ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάλυση & Υλοποίηση Διαδικτυακής Πλατφόρμας Παραγωγής Ραδιοφωνικού Προγράμματος

Του φοιτητή
Σαμαρά Ορέστη

Αρ. Μητρώου: 113791

Επιβλέπων καθηγητής
Διαμαντάρας Κωνσταντίνος

Θεσσαλονίκη, Ιούνιος 2015

Πρόλογος

Η Πτυχιακή αυτή Εργασία, αναφέρεται στα διάφορα στάδια σχεδιασμού και ανάπτυξης μίας web εφαρμογής διαχείρισης του Μαθητικού Διαδικτυακού Ραδιοφώνου, ενώ παράλληλα γίνεται αναφορά στα διάφορα προβλήματα που υπήρξαν καθώς και στις ενέργειες που οδήγησαν στην επίλυσή τους.

Το Μαθητικό Διαδικτυακό Ραδιόφωνο, αφορά μία συνεχώς αναπτυσσόμενη κοινότητα σχολείων, εκπαιδευτικών και μαθητών, πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης, που είτε μεμονωμένα είτε μέσω συνεργασιών, δημιουργούν ραδιοφωνικές παραγωγές τις οποίες στην συνέχεια αναρτούν προς μετάδοση μέσω διαδικτύου.

Η εφαρμογή διαχείρισης, που αναπτύχθηκε στα πλαίσια αυτής της Πτυχιακής Εργασίας, έρχεται να αντικαταστήσει το ήδη υπάρχον, πλέον μη λειτουργικό, σύστημα διαχείρισης, και να κάνει την διαχείριση των συμμετεχόντων αλλά και των ραδιοφωνικών παραγωγών τους πολύ πιο εύκολη και αποδοτική, χρησιμοποιώντας νέες τεχνολογίες και επιτρέποντας την αποτελεσματικότερη χρήση των διαθέσιμων υπολογιστικών πόρων.

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω θερμά τους κυρίους Γενιτζέ Νότη και Γιαγκούλη Νίκο, υπεύθυνους διαχείρισης του *European School Radio,* για την άψογη συνεργασία τους και την απρόσκοπτη παροχή διευκρινήσεων, αναφορικά με τις απαιτήσεις λειτουργίας της εφαρμογής διαχείρισης.

Ακόμη θα ήθελα να ευχαριστήσω τον κύριο Αναστάσιο Φιλέλη, υπεύθυνο του Κέντρου Διαχείρισης Δικτύων του Α.Τ.Ε.Ι.Θ. για την πολύτιμη βοήθεια του, αναφορικά με την ανάπτυξη και υποστήριξη διαφόρων υπηρεσιών, απαραίτητων για την σωστή λειτουργία της εφαρμογής διαχείρισης.

Περιεχόμενα

Κεφάλαιο 1 - Εισαγωγή

Στα πλαίσια αυτής της Πτυχιακής Εργασίας, αναπτύχθηκε μία διαδικτυακή (web based) εφαρμογή, κεντρικής και ενοποιημένης (centralized - unified) διαχείρισης και ταυτόχρονης παρουσίασης, των συμμετεχόντων μερών αλλά και του υλικού (ραδιοφωνικών παραγωγών, σχετικών banners κ.λπ.) που μεταδίδεται, στο Μαθητικό Διαδικτυακό Ραδιόφωνο.

Σε αυτό το κεφάλαιο, παρουσιάζεται το Μαθητικό Διαδικτυακό Ραδιόφωνο και η λειτουργία του, καθώς και το πως αυτό διαφοροποιείται από λοιπές δράσεις διαδικτυακού ραδιοφώνου. Ταυτόχρονα, περιγράφονται οι απαιτήσεις της εφαρμογής διαχείρισης, που απορρέουν από την λειτουργία του Μαθητικού Διαδικτυακού Ραδιοφώνου και τις μέχρι σήμερα αναγνωρισμένες αλλά και μελλοντικές ανάγκες.

Στην συνέχεια, στο δεύτερο κεφάλαιο, γίνεται λόγος για ήδη υπάρχουσες λύσεις διαχείρισης, τα πλεονεκτήματα και τα μειονεκτήματα τους, καθώς και τα αίτια που οδήγησαν στην ανάπτυξη μίας εξ ολοκλήρου νέας εφαρμογής.

Ακολούθως, στο τρίτο κεφάλαιο, αναφέρονται εκτένως όλα τα στάδια σχεδιασμού και ανάπτυξης της εφαρμογής διαχείρισης, τα προβλήματα που προέκυψαν καθώς και οι ενέργειες που οδήγησαν στην επίλυσή τους. Παράλληλα παρουσιάζεται σε βάθος το λογισμικό και τα εργαλεία που χρησιμοποιήθηκαν κατά την φάση σχεδιασμού, ανάπτυξης αλλά και για την λειτουργική υποστήριξη της εφαρμογής.

Τέλος, στο τέταρτο κεφάλαιο, υπάρχει μία συνοπτική πρόταση για μελλοντικές επεκτάσεις της εφαρμογής.

Μαθητικό Διαδικτυακό Ραδιόφωνο

Το Μαθητικό Διαδικτυακό Ραδιόφωνο, με ονομασία, *European School Radio*, είναι το πρώτο και μοναδικό διαδικτυακό ραδιόφωνο στην Ελλάδα που λειτουργεί συνεχώς (24x7) και ξεκίνησε την λειτουργία του την Άνοιξη του 2010.

Τι είναι

Αφορά ραδιοφωνικές παραγωγές, ηχογραφήμενες ή ζωντανές (live), που πραγματοποιούνται από μαθητές πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης, και μέσω του εκάστοτε υπεύθυνου δασκάλου ή καθηγητή αναρτώνται προς δημοσίευση και μετάδοση μέσω διαδικτύου.

Ποιος ο σκοπός του

Το *European School Radio*, έχει ως γενικό σκοπό την δημιουργία ενός ενοποιημένου δικτύου σχολείων, δημόσιων και ιδιωτικών, κάθε τύπου της πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης, ώστε απο κοινού να παράγουν ραδιοφωνικό πρόγραμμα.

Οι ραδιοφωνικές παραγωγές έχουν χαρακτήρα ταυτόχρονα ψυχαγωγικό, ενημερωτικό αλλά και εκπαιδευτικό.

Ποια είναι η διοικητική του δομή

Το *European School Radio* είναι μία ενέργεια που βρίσκεται υπό την αιγίδα του Υπουργείου Πολιτισμού, Παιδείας & Θρησκευμάτων. Υποστηρίζεται από το Πανελλήνιο Σχολικό Δίκτυο (Π.Σ.Δ.) και το Αλεξάνδρειο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Θεσσαλονίκης (Α.Τ.Ε.Ι.Θ.) των οποίων και τις υπηρεσίες χρησιμοποιεί (VoD, ιστολόγιο (blog), e-mail, υποδομή δικτύου).

Η συντονιστική ομάδα, που αποτελείται από καθηγητές αλλά και μαθητές, είναι υπεύθυνη για την εύρυθμη λειτουργία του ραδιοφώνου αλλά και για την διασφάλιση της ποιότητας του περιεχομένου των παραγωγών που αναρτούνται προς μετάδοση.

Απαιτήσεις εφαρμογής διαχείρισης

Ο συνεχώς αυξανόμενος αριθμός συμμετοχών, η προσέλευση σχολείων και μαθητών από χώρες του εξωτερικού καθώς και η κατακόρυφη αύξηση του αριθμού των ραδιοφωνικών παραγωγών που αναρτούνταν προς μετάδοση, οδήγησαν το ήδη υπάρχον σύστημα διαχείρισης σε κορεσμό.

Επιπλέον, ορισμένες λειτουργίες, όπως για παράδειγμα, ο προγραμματισμός μετάδοσης των ραδιοφωνικών παραγωγών, επαφίονταν στην επιλογή του τελικού χρήστη (υπεύθυνου δασκάλου/καθηγητή), ενώ στην συνέχεια ήταν απαραίτητος ο έλεγχος και επιβεβαίωση από την συντονιστική ομάδα και σε ορισμένες περιπτώσεις η ανάγκη επίλυσης συγκρούσεων στις επιλεγμένες ημέρες και ώρες. Η όλη διαδικασία ήταν αργή, χωρίς ευελιξία και ευάλωτη σε σφάλματα.

Τέλος, το υπάρχον σύστημα διαχείρισης, αποτελείται από διασκορπισμένα επιμέρους συστήματα, διαφορετικών τεχνολογιών, δυσχαιρένοντας την ομαλή λειτουργία του *European School Radio*, ενώ ταυτόχρονα καθιστά την συνολική διαχείριση, συντήρηση και επέκταση του συστήματος πολύ δύσκολη και απαγορευτικά χρονοβόρα.

Με γνώμονα τις παραπάνω δυσκολίες, αλλά και τις υπάρχουσες και μελλοντικές ανάγκες που απορρέουν από την λειτουργία του *European School Radio* κατά τα

τελευταία πέντε (5) χρόνια, ξεκίνησε η ανάπτυξη της τρέχουσας εφαρμογής διαχείρισης με τα ακόλουθα χαρακτηριστικά, που παρουσιάζονται ομαδοποιημένα ανά τομείς.

