρμοστεί στο Έργο, είναι η **ευέλικτη ανάπτυξη** (agile development) γιατί:

- Χρησιμοποιούνται επαναλήψεις μεταβλητού μήκους ανάλογα με τη φάση και τις ανάγκες του Έργου.
- Χρησιμοποιούνται επιπλέον εσωτερικές επαναλήψεις μικρής διάρκειας, μέχρι και οκτώ εβδομάδων, ώστε να αποφεύγονται μακροσκελές εργασίες χωρίς ενδιάμεσο επανέλεγχο / επαλήθευση με τον χρήστη.
- Η αξιολόγηση για το σχεδιασμό και την υλοποίηση των σχετικών εργασιών γίνεται σε κάθε μία από τις επαναλήψεις ξεχωριστά.
- Τα καθήκοντα εντός των επαναλήψεων είναι σαφώς καθορισμένα ώστε να ολοκληρώνονται πάντοτε εγκαίρως.

- Χρησιμοποιούνται μηχανισμοί που καθορίζουν πότε οι υποσχόμενες απαιτήσεις έχουν ολοκληρωθεί.

Η ευέλικτη μεθοδολογία διαχείρισης, βελτίωσης και συντήρησης του συνολικού συστήματος που θα αναπτυχθεί με σκοπό την διασφάλιση της ποιότητας αλλά και των προκαθορισμένων χρονοδιαγραμμάτων είναι η **μεθοδολογία Scrum**.

7.3 Μεθοδολογία Scrum

Η Scrum είναι μια ευέλικτη μεθοδολογία που χρησιμοποιείται συχνά για την ανάπτυξη ενός προϊόντος και κυρίως την ανάπτυξη λογισμικού. Η Scrum παρέχει ένα πλαίσιο διαχείρισης γενικού σκοπού και μπορεί να εφαρμοστεί σε εφαρμογές με πιεστικές ημερομηνίες παράδοσης, με σύνθετες απαιτήσεις και συγκεκριμένες ιδιαιτερότητες.

Έτσι, αντί να παρέχουμε ολοκληρωμένες και λεπτομερείς περιγραφές για το πώς θα πρέπει να γίνουν τα πάντα σε ένα Έργο, το μεγαλύτερο βάρος ανατίθεται στην ομάδα ανάπτυξης του συστήματος. Αυτό γίνεται γιατί η ομάδα γνωρίζει καλύτερα πώς να επιλύσει το πρόβλημα που παρουσιάστηκε. Αυτός είναι ο λόγος για παράδειγμα που μία συνάντηση σχεδίασης (planning meeting – π.χ., βλ. προγραμματισμένες *Εσωτερικές Συναντήσεις* του Έργου) περιγράφεται με βάση το επιθυμητό αποτέλεσμα (μία δέσμευση για να τεθούν τα χαρακτηριστικά που πρέπει να αναπτυχθούν μέχρι την επόμενη συνάντηση) και όχι ένα σύνολο εισαγωγικών κριτηρίων, καθορισμοί καθηκόντων, κριτήρια επικύρωσης και κριτήρια εξόδου όπως θα παρέχονταν στις περισσότερες μεθοδολογίες¹⁶.

Τα έργα που βασίζονται στη μεθοδολογία Scrum σημειώνουν πρόοδο των εργασιών σε μια σειρά από επαναλήψεις βασιζόμενες στο χρόνο, που ονομάζονται sprints. Ο χρόνος οριοθετείται σε επιμέρους χρονοκύβους και τα sprints επαναλαμβάνονται σε προκαθορισμένο χρονικό διάστημα (4 εβδομάδες για το παρόν Έργο). Κατά την έναρξη ενός sprint, τα μέλη της ομάδας δεσμεύονται να παρέχουν κάποια χαρακτηριστικά που αναφέρονται στο ανεκτέλεστο υπόλοιπο των τμημάτων του έργου. Στο τέλος του sprint, αυτά τα χαρακτηριστικά ολοκληρώνονται, κωδικοποιούνται, δοκιμάζονται και ενσωματώνονται στο υπό εξέλιξη προϊόν ή σύστημα. Στο τέλος του sprint, γίνεται μια επανεξέταση κατά την οποία η ομάδα παρουσιάζει την πρόοδο στον Τεχνικό Υπεύθυνο του έργου καθώς επίσης και στα υπόλοιπα ενδιαφερόμενα μέρη που παρέχουν ανατροφοδότηση και θα μπορούσαν να επηρεάσουν το επόμενο sprint.