Διαχείριση

Ενοποίηση διαχειριστικών υποσυστημάτων

Η πρώτη και κύρια απαίτηση ήταν η πλήρης ενοποίηση όλων των ήδη υπαρχόντων συστημάτων σε μία εφαρμογή. Έτσι η διαχείριση γίνεται πιο εύκολη, ταχύτερη και ασφαλέστερη ενώ ταυτόχρονα πολλές λειτουργίες αυτοματοποιούνται. Άμεσο αποτέλεσμα αυτής της ενοποιήσης, είναι και η δυνατότητα εύκολης συντήρησης, αλλά και επέκτασης της εφαρμογής.

Λογαριασμοί συμμετεχόντων

Ο συνεχόμενα αυξανόμενος αριθμός συμμετεχόντων μελών, στο *European School Radio*, απαιτούσε την σωστή και αποδοτική διαχείριση των λογαριασμών τους.

Επιπλέον, οι διαφορετικοί τύποι λογαριασμών (σχολείο, μαθητής, εξωτερικός συνεργάτης), με διαφορετικά στοιχεία έκαστος, θα έπρεπε να ενοποιηθούν σε μία εύκολα διαχειρίσιμη και επεκτάσιμη δομή.

Τέλος, πρέπει να υπάρχει δυνατότητα αναζήτησης βάσει διαφόρων στοιχείων, αλλά και η επιλογή παύσης ή τερματισμού ενός λογαριασμού ή συγκεκριμένου χρήστη ενός λογαριασμού σε περιπτώσεις μη συμμόρφωσης με τα απαιτούμενα επίπεδα ποιότητας του *European School Radio*, αποτρέποντας με αυτό τον τρόπο την είσοδο και την χρήση της εφαρμογής.

Παραμετροποιήσιμη λειτουργία ραδιοφώνου

Όλοι οι τομείς που αφορούν την λειτουργία του ραδιοφώνου θα έπρεπε να είναι πλέον πλήρως παραμετροποιήσιμοι. Οι διαθέσιμοι τύποι παραγωγών, οι ζώνες του προγράμματος μετάδοσης καθώς και οι θεματικές ενότητες διαμορφώνονται ως πλήρως παραμετροποιήσιμες από τους διαχειριστές της εφαρμογής δηλαδή την συντονιστική ομάδα ή άλλους χρήστες με τα κατάλληλα δικαιώματα χρήσης.

Ασφάλεια - Δικαιώματα χρήσης & ομάδες χρηστών

Στα πλαίσια της ασφαλούς χρήσης της εφαρμογής και του διαχωρισμού των ενεργειών στις οποίες επιτρέπεται να προβούν οι χρήστες, προστέθηκε η δυνατότητα ορισμού δικαιωμάτων χρήσης (permissions).

Επιπλέον υπάρχουν προρυθμισμένες αλλά ταυτόχρονα πλήρως παραμετροποιήσιμες ομάδες χρηστών. Οι ομάδες χρηστών, ουσιαστικά συγκεντρώνουν χρήστες με παρόμοια χαρακτηριστικά ή με παρόμοιες απαιτήσεις και δικαιώματα χρήσης.

Ραδιοφωνικές παραγωγές

Ο κύριος κορμός και λόγος ύπαρξης του *European School Radio* είναι οι παραγωγές που αναρτούνται από τα συμμετέχοντα μέλη. Αντίστοιχα, ένα μεγάλο μέρος των απαιτήσεων της νέας εφαρμογής διαχείρισης, αφορούσε την δήλωση, προγραμματισμό και μεταφόρτωση των παραγωγών αλλά και την συνεργασία μεταξύ συμμετεχόντων για την δημιουργία συνεργατικών παραγωγών.

Συνεργασίες

Για κάθε λογαριασμό, ο διαχειριστής του (υπεύθυνος εκπαιδευτικός, μαθητής, εξωτερικός συνεργάτης), έχει τη δυνατότητα να δηλώσει την πρόθεση συνεργασίας πάνω σε συγκεκριμένες θεματικές ενότητες.

Όταν χρήστες, άλλων λογαριασμών, προχωρούν στην δήλωση των παραγωγών τους, έχουν δυνατότητα, για τις θεματικές ενότητες που καλύπτει η εκάστοτε παραγωγή, να επιλέξουν κάποιον άλλο λογαριασμό (ή πολλαπλούς λογαριασμούς) και να αιτηθούν την σύναψη συνεργασίας. Κατά την διάρκεια της μεταφόρτωσης μίας παραγωγής, κι εφόσον η αίτηση σύναψης συνεργασίας έχει γίνει δεκτή, οι χρήστες μπορούννα προχωρήσουν στην ανάρτηση συνεργατικών εκπομπών.

Δήλωση & προγραμματισμός

Η λειτουργική δομή του ραδιοφώνου, οδήγησε στην δημιουργία πολλών διαφορετικών τύπων παραγωγών. Μπορούμε να συναντήσουμε εκπομπές τακτικές ή έκτακτες, σύντομα σποτ για την προώθηση μίας επερχόμενης παραγωγής, ολιγόλεπτες εκπομπές, σχετικές με συγκεκριμένες θεματικές ενότητες και άλλα.

Οι τακτικές και ορισμένοι άλλοι τύποι παραγωγών απαιτούν προηγούμενη δήλωση και προγραμματισμό. Οι χρήστες της εφαρμογής, δηλώνουν τα στοιχεία της παραγωγής, όπως για παράδειγμα, τον τίτλο και την περιγραφή της, και στην συνέχεια, μεταξύ άλλων, επιλέγουν την ζώνη όπου θα αναρτηθεί.

Βάσει της επιλεγμένης ζώνης, παρουσιάζεται ένα εβδομαδιαίο πρόγραμμα, απ’ όπου οι χρήστες καλούνται να επιλέξουν την επιθυμητή ημέρα και ώρα μετάδοσης.

Η όλη διαδικασία του προγραμματισμού των παραγωγών, αυτοματοποιείται, και η οποιαδήποτε επέμβαση των διαχειριστών της εφαρμογής είναι πλέον περιττή.

Μεταφόρτωση (upload)

Η μεταφόρτωση των αρχείων των παραγωγών, είτε πρόκειται για αρχεία ήχου, εικόνες ή συνδυασμό διαφορετικών τύπων αρχείων, γίνεται μέσω μίας κοινής διεπαφής, ελαχιστοποιώντας την όποια σύγχυση των χρηστών της εφαρμογής.

Παράλληλα, το μέγεθος των αρχείων, που σε ορισμένες περιπτώσεις ανέρχεται σε αρκετές δεκάδες megabyte (MB), καθιστά την ένδειξη προόδου (progress bar) της

μεταφόρτωσης υποχρεωτική, αφού σε διαφορετική περίπτωση οι χρήστες δεν έχουν κανένα στοιχείο αναφορικά με την πορεία της μεταφόρτωσης των επιλεγμένων αρχείων.

Επιπλέον, όλοι οι τύποι παραγωγών, έχουν προκαθορισμένη μέγιστη διάρκεια, και τα αρχεία ήχου που μεταφορτώνονται, ελέγχονται ως προς την διάρκειά τους.

Ιστορικό παραγωγών (podcast)

Όλες οι παραγωγές που μεταφορτώνονται στην εφαρμογή διαχείρισης, θα πρέπει να προβάλονται στους επισκέπτες της εφαρμογής, σε ημερολογιακή διάταξη συμπεριλαμβανομένων όλων των στοιχείων τους (τίτλος, περιγραφή, banner κ.λπ.). Ταυτόχρονα, θα πρέπει να υπάρχει δυνατότητα αναπαραγωγής του εκάστοτε αρχείου ήχου (αφορά υποστηριζόμενους φυλλομετρητές), αλλά και δυνατότητα λήψης (download) του αρχείου.

Επιπλέον στο ιστορικό παραγωγών, θα πρέπει να υπάρχει ένα σύνολο από φίλτρα αναζητήσης που θα επιτρέπουν στον εκάστοτε χρήστη την εύρεση παραγωγών βάσει συγκεκριμένων κριτηρίων.

Αρχείο

Το σύνολο των παραγωγών που μεταφορτώνονται σε έναν συγκεκριμένο λογαριασμό (σχολείο, μαθητής ή εξωτερικός συνεργάτης), διατηρούνται στο αρχείο, επιτρέποντας στους χρήστες του λογαριασμού, την προβολή, αναπαραγωγή και λήψη (download) των παραγωγών τους.

Το αρχείο διαφοροποιείται από το ιστορικό παραγωγών (podcast), ως προς την περιορισμένη οπτική που προσφέρει, αφού αφορά αποκλειστικά αρχεία που έχουν μεταφορτώθει από χρήστες ενός συγκεκριμένου λογαριασμού και όχι τα αρχεία παραγωγών όλης την εφαρμογής.

Πολύγλωσση διεπαφή

Η συμμετοχή σχολείων και μαθητών από χώρες του εξωτερικού, απαιτεί την μετάφραση της διεπαφής χρήσης (UI) σε διάφορες γλώσσες.

Η εφαρμογή είναι δομημένη, στο σύνολό της, με τέτοιο τρόπο, ώστε όλα τα κείμενα, να διατηρούνται σε ξεχωριστά αρχεία, ομαδοποιημένα κατά γλώσσα, επιτρέποντας με αυτό τον τρόπο, την άμεση, γρήγορη και εύκολη μετάφραση.

Ενοποίηση με WordPress & ανάκτηση περιεχομένου

Ένα από τα μέρη του ήδη υπάρχοντος συστήματος διαχείρισης, είναι κι ένα ιστολόγιο (blog), βασισμένο στην πλατφόρμα WordPress, όπου αναρτώνται νέα και ανακοινώσεις, ενώ παράλληλα υπάρχει αρκετό υλικό βοήθειας, όρων και κανόνων που διέπουν την λειτουργιά του *European School Radio*.

Στην νέα εφαρμογή διαχείρισης, στο δημόσια προσβάσιμο τμήμα, υπάρχουν ορισμένες ενότητες, των οποίων το περιεχόμενο συμπληρώνεται αυτόματα από τα αντίστοιχα άρθρα ή και σελίδες που έχουν αναρτηθεί στο ιστολόγιο.

Κεφάλαιο 2 – Υπάρχουσες επιλογές

Η απουσία λύσεων που πραγματεύονται την διαχείριση ραδιοφωνικού προγράμματος θα ήταν αδύνατη. Υπάρχουν διάφορες εφαρμογές, είτε ανοικτού κώδικα (open source) ή διαθέσιμες επί πληρωμής, με περισσότερες ή λιγότερες και γενικές ή πιο εξειδικευμένες δυνατότητες.

Η πιο γνωστή εφαρμογή ανοικτού κώδικα (Creative Commons 3.0), που κάλυπτε το μεγαλύτερο μέρος των απαιτήσεων του *European School Radio* ήταν το AirTime

Airtime

Το AirTime είναι μία web based εφαρμογή, μέσω της οποίας ο διαχειριστής μπορεί να ορίσει άλλους χρήστες οι οποίοι θα ενεργούν ως DJs και θα διαχειρίζονται με την σειρά τους, το ραδιοφωνικό πρόγραμμα και τις παραγωγές.

Δυστυχώς, ενώ το AirTime είναι μία από τις καλύτερες εφαρμογές στον τομέα διαχείρισης ραδιοφωνικού προγράμματος, δεν είναι αρκετά ευέλικτο, και έχει κάποιους περιορισμούς που κατέστησαν την χρήση του αδύνατη.

Συγκεκριμένα, το μοντέλο χρηστών και των δικαιωμάτων που κατέχουν, δεν είναι καθόλου παραμετροποιήσιμο με αποτέλεσμα, να είναι πρακτικά αδύνατος ο περιορισμός και έλεγχος του ραδιοφωνικού προγράμματος που θα αναρτούνταν και οι ενέργειες των χρηστών. Παράλληλα, άπειροι χρήστες, θα είχαν δυνατότητα να προξενήσουν οποιουδήποτε τύπου βλάβη στο πρόγραμμα, προχωρώντας για παράδειγμα σε κάποια ακούσια διαγραφή παραγωγών, δεδομένου πως μετά την χορήγηση πρόσβασης στην εφαρμογή, όλοι έχουν πρόσβαση σε ένα κοινό χώρο διαχείρισης και δυνατότητα προβολής και επεξεργασίας όλων των παραγωγών. Αυτή η συμπεριφορά του AirTime είναι αντιδιαμετρικά αντίθετη με το μοντέλο χρήσης που εφαρμόζει το *European School Radio*, όπου ο κάθε χρήστης είναι υπεύθυνος αποκλειστικά για τις παραγωγές του λογαριασμού στον οποίο ανήκει, και δεν έχει καμία δικαιοδοσία επί των παραγωγών των άλλων μελών.

Επιπλέον, το *European School Radio*, είχε επιλέξει ως μέσο μετάδοσης του ραδιοφωνικού προγραμμάτος, την εφαρμογή Jazler, η οποία δεν υποστηρίζεται από το AirTime.

Τέλος, ένας ακόμη λόγος για τον οποίο η χρήση του AirTime κατέστη αδύνατη, ήταν η επίσης αποκλειστική υποστήριξη για λειτουργικά συστήματα Debian και Ubuntu. Η υποδομή που προϋπήρχε και φιλοξενούσε την ήδη υπάρχουσα εφαρμογή διαχείρισης, λειτουργούσε με χρήση του λειτουργικού συστήματος Windows, ενώ η χρήση επιπλέον εξοπλισμού ήταν αδύνατη λόγω κόστους. Ταυτόχρονα, η λύση ενός

εξυπηρετητή, με λειτουργικό σύστημα Debian ή Ubuntu, όπου θα φιλοξενούσε παράλληλα και το λειτουργικό σύστημα Windows, μέσω κάποιας εφαρμογής εικονικής μηχανής (VM), απορρίφθηκε απευθείας, καθώς μία τέτοια υλοποίηση, θα δημιουργούσε αρκετά προβλήματα απόδοσης.

Κεφάλαιο 3 – Εφαρμογή διαχείρισης

Η εφαρμογή διαχείρισης αποτελείται από τρία διακριτά μέρη, το δημόσιο (default), το προφίλ (profile) και την διαχείριση (administration).

Το δημόσιο τμήμα (default), είναι προσβάσιμο από όλους τους επισκέπτες, και περιέχει ουσιαστικά την παρουσίαση του *European School Radio*, το σύνολο των συμμετεχόντων μερών, το ιστορικό των παραγωγών (podcast) και άρθρα βοήθειας, ενώ παράλληλα ενσωματώνει την διεπαφή της λειτουργίας εισόδου - πιστοποίησης (login).

Το προφίλ (profile), είναι ιδιωτικό τμήμα, προσβάσιμο αποκλειστικά από τα μέλη του εκάστοτε λογαριασμού και είναι πλήρως προσαρμοσμένο στις ενέργειες του εκάστοτε λογαριασμού. Μέσω της διεπαφής του προφίλ οι χρήστες του εκάστοτε λογαριασμού έχουν δυνατότητα δήλωσης, προγραμματισμού και μεταφόρτωσης των παραγωγών τους, ενώ παράλληλα έχουν και την πλήρη διαχείριση των στοιχείων του λογαριασμού τους.

Τέλος, το τμήμα της διαχείρισης (administration) επιτρέπει στους χρήστες, που κατέχουν τα ανάλογα δικαιώματα, να προχωρούν στην πλήρη διαχείριση του ραδιοφώνου, καθοριζόντας τους τύπους των διαθέσιμων παραγωγών, τις ζώνες του προγραμμάτος και τις διαθέσιμες θεματικές ενότητες, ενώ ταυτόχρονα υπάρχει δυνατότητα για πλήρη διαχείριση των λογαριασμών των συμμετεχόντων μερών.

Το σύνολο της εφαρμογής, έχει αναπτυχθεί βασισμένο στην αρχιτεκτονική MVC (Model - View - Controller), με χρήση της γλώσσας προγραμματισμού PHP σε συνδυασμό με το framework CodeIgniter και της βάσης δεδομένων MySQL, ενώ παράλληλα έχουν χρησιμοποιηθεί και άλλες τεχνολογίες, που αφορούν κατά κύριο λόγο την προβολή της εφαρμογής, όπως HTML, CSS και το javascript framework jQuery.

Αποθήκευση δεδομένων - MySQL & XML

Για την αποθήκευση του μεγαλύτερου συνόλου των δεδομένων της εφαρμογής διαχείρισης, χρησιμοποιήθηκε το RDBMS, MySQL.

Η επιλογή της MySQL ως backend data storage, ήταν ουσιαστικά μονόδρομος, αφού είναι ένα από τα ελάχιστα συστήματα διαχείρισης βάσεων δεδομένων το οποίο διατίθεται δωρεάν, ενώ ταυτόχρονα δεν υπολείπεται σε λειτουργικότητα, ακόμη και των πιο ακριβών RDBMSes, υλοποιεί το μεγαλύτερο μέρος του προτύπου SQL, είναι γρήγορη, ασφαλής, με πολύ μεγάλη κοινότητα χρηστών, και τέλος προσφέρει άριστη επεκτασιμότητα (scalability), επιτρέποντας την χρήση της σε σύνολα δεδομένων, οποιουδήποτε μεγέθους.

Σε ένα μικρό υποσύνολο των δεδομένων, το οποίο αφορούσε τις ρυθμίσεις της εφαρμογής (application settings), χρησιμοποιήθηκαν αρχεία XML. Η ανάπτυξη μίας διεπαφής για την διαχείριση των διαθέσιμων ρυθμίσεων, ήταν πολύπλοκη, η χρήση της θα ήταν δύσκολη ακόμη και για έμπειρους χρήστες, και οι ελάχιστες αλλαγές που αφορούν το σύνολο των διάθεσιμων ρυθμίσεων, δεν θα μπορούσαν να δικαιολογήσουν κάτι τέτοιο. Ταυτόχρονα, σε περιπτώσεις όπου απαιτούνται αλλαγές, κρίθηκε πως η επεξεργασία ενός αρχείου XML, είναι πολύ πιο εύκολη, συγκριτικά με την διαχείριση μίας βάσης δεδομένων, είτε αυτή γίνεται μέσω γραμμής εντολών είτε μέσω γραφικού περιβάλλοντος (π.χ. phpMyAdmin)

CAS – Central Authentication Service

Το CAS είναι μία υπηρεσία πιστοποίησης χρηστών και χρησιμοποιείται σε συνδυασμό με κάποια άλλη υπηρεσία διατήρησης στοιχείων ταυτοποίησης χρηστών, όπως LDAP, Active Directory, Kerberos ή κάποια βάση δεδομένων. Βασίζεται στην αρχιτεκτονική server – client, έχει αναπτυχθεί με χρήση Java και συγκεκριμένα με την χρήση του framework Spring, ενώ ο client έχει υλοποιηθεί για διάφορα προγραμματιστικά περιβάλλοντα, συμπεριλαμβανομένων των Java, PHP, .NET, αλλά και ως module του εξυπηρετητή (server) HTTP Apache (mod_auth_cas).