¹⁶ Βλ. <http://www.mountaingoatsoftware.com/topics/scrum>

Τα βασικά χαρακτηριστικά της διαδικασίας Scrum είναι τα παρακάτω:

- Πρόκειται για μια ευέλικτη μεθοδολογία που μας επιτρέπει να επικεντρωθούμε στην παροχή της υψηλότερης επιχειρηματικής αξίας στο συντομότερο χρονικό διάστημα.
- Το προϊόν αναπτύσσεται σταδιακά σε μια σειρά από sprints.
- Οι απαιτήσεις καταγράφονται ως στοιχεία σε μια λίστα ανεκτέλεστου προϊόντος (product backlog).
- Οι ομάδες αυτό-οργανώνονται για να καθορίσουν τον καλύτερο τρόπο παράδοσης των υψηλότερης προτεραιότητας χαρακτηριστικών.
- Χρησιμοποιεί ένα σύνολο πρακτικών για να δημιουργήσει ένα ευέλικτο περιβάλλον παράδοσης των εκροών.
- Κάθε 4 εβδομάδες ο καθένας μπορεί να δει μια πραγματική έκδοση λειτουργικού λογισμικού και να αποφασίσει αν θα την αποδεχθεί, όπως είναι ή αν θα συνεχίσει τη βελτίωσή της για μια ακόμη περίοδο sprint.
- Οι φάσεις του σχεδιασμού και της ολοκλήρωσης αποτελούνται από διαδικασίες, εισόδους και εξόδους οι οποίες είναι καλά καθορισμένες. Η γνώση για τον τρόπο που εκτελούνται οι διαδικασίες είναι σαφής και η ροή γραμμική.
- Το sprint είναι μια εμπειρική διαδικασία στη διάρκεια του οποίου πολλά καθήκοντα είναι απροσδιόριστα και μη ελεγχόμενα. Αντίστοιχα, έλεγχοι που περιλαμβάνουν τη διαχείριση επικινδυνότητας εφαρμόζονται σε κάθε επανάληψη του sprint ώστε να αποφύγουμε δυσμενής καταστάσεις και μεγιστοποιώντας την ευελιξία.

Τα εργαλεία που θα χρησιμοποιηθούν κατά τη διάρκεια ανάπτυξης του έργου για την διαχείριση και την καταγραφή της προόδου είναι πρωτίστως το **OpenProject Community Edition** που υποστηρίζει πλήρως τη μεθοδολογία Scrum και το οποίο θα στηθεί σε server της Ένωσης, ενώ σε πιο τεχνικό επίπεδο θα χρησιμοποιηθεί το **Trello** και το **MantisBT** στην έκδοση 2.5.1 για τη συνεργασία των μελών της ομάδας ανάπτυξης του λογισμικού.