Επιπλέον, η υπηρεσία CAS, προσφέρει λειτουργία Single Sign On (SSO). Η λειτουργία SSO, επιτρέπει σε έναν χρήστη την πρόσβαση σε διάφορες εφαρμογές, που κάνουν χρήση της υπηρεσίας CAS, έχοντας πιστοποιηθεί ωστόσο μόνο μία φορά. Κάτι τέτοιο είναι εφικτό, δεδομένου ότι το κομμάτι της πιστοποίησης των χρηστών, αποσπάται από το πλαίσιο μίας συγκεκριμένης εφαρμογής και λειτουργεί ως ανεξάρτητη υπηρεσία.

Μηχανισμός λειτουργίας CAS

Η αίτηση ενός χρήστη, για πρόσβαση σε κάποια προστατευόμενη περιοχή ή γενικότερα για την προβολή και χρήση ενός προστατευόμενου πόρου, στην εφαρμογή Α, ανακατευθύνεται (redirect) προς την υπηρεσία πιστοποίησης CAS, ενώ το αυτόματα δημιουργούμενο URL, περιέχει και τα στοιχεία της αιτούμενης εφαρμογής, δηλαδή στην προκειμένη περίπτωση, την εφαρμογή Α. Το τελικό URL ανακατεύθυνσης από την εφαρμογή Α, προς την υπηρεσία πιστοποίησης CAS, έχει συνήθως, την ακόλουθη μορφή:

```
http://cas_server/cas/login?service=http://other_server/applicationA
```

Όταν η υπηρεσία CAS (cas_server) λάβει αυτό το αίτημα, ενεργοποιείται η λειτουργία πιστοποίησης, είτε προτρέποντας τον χρήστη να εισάγει τα στοιχεία σύνδεσης του (όνομα χρήστη και κωδικός πρόσβασης) είτε ελέγχοντας αν υπάρχει ήδη

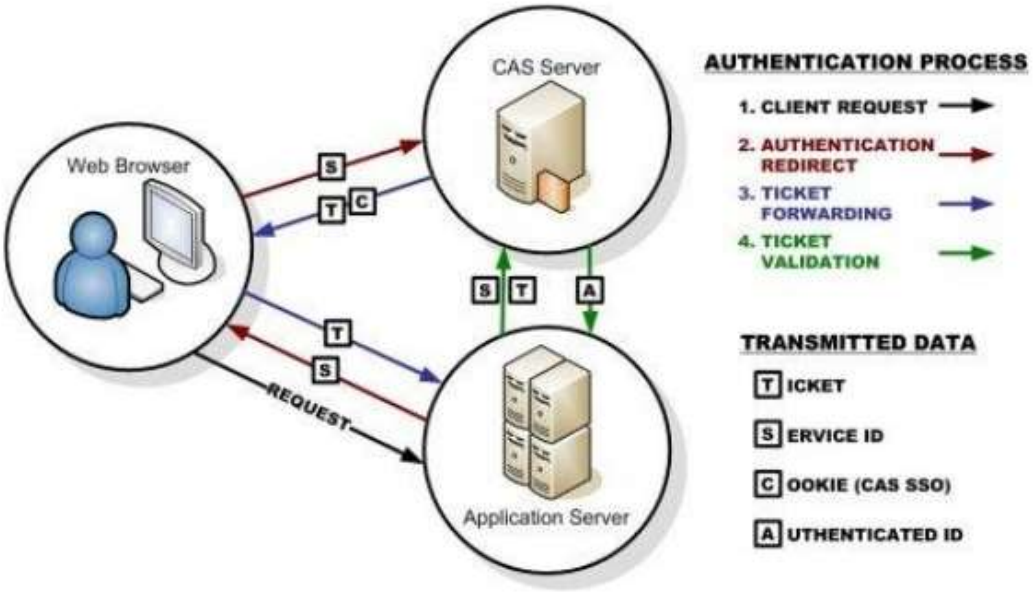
κάποια ενεργή σύνοδος SSO (Single Sign-On session). Αν η πιστοποίηση είναι επιτυχής, επιβεβαιώνεται πως επιτρέπεται η χρήση της υπηρεσίας CAS, από την εφαρμογή A, με έλεγχο στο σύνολο των εγγεγραμένων εφαρμογών (registered services). Αν η επιβεβαίωση ολοκληρωθεί, επίσης επιτυχώς, η υπηρεσία CAS εκίδει ένα service ticket (ST), το οποίο και προσθέτει ως παράμετρο στο query string, μίας νέας αίτησης (URL) που δημιουργείται αυτόματα (ανακατεύθυνση - redirection) προς την εφαρμογή A και έχει την ακόλουθη μορφή:

```
http://other_server/applicationA?ticket=ST-8670-123buTvFFjo980
```

Η εφαρμογή A, λαμβάνει μέσω του πελάτη (client) CAS, το ticket που έχει στείλει η υπηρεσία CAS, και εκκινεί μία νέα ασφαλή σύνδεση (HTTPS) προς την υπηρεσία CAS. Η νέα αυτή σύνδεση γίνεται αυτόματα από τον πελάτη (client) CAS της εφαρμογής A, και ονομάζεται “service validation” ή “ticket validation”.

Η υπηρεσία CAS, λαμβάνει και επιβεβαίωνει, την νέα ασφαλή σύνδεση μεταξύ των δύο server (εφαρμογή A & υπηρεσία CAS), και στη συνέχεια εξυπηρετεί το αίτημα του πελάτη CAS της εφαρμογής A, επιστρέφοντας ως απάντηση (response), ένα αρχείο XML, το οποίο περιέχει έναν κωδικό κατάστασης (status code) και το username του πιστοποιημένου χρήστη.

Όλη η διαδικασία, γίνεται αδιάφανα, χωρίς την παρέμβαση του χρήστη, ο οποίος ουσιαστικά έκανε μόνο μία αίτηση για προβολή και χρήση κάποιου προστατευόμενου πόρου της εφαρμογής A. Η ολοκλήρωση της διαδικασίας επικοινωνίας με την υπηρεσία CAS, δηλώνει την επιτυχή πιστοποίηση του χρήστη, άρα και την δυνατότητα πρόσβασης σε προστατευόμενους πόρους ή ιδιωτικές περιοχές της εφαρμογής.



Εικόνα 1 - Ροή λειτουργίας CAS

Χρήση CAS στην εφαρμογή διαχείρισης

Για την εφαρμογή διαχείρισης, που αναπτύχθηκε στα πλαίσια αυτής της πτυχιακής εργασίας, αρχικός στόχος ήταν η χρήση της υπηρεσίας CAS που παρέχει το Πανελλήνιο Σχολικό Δίκτυο, παράλληλα με μία ακόμη υλοποίηση της υπηρεσίας CAS η οποία θα παρεχόταν από το *European School Radio* και θα είχε σκοπό να εξυπηρετήσει τους λογαριασμούς χρηστών εκτός Ελλάδας και τους χρήστες που δεν διέθεταν λογαριασμό στο Πανελλήνιο Σχολικό Δίκτυο (μαθητές και εξωτερικοί συνεργάτες).

Η υλοποίηση της υπηρεσίας CAS του *European School Radio*, ξεκίνησε με χρήση του πακέτου JASIG CAS, του HTTP server Apache Tomcat, και της εφαρμογής διαχείρισης ανάπτυξης Maven.

Ένα από τα αρχικά προβλήματα που ανέκυψαν ήταν η αδυναμία λειτουργίας της υπηρεσίας CAS χωρίς την ύπαρξη ενός έγκυρου πιστοποιητικού SSL. Το πιστοποιητικό SSL είναι απαραίτητο κατά το τελευταίο στάδιο της διαδικασίας πιστοποίησης, όπου ο εξυπηρετητής της εφαρμογής επικοινωνεί μέσω ασφαλούς σύνδεσης (HTTPS), με τον εξυπηρετητή της υπηρεσίας CAS, για την επαλήθευση του ST (Service Ticket). Το πιστοποιητικό εκδόθηκε από την Αρχή Πιστοποίησης του Αλεξάνδρειου ΤΕΙ Θεσσαλονίκης, και η χρήση του οδήγησε στην επίλυση του προβλήματος και την συνέχιση της υλοποίησης της εφαρμογής.

Κατά την διάρκεια της ανάπτυξης προέκυψαν επίσης διάφορα προβλήματα που αφορούσαν την μη ορθή χρήση της εφαρμογής Maven, και πιο συγκεκριμένα τα αρχεία

POM (Project Object Model). Τα αρχεία POM είναι μορφής XML, και περιέχουν πληροφορίες και ρυθμίσεις σχετικά με την κατασκευή (build) μίας εφαρμογής.

Το αποτέλεσμα της λανθασμένης δομής των αρχείων POM ήταν, το τελικό πακέτο που θα γινόταν deploy στον server Tomcat, να μην περιέχει σωστά αρχεία ρυθμίσεων (configuration) ή σε ορισμένες περιπτώσεις να απουσιάζουν κάποιες βιβλιοθήκες εξάρτησης (dependencies). Τέτοιου είδους προβλήματα επιλύθηκαν πολύ σύντομα με προσεκτικότερη χρήση των διαθέσιμων επιλογών και ρυθμίσεων της εφαρμογής Maven και αναλυτικότερη ανάγνωση της σχετικής τεκμηρίωσης.