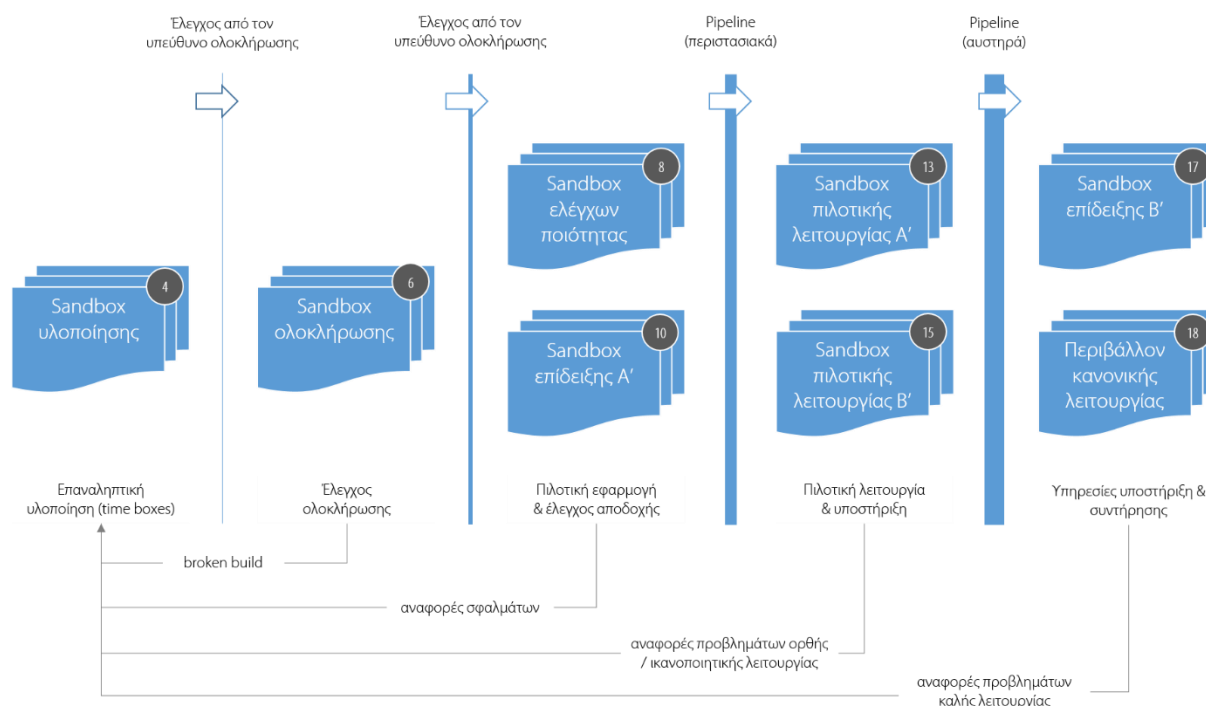
7.4 Διασφάλιση ποιότητας και ελέγχου λογισμικού

Σημαντικό μέρος στην ομαλή πορεία υλοποίησης έχει η φάση ελέγχου του λογισμικού, ώστε να διασφαλίζεται σε κάθε στάδιο η ποιότητα του. Στο Έργο, ο Προσφέρων σκοπεύει να εφαρμόσει σύγχρονες και βέλτιστες πρακτικές ελέγχου που διασφαλίζουν την ποιότητα του παραγόμενου λογισμικού. Βασική αρχή της ομάδας ανάπτυξης της Ένωσης είναι η χρήση συστήματος διαχείρισης εκδόσεων του πηγαίου κώδικα.

7.4.1 Sandboxing

Κατά τη ανάπτυξη των εφαρμογών του Έργου, θα λειτουργήσει σε «**σκάμμα άμμου**» (sandbox), όπως ονομάζεται το περιβάλλον που έχει αναπτυχθεί ειδικά για την παράλληλη ανάπτυξη τμημάτων λογισμικού και για δοκιμές λογισμικού, ώστε αυτές να γίνονται με ασφάλεια χωρίς τυχόν αστοχίες να πέσουν στο «σκληρό έδαφος» (δηλαδή χωρίς να οδηγήσει σε ανεπανόρθωτα ή χρονοβόρα προβλήματα) του κυρίως περιβάλλοντος παραγωγής. Σε συνθήκες sandbox απομονώνονται οι μη ελεγμένες αλλαγές στον κώδικα (test server) και οι όποιοι πειραματισμοί στο πλαίσιο των δοκιμών από το κυρίως (master) περιβάλλον παραγωγής ή /και το τελικό αποθετήριο. Το sandboxing προστατεύει τους "ζωντανούς" εξυπηρετητές και τα δεδομένα τους και άλλες συλλογές κώδικα, δεδομένων ή / και περιεχομένου, από αλλαγές που θα μπορούσαν να είναι καταστροφικές (ανεξάρτητα από την πρόθεση του συντάκτη αυτών των αλλαγών) ή να είναι απλώς δύσκολο να ανακτηθούν / επανέλθουν στην αρχική τους κατάσταση. Τα sandboxes αναπαράγουν τουλάχιστον τις ελάχιστες λειτουργίες που απαιτούνται για την ακριβή δοκιμή, ολόκληρων ή τμημάτων, των υπό ανάπτυξη συστημάτων (π.χ., χρήση των ίδιων μεταβλητών περιβάλλοντος ή πρόσβαση σε αντίγραφο της βάσης δεδομένων που χρησιμοποιείται από το master περιβάλλον.

Στο παρακάτω σχήμα παρουσιάζεται ενδεικτικά η εφαρμογή της τεχνικής sandboxing.