Το κυριότερο εμπόδιο στην χρήση της εφαρμογής ήταν η σύνδεση με την βάση δεδομένων της εφαρμογής διαχείρισης απ’ όπου και θα γινόταν η πιστοποίηση των χρηστών. Το πρόβλημα που ανέκυπτε συνεχώς, ήταν η αδυναμία έκδοσης ST από την υπηρεσία CAS, και η περαιτέρω ανάλυση των καταγραφών σφαλμάτων (logs) οδήγησε στο συμπέρασμα, πως η αιτία αυτής της αδύναμιας έκδοσης ST, ήταν η προηγούμενη αδυναμία ανάκτησης δεδομένων από την βάση δεδομένων.

Στην προσπάθεια επίλυσης αυτού του προβλήματος, έγινε πλήρης έλεγχος και αποσφαλμάτωση της σύνδεσης μεταξύ της υπηρεσίας CAS και της βάσης δεδομένων, και συγκεκριμένα του bean `dataSource`, το οποίο ήταν υπεύθυνο για την δημιουργία και διατήρηση ενός pool συνδέσεων, χωρίς κάποιο συγκεκριμένο αποτέλεσμα αφού η σύνδεση μεταξύ των δύο (υπηρεσία CAS και βάση δεδομένων) ολοκληρωνόταν επιτυχώς. Στην συνέχεια, και αφού επιβεβαιώθηκε πως το πρόβλημα δεν επαφίοταν στην σύνδεση μεταξύ υπηρεσίας CAS και βάσης δεδομένων, ο έλεγχος προχώρησε στο επόμενο στοιχείο που χρησιμοποιούνταν κατά την διαδικασία πιστοποίησης.

Το bean `principalResolver` είναι υπεύθυνο για αντιστοίχιση των μοναδικών αναγνωριστικών (IDs) βάσει των οποίων οι πιστοποιημένοι χρήστες, αναγνωρίζονται από τις εφαρμογές που χρησιμοποιούν την υπηρεσία CAS. Επίσης, το bean `principalResolver` περιέχει προαιρετικά χαρακτηριστικά (attributes) για τον κάθε χρήστη (προσωπικά στοιχεία, στοιχεία επικοινωνία, κ.α.) τα οποία η υπηρεσία CAS μπορεί να παραχωρήσει στις εφαρμογές που θα το αιτηθούν, μαζί με το username του πιστοποιημένου χρήστη, που στέλνει ως απάντηση σε κάθε response.

Ως principalResolver χρησιμοποιήθηκε η, παρεχόμενη από τον οργανισμό JASIG, κλάση `PersonDirectoryPrincipalResolver` του πακέτου `org.jasig.cas.authentication.principal`. Δυστυχώς η απουσία καταγραφών σφαλμάτων καθώς και η απούσια βοήθειας σχετικής με το συγκεκριμένο πρόβλημα, οδήγησαν στην αδυναμία επίλυσης αυτού του προβλήματος.

Ταυτόχρονα, η διαχειριστική επιτροπή του *European School Radio*, έκκρινε πως η χρήση της υπηρεσίας CAS δεν προσέφερε κάποιο συγκριτικό πλεονέκτημα έναντι της

τοπικής πιστόποιησης, δεδομένου ότι δεν υπήρχαν πλέον διαφορετικές επιμέρους εφαρμογές, άρα και ανάγκη για παροχή λειτουργικότητας SSO (Single Sign On).

Για αυτούς τους λόγους, η χρήση της υπηρεσίας CAS απορρίφθηκε εξ ολοκλήρου και αντ' αυτής χρησιμοποιήθηκε ένας τοπικός μηχανισμός πιστοποίησης χρηστών, ο οποίος αναλύεται εκτενώς σε ακόλουθη ενότητα.

Αρχιτεκτονική MVC

Το σύνολο της εφαμογής διαχείρισης αναπτύχθηκε ακολουθώντας την αρχιτεκτονική MVC. Η αρχιτεκτονική MVC, χωρίζει μία εφαρμογή σε τρία διακριτά τμήματα, τα οποία ωστόσο μπορούν να επικοινωνούν και να αλληλεπιδρούν μεταξύ τους, με τελικό σκοπό τον διαχωρισμό της λογικής (business logic) μίας εφαρμογής και της διαχείρισης των δεδομένων, από την προβολή τους στον τελικό χρήστη.

Τα τρία τμήματα της αρχιτεκτονικής MVC είναι **M**odel, **V**iew, **C**ontroller και η λειτουργία εκάστου, αναλύεται παρακάτω.

Model

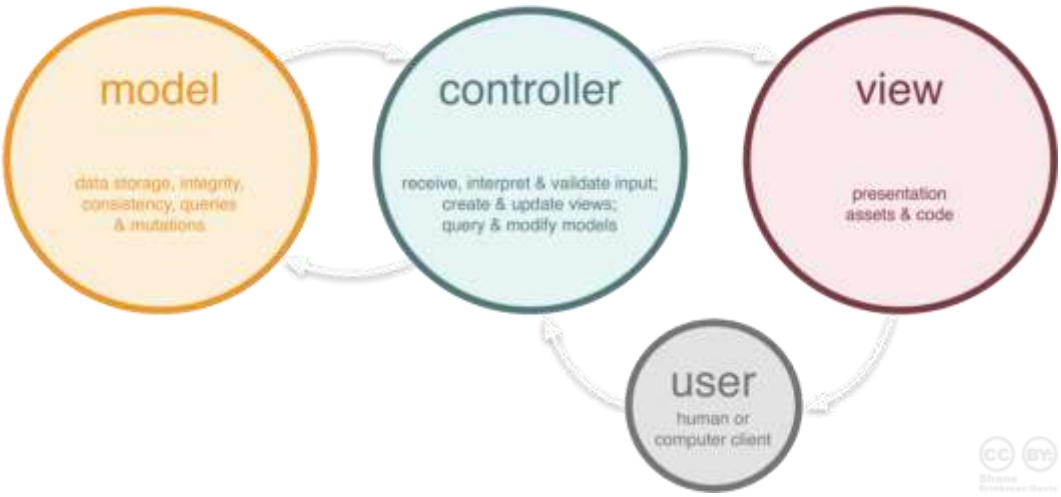
Το model είναι υπεύθυνο για την ανάκτηση και αποθήκευση δεδομένων και ως επί το πλείστον, αλληλεπιδρά με κάποια βάση δεδομένων ή άλλου τύπου data store όπως αρχεία XML ή non-SQL βάσεις δεδομένων.

View

Το view είναι υπεύθυνο για την παρουσίαση των δεδομένων που θα του «παραδώσει» ο controller αφού πρώτα αυτά ανακτηθούν μέσω του model. Επίσης με την χρήση ενός view, που παρουσιάζει κάποιου είδους φόρμα, είναι δυνατή η αίτηση και εκχώρηση δεδομένων από τον χρήστη, η μετέπειτα διαχείρισή τους μέσω του controller και η τελική αποθήκευση αλλαγών ή νέων δεδομένων στο data store που χρησιμοποιείται, μέσω του model.

Controller

Ο controller είναι το τμήμα που διαχειρίζεται την αλληλεπίδραση ανάμεσα στον χρήστη και την εφαρμογή. Είναι υπεύθυνο για την κίνηση δεδομένων από και προς το model και την ενημέρωση του view ούτως ώστε να αντιπροσωπεύει την τρέχουσα κατάσταση του model. Ο controller είναι το μοναδικό τμήμα στο οποίο έχει ουσιαστική πρόσβαση ο χρήστης της εφαρμογής, και είναι το σημείο όπου κωδικοποιείται η λογική (business login) της εφαρμογής.



Εικόνα 2 - Αρχιτεκτονική MVC

CodeIgniter

Το *CodeIgniter* είναι ένα MVC framework το οποίο χρησιμοποιείται για ανάπτυξη διαδικτυακών εφαρμογών σε γλώσσα PHP. Αναπτύχθηκε από την εταιρεία EllisLab και πλέον η συντήρησή του έχει μεταβιβαστεί στο Τεχνολογικό Ινστιτούτο της British Columbia.

Το *CodeIgniter* επιλέχθηκε, καθώς είναι πολύ πιο αποδοτικό έναντι άλλων frameworks παρόμοιας αρχιτεκτονικής (Zend, Yii, Fuel, CakePHP κ.λπ.), καθώς η απλότητα της δομής του, οδήγησε σε πολύ μικρό αντίκτυπο στην συνολική χρήση μνήμης και άλλων πόρων, χαρακτηριστικά που το κατατάσουν σε ένα από τα πιο γρήγορα frameworks.

Επιπλέον το σύνολο των ενσωματωμένων βιβλιοθηκών (libraries) που παρέχει, είναι πολύ “ελαφριές” διατηρώντας πλήρη ισορροπία ανάμεσα στις λειτουργίες που παρέχουν και την χρήση των διαθέσιμων υπολογιστικών πόρων, ενώ ταυτόχρονα δίνεται πολύ μεγάλο βάρος στην συνολική ασφάλεια της αναπτυσόμμενης εφαρμογής με επίσης ενσωματωμένες λειτουργίες XSS filtering, CSRF protection, session forgery κ.λπ. Τέλος, συνοδεύεται από εξαιρετική τεκμηρίωση, που καθιστά την εκμάθησή των βασικών αλλά και πιο εξειδικευμένων λειτουργιών του framework, πολύ πιο εύκολη και γρήγορη συγκριτικά με άλλα frameworks, που παρέχουν ελλειπή ή ακόμη και καθόλου τεκμηρίωση.

Φυσικά, σε ορισμένες περιπτώσεις η απλότητα του framework, οδηγεί σε περιορισμούς, όπως για παράδειγμα η αδυναμία υλοποίησης (implementation) interfaces, λόγω της εσωτερικής δομής και λειτουργίας του framework και της κλάσης του autoloader που δεν επιτρέπει κάτι τέτοιο.