Εικόνα 15. Εφαρμογή της τεχνικής sandboxing στο Έργο

Για τις ανάγκες του έργου θα χρησιμοποιηθεί το Docker (για το sandboxing) και το git (για το versioning). Η διαχείριση εκδόσεων συμβάλλει στην ποιότητα του λογισμικού καθώς:

- Επιτρέπει στην ομάδα ανάπτυξης την αναίρεσης μέρος του κώδικα
- Επιτρέπει την διατήρηση διαφορετικών εκδόσεων ενός αρχείου του πηγαίου κώδικα.
- Προσφέρει ελεγχόμενη πρόσβαση στον πηγαίο κώδικα του λογισμικού.
- Δίνει τη δυνατότητα για παράλληλη εργασία των προγραμματιστών στην ίδια βάση κώδικα (code base).
- Επειδή το σύστημα κρατάει όλες τις αλλαγές στον κώδικα, η ομάδα είναι σε θέση να εντοπίσει εύκολα ποιος έχει κάνει αλλαγές στον κώδικα και γιατί
- Επιτρέπει την παραγωγή πολλών κυκλοφοριών (releases) του λογισμικού, ενώ ταυτόχρονα η ομάδα ανάπτυξης εργάζεται στη βασική γραμμή ανάπτυξης.

Επιπλέον για τη διασφάλιση της ποιότητας του λογισμικού θα χρησιμοποιούνται όπου είναι δυνατόν τεχνικές scaffolding. Αυτό σημαίνει ότι μέρη του πηγαίου κώδικα που επαναλαμβάνονται, θα δημιουργούνται κατά τη δημιουργία τους αυτομάτως.

7.4.2 Εκσφαλμάτωση

Σε ότι αφορά τους ελέγχους (testing) θα γίνονται τόσο σε δομικό επίπεδο (white-box tests) ώστε να ελέγχονται τα βασικά low-level δομικά στοιχεία των υποσυστημάτων όσο και σε επίπεδο συμπεριφοράς του συστήματος (black-box tests). Οι έλεγχοι black-box θα εντατικοποιηθούν στην πιλοτική φάση του έργου, ενώ οι white-box έλεγχοι θα ξεκινήσουν από την πρώτη μέρα της έναρξης υλοποίησης των υποσυστημάτων.

Οι έλεγχοι που θα γίνονται στις επιμέρους μονάδες λογισμικού στο έργο θα αφορούν συγκεκριμένα σε:

- αλγοριθμικά σφάλματα
- υπολογιστικά λάθη και λάθη ακρίβειας
- σφάλματα υπερφόρτωσης
- σφάλματα ορίων ή χωρητικότητας
- σφάλματα χρονισμού ή σφάλματα συντονισμού
- σφάλματα απόδοσης
- σφάλματα ανάκαμψης

- σφάλματα υλικού και λογισμικού συστήματος
- σφάλματα τεκμηρίωσης
- Τα κυριότερα εργαλεία που θα χρησιμοποιηθούν για αυτοματοποιημένους ελέγχους στο έργο είναι:
- PHPUnit ως το κύριο framework
- Travis CI για διαρκείς κοινοποιήσεις κώδικα (continuous integration) για τον έλεγχο κατά την ενοποίηση (integration) και τη διασύνδεση των υποσυστημάτων μεταξύ τους
- Codeception για έλεγχο της επικοινωνίας REST του συστήματος με τρίτα συστήματα

7.4.3 Ποιοτικός έλεγχος

Τέλος, πρέπει να αναφερθεί ότι τα συστατικά μέρη του Πληροφοριακού Συστήματος θα αναπτυχθούν βάσει διεθνών προτύπων και κανόνων για την ανάπτυξη διαδικτυακών εφαρμογών. Κατ' ελάχιστο, θα ληφθούν υπόψη τα πρότυπα και οι διεθνείς προδιαγραφές της Κοινοπραξίας του Παγκόσμιου Ιστού (World Wide Web Consortium - W3C) για χρήση έγκυρου κώδικα XHTML, HTML, και CSS.

Καθ' όλη τη διάρκεια ανάπτυξης των διεπαφών των υποσυστημάτων που θα αναπτυχθούν θα γίνεται και ενδεδειγμένος έλεγχος προσβασιμότητας ώστε να διασφαλίζεται ανά πάσα στιγμή η προσέγγιση "σχεδίαση για όλους". Πιο συγκεκριμένα θα ελέγχεται η βαθμίδα προσβασιμότητας με βάσει το πρότυπο WCAG2.0. Στόχος θα είναι η επίτευξη της μέγιστης βαθμίδας σε κάθε μονάδα των υποσυστημάτων.

Τα κυριότερα εργαλεία ελέγχου που θα χρησιμοποιούνται κατά την ανάπτυξη είναι:

- WebAim
- Total Validator
- CSS Validator
- HTML5 Validator
- Colour contrast analyser
- aXe
- VoiceOver για OSX, iOS
- JAWS για Windows
- BrowseAloud

Σημειώσεις: (1) Η ομάδα έργου έχει μεγάλη εμπειρία στην εφαρμογή των διεθνών προτύπων καθώς και στον έλεγχο αυτών σε κάθε φάση της ανάπτυξης. **(2)** Η διασφάλιση της προσβασιμότητας δεν διασφαλίζεται μόνο με τη χρήση αυτοματοποιημένων εργαλείων αλλά απαιτεί και “χειροκίνητους” μη-αυτοματοποιημένους ελέγχους που βασίζονται στην εμπειρία των ελεγκτών. **(3)** Συνεπώς, η παραπάνω λίστα εργαλείων δεν είναι εξαντλητική αλλά ενδεικτική.

7.4.4 Αγωγός έκδοσης (pipeline approach)

Η προσέγγιση αυτή αφορά στον έλεγχο της κάθε έκδοσης (είτε στο sandbox είτε στο κυρίως περιβάλλον παραγωγής) ακολουθώντας καλά καθορισμένα βήματα που θα πρέπει να έχουν ολοκληρωθεί πριν ξεκινήσει ο κάθε κύκλος πιλοτικής εφαρμογής.

Τα βήματα είναι:

- Εκτέλεση αυτοματοποιημένων δοκιμών (automated tests)
- Εκτέλεση σε περιβάλλον δοκιμής / ελέγχου ποιότητας (expert-based)
- Τεκμηρίωση της έκδοσης (release tag)
- Παράδοση προς δοκιμή από χρήστες

Η προσέγγιση αυτή σημαίνει ότι ο όποιος νέος κώδικας εισαχθεί στο κυρίως περιβάλλον υλοποίησης ή παραγωγής, πρέπει να περάσει από τον «αγωγό». Δηλαδή μια λειτουργία μπορεί να φτάσει στην παραγωγή μόνο όταν ο αρμόδιος ΥΔΠ και ο ΤΔ το έχουν εγκρίνει. Συνεπώς, είναι πολύ σημαντικό η ομάδα υλοποίησης και ο «αγωγός» να μπορούν να ανταποκριθούν γρήγορα στις αλλαγές.

7.5 Τεκμηρίωση

Ως γνωστόν, δεν είναι ασυνήθιστο σε μια εταιρία ανάπτυξης υψηλής ποιότητας λογισμικού να δαπανά το 20% ή 30% από όλη την προσπάθεια ανάπτυξης στην τεκμηρίωση λογισμικού (software documentation), η οποία αποτελεί αναπόσπαστο, βασικό συστατικό της μηχανικής λογισμικού, και εξυπηρετεί ένα μεγάλο αριθμό στόχων. Τεκμηρίωση λογισμικού είναι η πληροφορία που έχει σχεδιαστεί με σκοπό να διευκολύνει την αλληλεπίδραση μεταξύ του λογισμικού και των ανθρώπων που το διοικούν, το ελέγχουν, το χειρίζονται ή το συντηρούν. Έτσι, όπως αναφέρθηκε παραπάνω, με τον όρο τεκμηρίωση εννοούμε δυο διαφορετικά στοιχεία τα οποία μάλιστα έχουν συχνά και διαφορετικούς αποδέκτες (α) το σύνολο του υλικού που **περιγράφει τα συστατικά μέρη και τις λειτουργίες ενός νέου συστήματος**, και τις τροποποιήσεις που έγιναν σε ένα υπάρχον σύστημα, καθώς και (β) το σύνολο του υλικού που **βοηθά τους**

χρήστες να κατανοήσουν τις λειτουργίες του συστήματος, να αντιμετωπίζουν προβλήματα, και κάνουν πιο αποτελεσματική και αποδοτική χρήση αυτού.