Ενίοτε, για να καμφθούν τέτοιου είδους προβλήματα, απαιτειται η μετατροπή του πυρήνα (core) του framework, διαδικασία που πρέπει να αποφεύγεται καθώς δυσχεραίνει ή καθιστά αδύνατη την μετέπειτα ενημέρωση (update) του framework και ενδέχεται να οδηγήσει σε σοβαρά κενά ασφαλείας ή άλλα σφάλματα, αν το σύνολο των μετατροπών δεν πραγματοποιηθεί από άτομα με βαθύτατες γνώσεις της αρχιτεκτονικής και των αρχών λειτουργίας του framework. Συνηθήστερα βέβαια, προτιμάται η χρήση άλλων εναλλακτικών μηχανισμών όπως βιβλιοθήκες ή επεκτάσεις τρίτων (third party addons) για την αντιμετώπιση οποιουδήποτε περιορισμού και την μετατροπή ή την επέκταση κάποιας συγκεκριμένης λειτουργίας.

Δρομολόγηση αιτήσεων (request routing)

Κάθε αίτηση χρήστη, για την προβολή μίας συγκεκριμένης σελίδας της εφαρμογής ή για την εκτέλεση κάποιας ενέργειας, δρομολογείται εσωτερικά από το framework. Η διαδικασία εσωτερικής δρομολόγησης που πραγματοποιείται αφορά την αντιστοίχιση τμημάτων του URL, του request, σε controllers, functions και ενίοτε παραμέτρους.

Ο `loader`, μία από τις πιο βασικές κλάσεις του framework, αναλαμβάνει στη συνέχεια να αναλύσει περαιτέρω το αποτέλεσμα αυτής της αντιστοίχισης, και να καταλήξει στην φόρτωση συγκεκριμένων κλάσεων και την κλήση συγκεκριμένων συναρτήσεων με τις κατάλληλες παραμέτρους.

Η εσωτερική δρομολόγηση που γίνεται από το CodeIgniter, ακολουθεί το μοτίβο που αναλύεται στην συνέχεια με την χρήση ενός παραδείγματος. Έστω ότι ένας διαχειριστής της εφαρμογής, θέλει να προχωρήσει στην προβολή όλων των λογαριασμών της εφαρμογής. Ακολουθώντας τους διαθέσιμους συνδέσμους του μενού περιήγησης, καταλήγει να οδηγηθεί στην διεύθυνση:

```
http://www.esr.gr/admin/accounts/details/55773246ee62b8-98241676
```

Βάσει του μηχανισμού δρομολόγησης, η παραπάνω διεύθυνση URL, αντιστοιχίζεται στην function `details`, του controller `accounts` που εντοπίζεται στον υποφάκελο `admin`. Όταν θα εκτελεστεί η παραπάνω function, θα έχει ως παράμετρο την τιμή `55773246ee62b8-98241676`. Συνοψίζοντας το παραπάνω παράδειγμα σε ένα πιο γενικό πλαίσιο, καταλήγουμε στο παρακάτω μοτίβο

```
protocol://domain.tld/controller/function/param_1/param_2/.../param_n
```

όπου ορίζεται το χρησιμοποιούμενο πρωτόκολλο σύνδεσης, το οποίο συνηθέστερα είναι το HTTP, ακολουθούμενο από το όνομα χώρου και την κατάληξη (domain.tld).

Στην συνέχεια συναντάμε τη διαδρομή προς το αρχείο του επιθυμητού controller, η οποία αντιστοιχίζεται στο αρχείο με όνομα `controller` που βρίσκεται μέσα στον

φάκελο `application/controllers` του framework. Αν δεν ορίζεται το όνομα αρχείου του επιθυμητού controller απευθείας, αλλά προηγείται μία διαδρομή, αυτή μεταφράζεται ως υποφάκελος του βασικού φακέλου των controllers (`application/cotrollers`).

Μετά την δήλωση του επιθυμητού controller, βρίσκεται το όνομα της function που θα εκτελεστεί, ακολουθούμενο από μία προαιρετική λίστα παραμέτρων. Οι παράμετροι αντιστοιχίζονται στις παραμέτρους της επιλεγμένης function, επομένως αν είχαμε την διεύθυνση URL του προηγούμενου παραδείγματος

```
http://www.esr.gr/admin/accounts/details/55773246ee62b8-98241676
```

και την function `public function details($accountId)`, ορισμένη εντός της κλάσης του controller `accounts`, τότε η μεταβλητή `$accountId` θα είχε την τιμή `55773246ee62b8-98241676`.

Φυσικά στο URL μπορούν να προσαρτηθούν παράμετροι ως μέρος του query string, πλέον των παραμέτρων που έχουν ήδη χρησιμοποιηθεί ή μεμονωμένα, με την χρήση του λατινικού ερωτηματικού (?) ως χαρακτήρα οριοθέτησης (delimiter). Οι παράμετροι στο query string, ακολουθούν την μορφή key/value, και σύμφωνα με το RFC 2616 HTTP 1.1, έχουν μέγιστο μέγεθος 2KiB, αν και ορισμένοι web servers και προγράμματα περιήγησης (browsers), μπορούν να επεξεργασθούν URLs με συνολικό μέγεθος 8KiB.

Σε περιπτώσεις όπου δεν παρέχεται το όνομα ενός controller ή κάποιας συγκεκριμένης function, χρησιμοποιούνται εξ ορισμού τιμές. Συγκεκριμένα για τον controller, χρησιμοποιείται η τιμή `Home` και για την function, η `index`. Η εξ ορισμού τιμή του controller μπορεί να προσαρμοσθεί μέσω της επιλογής `default_controller`, που εντοπίζεται στο αρχείο `routes.php` και το οποίο είναι μέρος των ρυθμίσεων του framework, ενώ η εξ ορισμού τιμή της function δεν είναι παραμετροποιήσιμη.

Οι εξ ορισμού τιμές, ουσιαστικά υποδεικνύουν στον `loader`, την φόρτωση συγκεκριμένης κλάσης και function, όταν δεν υπάρχουν συγκεκριμένες αιτήσεις. Αυτή η λειτουργία ωστόσο δεν επιτρέπει, κανένα είδος «υβριδικής» χρήσης, με την έννοια ότι δεν επιτρέπεται να παραληφθεί το όνομα του controller και στην θέση του να τοποθετηθεί το όνομα της επιθυμητής function ή αντίστοιχα να παραληφθεί το όνομα της function, και στην θέση του, να τοποθετηθεί η λίστα παραμέτρων. Μία τέτοια ενέργεια, θα οδηγούσε πολύ πιθανά σε κρίσιμο σφάλμα, αφού δεν θα ήταν δυνατή η φόρτωση κανενός controller ή και function, με άμεσο αποτέλεσμα την αδυναμία εξυπηρέτησης της αιτούμενης ενέργειας.

Βιβλιοθήκες

Το CodeIgniter, έχει ένα αρκετά μεγάλο αριθμό, ενσωματωμένων βιβλιοθηκών, για την παροχή διάφορων λειτουργιών, όπως την διαχείρισης δεδομένων ή την αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου (email). Μία πλήρης αναφορά όλων των διαθέσιμων βιβλιοθηκών, είναι διαθέσιμη στην τεκμηρίωση του framework.

Πέραν των ενσωματωμένων βιβλιοθηκών, υπάρχει δυνατότητα προσθήκης νέων βιβλιοθηκών, για την περαιτέρω προσαρμογή της λειτουργίας του framework στις ανάγκες της κάθε εφαρμογής και την προσθήκη επιπλέον λειτουργικότητας.

Για την εφαρμογή διαχείρισης που αναπτύχθηκε στα πλαίσια αυτής της πτυχιακής εργασίας, δημιουργήθηκαν διάφορες επιπλέον βιβλιοθήκες οι οποίες και περιγράφονται συνοπτικά παρακάτω.

AJAX

Επωμίζεται την διαχείριση κλήσεων AJAX με έκδοση απαντήσεων (response) σε μορφή JSON. Κάθε response, αποτελείται από ένα status code, boolean μορφής (true/false), και δεδομένα (data) που μπορούν να είναι οποιασδήποτε μορφής (plain, array, object).

Authentication

Είναι η υπεύθυνη βιβλιοθήκη για την πιστοποιήση, σύνδεση και αποσύνδεση των χρηστών (log in και log out), είτε τοπική είτε μέσω CAS του Π.Σ.Δ., και για τον έλεγχο της σύνδεσης ενός χρήστη (isLoggedIn). Επίσης εσωκλείει μία μεθόδο για τον κατακερματισμό (hash) των κωδικών πρόσβασης με την προσθήκη επιπλέον τυχαίων χαρακτήρων (salt), και τέλος μία ακόμη μέθοδο υπεύθυνη για την επαναφορά της συνόδου (session) σε προηγούμενη κατάσταση, σε περιπτώσεις όπου δεν έχει προηγηθεί πιστοποίηση του χρήστη στην τρέχουσα σύνοδο, αλλά είναι ενεργό το cookie της επιλογής απομνημόνευσης σύνδεσης (remember me).

Authorization

Ατοτελεί μία βιβλιοθήκη ελέγχου δικαιωμάτων πρόσβασης βάσει της ομάδας στην οποία ανήκει ο εκάστοτε χρήστης.

Configuration

Η βιβλιοθκη αυτή, χρησιμοποιείται για την ανάκτηση ρυθμίσεων (system settings) και προτιμήσεων των χρηστών (preferences).