Ως προς το πρώτο μέρος, η σωστή τεκμηρίωση των εφαρμογών, συμπεριλαμβάνει τη τεκμηρίωση του πηγαίου κώδικα (source code), έτσι ώστε η όποια μετέπειτα προσαρμογή του κώδικα, ανάλογα με τις ανάγκες, να μπορεί εύκολα να ελεγχθεί, αλλαχθεί και να γίνει δηλαδή εκσφαλμάτωση. Αυτό φυσικά προϋποθέτει ότι πηγαίος κώδικας είναι όσο το δυνατόν περισσότερο δομημένος και καθαρός. Η απλότητα, η διαύγεια και η κομψότητα είναι τα χαρακτηριστικά των καλών προγραμμάτων. Η σκοτεινότητα και η πολυπλοκότητα είναι ενδείξεις ανεπαρκούς σχεδιασμού και όχι ευθείας σκέψης και αποκαλύπτονται στην τεκμηρίωση. Επιπλέον, η τεκμηρίωση περιγράφει τους αλγόριθμους, τις μεθόδους και τις τεχνικές που οι προγραμματιστές έχουν χρησιμοποιήσει για τη δημιουργία του λογισμικού, και τεκμηριώνουν τα επίπεδα συμμόρφωσης με τις απαιτούμενες τεχνικές προδιαγραφές. Επίσης, οι αναλυτές συστημάτων και οι σχεδιαστές χρησιμοποιούν τεύχη τεκμηρίωσης για να μεταδώσουν τις απαιτήσεις του συστήματος και το σχεδιασμό του στους χρήστες, στη διοίκηση και στις ομάδες υλοποίησης, κατά τη διάρκεια της διαδικασίας ανάπτυξης του λογισμικού.

Ως προς το δεύτερο μέρος, το οποίο εφοδιάζει τους χρήστες με πληροφορίες, οδηγίες, παραδείγματα και εξηγήσεις για το πως να χρησιμοποιούν το λογισμικό, η τεκμηρίωση έχει τους εξής στόχους από πλευράς του Προσφέροντα:

- Να υπερνικήσει τους φόβους των χρηστών. Η επιτυχημένη τεκμηρίωση οδηγεί σε επιτυχημένη πρώτη αντιμετώπιση του λογισμικού και επομένως σε μεγαλύτερη αποδοχή και χρήση.
- Να αυξήσει την αποτελεσματικότητα. Το προσωπικό του φορέα Λειτουργίας χρειάζεται να αντιλαμβάνεται πλήρως το σύστημα με το οποίο θα εργάζεται. Και φυσικά αυτό δεν μπορεί να βασίζεται σε πληροφορίες που μεταδίδονται από στόμα σε στόμα (π.χ., σε ένα σεμινάριο ή από συνάδελφο σε συνάδελφο), γιατί το προσωπικό (μπορεί να) αλλάζει.

Συνολικά, η τεκμηρίωση που η Ομάδα Έργου σκοπεύει να αναπτύξει συμπεριλαμβάνει:

Τεκμηρίωση για τους χρήστες, με όλες τις απαραίτητες πληροφορίες που χρειάζονται οι χρήστες για να κατανοήσουν τα οφέλη που θα έχουν από τη χρήση των συστημάτων και για να εκτελέσουν αποτελεσματικά την εργασία τους, συμπεριλαμβανομένων των απαιτούμενων εγχειριδίων χρήσης (user manuals), σε μορφές κατάλληλες για διάφορες κατηγορίες χρηστών (tutorials, thematic και lists of reference) και σε προσβάσιμες ψηφιακές μορφές.

Τεχνική τεκμηρίωση υλοποίησης, με απαραίτητες πληροφορίες για τον πηγαίο κώδικα, τους αλγόριθμους, τις διεπαφές (APIs), κ.λπ., π.χ., για την εισαγωγή νέων προς το σύστημα προγραμματιστών στη φιλοσοφία, τη δομή και τα επιμέρους τμήματα του λογισμικού.

Τεκμηρίωση διαχείρισης λειτουργίας, η οποία θα απευθύνεται στο προσωπικό που θα διαχειρίζεται το σύστημα και θα παραθέτει τους κανόνες ασφάλειας, εφεδρείας, ελέγχου, συντήρησης, και βοηθητικές λειτουργίες.