Layout

Βασική βιβλιοθήκη διαχείρισης της δομής (layout) των σελίδων της εφαρμογής. Ουσιαστικά είναι μία κλάση περίβλημα (wrapper), που παρακάμπτει των default τρόπο προβολής των views του framework. Η `Layout` διασπά την εκάστοτε σελίδα σε εφτά (7)

επιμέρους τμήματα (header, navbar, main body, footer κ.λπ.), επιτρέποντας την πλήρη προσαρμογή της τελικής δομής της σελίδας, σε οποιαδήποτε ανάγκη.

Επιπλέον, παρέχονται επιλογές για την προσθήκη scripts, stylesheets ή άλλων δεδομένων στην κάθε σελίδα, ενώ είναι επίσης δυνατή η χρήση πολλαπλών views σε μία σελίδα, επιτρέποντας την διάσπαση και επαναχρησιμοποίηση κοινών τμημάτων σε πολλές σελίδες, όπως για παράδειγμα, πλαϊνές μπάρες πλοήγησης ή επιλογών (sidebars).

Notifier

Η `Notifier`, είναι μία βιβλιοθήκη, που επιτρέπει την εμφάνιση μηνυμάτων από έναν controller σε ένα (ή περισσότερα) view. Τα μηνύματα μπορούν να αφορούν γενικές πληροφορίες (info), σφάλματα (errors), προειδοποιήσεις (warnings) ή απλά επιβεβαιώσεις για την ολοκλήρωση κάποιας ενέργειας (success).

Εξ ορισμού, όλα τα μηνύματα, διατηρούνται για την διάρκεια ενός request, και στην συνέχεια διαγράφονται αυτόματα, είτε προβληθούν είτε όχι. Υπάρχει, ωστόσο, δυνατότητα διατήρησης επιλεγμένων μηνυμάτων για ένα επιπλέον request, με χρήση του μηχανισμού της συνόδου σύνδεσης (session).

Uploader

Βιβλιοθήκη που αναλαμβάνει την μεταφόρτωση (upload) αρχείων, ενώ παράλληλα έχει δυνατότητες ελέγχου και περιορισμού των αρχείων, βάση του τύπου τους (έλεγχος επέκτασης αρχείου και τύπου MIME), ενώ παράλληλα μπορεί να εκτελεί διάφορες ενέργειες μετονομασίας, όπως αφαίρεση κενών χαρακτήρων (spaces) από το όνομα του αρχείου, καθώς και κρυπτογράφησης του (MD5 hash). Επιπλέον, η βιβλιοθήκη `Uploader`, υποστηρίζει την μεταφόρτωση πολλαπλών αρχείων.

Util

Η `Util`, αν και αναφερόμενη τελευταία, είναι η δεύτερη πιο χρησιμοποιούμενη βιβλιοθήκη, μετά την `Layout`, αφού παρέχει διάφορες λειτουργίες – εργαλεία, όπως την μετατροπή ημερομηνιών ανάμεσα σε αναγνώσιμη μορφή και σε μορφή SQL, την μετατροπή διάρκειας η οποία είναι εκφρασμένη σε λεπτά, σε μορφή ώρας, έλεγχο δεδομένων για έγκυρη μορφή JSON και άλλα.

Διαχείριση εισαγωγής δεδομένων από χρήστες

Στα πλαίσια της ασφαλέστερης διαχείρισης των δεδομένων που εισάγονται στην εφαρμογή από χρήστες, το framework κάνει μία βασική εξυγίανση (sanitization) οποιουδήποτε input stream ή συλλογής που μπορεί να εισάγει δεδομένα στην εφαρμογή, όπως `$_POST`, `$_GET`, `$_COOKIE`, `$_SERVER` και `$_FILE`. Με αυτό τον τρόπο αποτρέπονται βασικοί κίνδυνοι, που οφείλονται στην μη ελεγχόμενη διοχετεύση

των δεδομένων που εισάγουν οι χρήστες, απευθείας στην εφαρμογή, όπως για παράδειγμα, SQL injection.

Η κύρια βιβλιοθήκη του framework, που διαχειρίζεται το σύνολο αυτής της διαδικασίας εξυγίανσης και γενικότερης διαχείρισης, είναι η `input`, και φορτώνεται αυτόματα με την εκκίνησή του.

Μέσω της βιβλιοθήκης input έχουμε πρόσβαση σε οποιοδήποτε δεδομένο ή συλλογή δεδομένων επιθυμούμε. Για παράδειγμα η εκτέλεση της εντολής `$this->input->post()` θα μας επέστρεφε το σύνολο των δεδομένων που θα υπήρχαν στην συλλογή $_POST. Όλες οι μέθοδοι (functions) της input δέχονται δύο παραμέτρους. Η πρώτη παράμετρος, αφορά το όνομα μίας συγκεκριμένης τιμής που θέλουμε να ανακτήσουμε, ενώ η δεύτερη παράμετρος, αφορά την ρητή χρήση φίλτρων XSS.

Αναλυτικότερα, αν είχαμε μία HTML φόρμα, που είχε δύο πεδία, το όνομα χρήστη (username) και τον κωδικό πρόσβασης (password), και η φόρμα αποστελλόταν με την μέθοδο POST, τότε η εκτέλεση της εντολής `$this->input->post()`, θα είχε ως αποτέλεσμα την εμφάνιση του ακόλουθου output.

```
Array (
        [username] => xxxxxxxx
        [password] => xxxxxxxx
)
```

Εικόνα 3 - Παράδειγμα output $_POST

Αντίστοιχα, η εκτέλεση της εντολής `$this->input->post("username")` θα εμφάνιζε την τιμή που θα είχε πληκτρολογήσει ο χρήστης, στο πεδίο όνομα χρήστη της φόρμας.

Παρόμοια είναι και η λειτουργία των υπόλοιπων μεθόδων της συγκεκριμένης βιβλιοθήκης, και η αναφορά τους δεν κρίνεται απαραίτητη.

Βασική δομή εφαρμογής

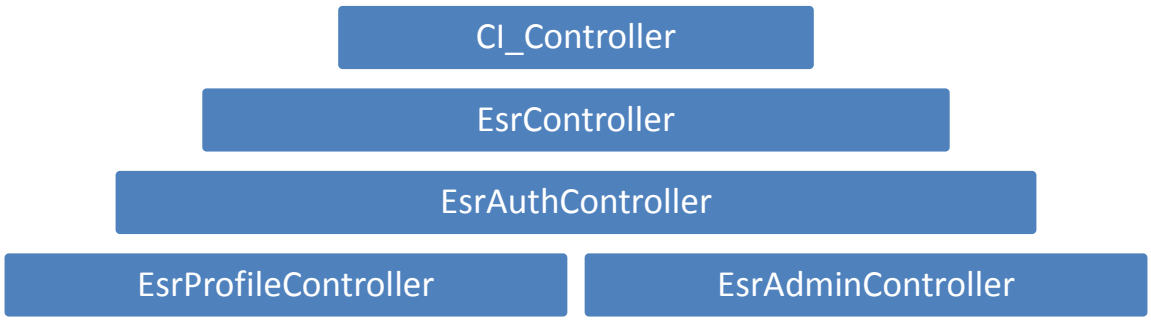
Όπως έχει ήδη αναφερθεί στην αρχή του κεφαλαίου, η εφαρμογή είναι οργανωμένη σε τρεις διακριτούς τομείς.

Όλες οι ενότητες της εφαρμογής, δηλαδή όλοι οι controllers, επεκτείνουν (extends) τον βασικό controller `EsrController`, ο οποίος με την σειρά του επεκτείνει τον κύριο controller του framework, τον `CI_Controller`, και είναι υπεύθυνος για

κάποιες εργασίες αρχικοποιήσης και κυρίως για τον αυτόματο εντοπισμό της γλώσσας της εφαρμογής.

Οποιοσδήποτε τομέας ή συγκεκριμένη σελίδα απαιτεί ελεγχόμενη πρόσβαση, επεκτείνει τον controller `EsrAuthController`, ο οποίος με την σειρά του, επεκτείνει τον βασικό controller `EsrController`, και είναι υπεύθυνος για τις λειτουργίες πιστοποιήσης και εξουσιοδότησης (authentication – authorization) για οποιαδήποτε υποκείμενη ενέργεια, με χρήση των αντίστοιχων βιβλιοθηκών.

Επιπλέον, η ανάγκη συγκέντρωσης κάποιων συχνά εμφανιζόμενων εντολών, όπως εντολές για τον ορισμό του θέματος στυλ (theme) ή εντολές φόρτωσης αρχείων γλώσσας, σε ένα κεντρικό σημείο, οδήγησε στην δημιουργία δύο ακόμη controllers, του `EsrProfileController` και του `EsrAdminController`, που επεκτείνουν (extends) των controller `EsrAuthController`, και αφορούν αντίστοιχα τον τομέα των προφίλ (profile) και της διαχείρισης (administration).



Εικόνα 4 – Βασική δομή & κληρονομικότητα controllers

Λειτουργίες CRUD

Οι λειτουργίες CRUD (Create – Read – Update – Delete), και πιο συγκεκριμένα η κωδικοποίησή τους, αφορούν το μεγαλύτερο μέρος της ανάπτυξης μίας εφαρμογής.

Μέσω των λειτουργιών CRUD, επιτυγχάνεται η διάδραση των χρηστών με την εφαρμογή, γεγονός που αυτομάτως καθιστά την ανάπτυξη τους δύσκολη, τόσο σε σχεδιαστικό επίπεδο, όσο και σε επίπεδο υλοποίησης, δεδομένου ότι εισάγονται πολλοί περιορισμοί για την διασφάλιση της παροχής των απαιτούμενων δεδομένων σε σωστή μορφή καθώς και μεγάλο επίπεδο λογικής, αναφορικά με την παρουσίαση των δεδομένων, ούτως ώστε να γίνεται πιο κατανοητή, επομένως και πιο αποδοτική η χρήση της εφαρμογής.

Στα πλαίσια χρήσης του framework CodeIgniter, κάθε κλάση υλοποιεί μία ή περισσότερες από τις λειτουργίες CRUD.

Create

Η λειτουργία Create (**C**RUD) αφορά την εισαγωγή δεδομένων, και την δημιουργία νέων εγγραφών. Λειτουργίες Create, συναντάμε σε ένα πολύ μεγάλο μέρος της εφαρμογής, ξεκινώνοντας από την διαδικασία της εγγραφής (registration), συνεχίζοντας σε διάφορα τμήματα του προφίλ (profile) και κλείνοντας στον τομέα διαχείρισης (administration) όπου η λειτουργία Create απαντάνται στο σύνολο των κλάσεων.

Η λειτουργία Create, υλοποιείται με την μέθοδο `insert()`, και ακολουθεί συνηθέστερα μία διαδικασία δύο βημάτων. Αρχικά, γίνεται έλεγχος για την ύπαρξη δεδομένων στην συλλογή `$_POST`, ώστε να διαπιστωθεί αν είναι δυνατή η συνέχεια στο δεύτερο βήμα ή όχι. Σε περίπτωση που δεν υφίστανται δεδομένα στην συλλογή `$_POST`, τότε καλείται το απαραίτητο view, με χρήση της βιβλιοθήκης `Layout`, το οποίο περιέχει μία φόρμα εισαγωγής στοιχείων. Σε αυτό το βήμα, ενδέχεται να απαιτείται η ανάκτηση δεδομένων από κάποιο model, για την προσυμπλήρωση κάποιων πεδίων της φόρμας ή για την εμφάνιση σχετικών επιλογών.

Στην συνέχεια, και αφού ο χρήστης προχωρήσει στην αποστολή της σχετικής φόρμας, συνεχίζεται η εκτέλεση της μεθόδου `insert()`, στο δεύτερο βήμα, όπου και γίνονται διάφορες λειτουργίες ελέγχου και επαλήθευσης των δεδομένων που έχουν εισαχθεί και στην συνέχεια ολοκληρώνεται η διαδικασία με την αποθήκευση των δεδομένων με χρήση των σχετικών models ή με την παράδοσή τους σε άλλες μεθόδους ή άλλους μηχανισμούς για περαιτέρω επεξεργασία. Σε περίπτωση που η επαλήθευση των δεδομένων αποτύχει (απουσία απαιτούμενων πεδίων, λανθασμένη μορφή κ.λπ.) τότε τα σχετικά μηνύματα σφάλματος καταχωρούνται στην σύνοδο σύνδεσης (session) με χρήση της βιβλιοθήκης Notifier, και στην συνέχεια η φόρμα εισαγωγής δεδομένων επανεμφανίζεται και υπάρχει δυνατότητα εμφάνισης όλων ή επιλεγμένων μηνυμάτων.

Read

Η λειτουργία Read (C**R**UD) αφορά την ανάγνωση του περιεχομένου και άλλων δεδομένων της εφαρμογής. Η λειτουργία Read, βρίσκεται σε κάθε σημείο της εφαρμογής, με ελάχιστες εξαιρέσεις, όπως κάποιες ασύγχρονες κλήσεις AJAX, όπου και πάλι υπάρχει ένα σύνολο δεδομένων προς εμφάνιση, ωστόσο η προβολή του παραγματοποιείται από άλλα μέρη της εφαρμογής.

Η λειτουργία Read, συνηθέστερα υλοποιείται με την μέθοδο `index()`. Η μέθοδος `index()`, όπως έχει προαναφερθεί, είναι αυτή που καλείται εξ ορισμού, σε περίπτωση απουσίας ρητής αίτησης για την κλήση κάποιας άλλης μεθόδου.

Στην μέθοδο `index()` πραγματοποιείται η ανάκτηση ενός συνόλου δεδομένων με την χρήση διαφόρων σχετικών models, και ακολουθεί η εμφάνιση του, συνήθως σε μορφή πίνακα ή πλέγματος (grid).

Η λειτουργία Read, μπορεί να χαρακτηριστεί ως μία από τις πιο σημαντικές, αφενός διότι προσφέρει στον χρήστη της εφαρμογής πρόσβαση στο σύνολο της πληροφορίας που αυτός επιζητεί, και αφετέρου διότι λειτουργεί ως «πύλη» προς όλες τις υπόλοιπες λειτουργίες.

Update

Η λειτουργία Update (CR**U**D) αφορά την επεξεργασία ήδη υπαρχόντων δεδομένων και την πιθανή μετατροπή των αποθηκευμένων εγγραφών. Η παρουσία της λειτουργίας Update, συνήθως ακολουθεί την λειτοργία Create, δεδομένου ότι όπου υπάρχει δυνατότητα δημιουργίας νέων εγγραφών, υπάρχει και ανάγκη επεξεργασίας των, είτε για την ανανέωσή τους ή για την επιδιόρθωση τυχόν σφαλμάτων.

Η λειτουργία Update, υλοποιείται με την μέθοδο `edit()`, και πραγματοποιείται σε τρία βήματα. Αρχικά γίνονται διάφοροι έλεγχοι ούτως ώστε να επαληθευτεί πως είναι δυνατή η ανάκτηση και επεξεργασία του ζητούμενου πόρου, και στην συνέχεια γίνεται ο έλεγχος της συλλογής `$_POST`, ώστε να διαπιστωθεί αν υπάρχουν δεδομένα άρα και αν είναι δυνατή η συνέχεια στο επόμενο βήμα. Αν δεν υφίστανται δεδομένα, τότε εμφανίζεται η φόρμα επεξεργασίας, η οποία είναι μορφολογικά πανομοιότυπη με την φόρμα εισαγωγής δεδομένων, με την διαφορά ότι τα πεδία είναι προσυμπληρωμένα με τα στοιχεία του επιλεγμένου επεξεργαζόμενου πόρου.

Όταν ο χρήστης προχωρήσει στην αποθήκευση των αλλαγών που έχει πραγματοποιήσει, ακολουθείται μία διαδικασία ελέγχου και επαλήθευσης των δεδομένων που έχουν εισαχθεί, αντίστοιχη αυτής της λειτουργίας Create, και η Update, ολοκληρώνεται με την αποθήκευση των αλλαγών με κλήση των κατάλληλων μεθόδων των σχετικών models.

Delete

Η λειτουργία Delete, όπως απορέει και από την ονομασία της, αφορά την διαγραφή αποθηκευμένων δεδομένων και γενικότερα την απομάκρυνση περιεχομένου από την εφαρμογή. Η λειτουργία Delete, εμφανίζεται σε λιγότερα σημεία της εφαρμογής, και κατά κύριο λόγο υλοποιείται στον τομέα διαχείρισης (administration).

Η λειτουργία Delete, υλοποιείται με την μέθοδο `delete()` και ολοκληρώνεται σε δύο βήματα. Αρχικά γίνονται διάφοροι έλεγχοι για να επαληθευτεί πως είναι δυνατή η ανάκτηση και διαγραφή του ζητούμενου πόρου και στην συνέχεια γίνεται ένας επιπλέον έλεγχος που αφορά την ύπαρξη μίας τιμής ασφαλείας (security token).

Σε περίπτωση που δεν υφίσταται η τιμή ασφαλείας, τότε δημιουργείται μία, με χρήση μεθόδων γεννητριών ψευδοτυχαίων συμβολοσειρών, και αποθηκεύεται στην σύνοδο σύνδεσης (session), ενώ παράλληλα παρουσιάζεται στον χρήστη, ένας διάλογος ειδοποίησης και επιβεβαίωσης της διαγραφής. Εάν ο χρήστης, επιλέξει να συνεχίσει με την διαδικασία της διαγραφής τότε, στο δεύτερο βήμα, γίνεται επαλήθευση της τιμής ασφαλείας, η οποία δίδεται ως παράμετρος στην μέθοδο `delete()`, μέσω του URL, με την ήδη αποθηκευμένη στη σύνοδο σύνδεσης (session) τιμή. Για να είναι δυνατή η διαγραφή του επιλεγμένου πόρου, θα πρέπει οι δύο τιμές ασφαλείας να είναι όμοιες, διαφορετικά η διαδικασία διακόπτεται και εμφανίζεται σχετικό μήνυμα σφάλματος.

Η διαδικασία θεωρείται ολοκληρωμένη, είτε σε περίπτωση που ακύρωθεί από τον χρήστη στο στάδιο της επιβεβαιώσης, είτε με την επιτυχή διαγραφή του επιλεγμένου πόρου.

Λογαριασμοί & χρήστες

Το σύνολο της εφαρμογής και της λειτουργικότητάς της, βασίζεται στην έννοια του λογαριασμού, αφού κάθε συμμετοχή αναπαρίσταται από έναν λογαριασμό.

Κάθε λογαριασμός, ταυτοποιείται από ένα μοναδικό, τυχαίο ID και περιλαμβάνει πληροφορίες σχετικά με τον τύπο του λογαριασμού (σχολείο, μαθητής, εξωτερικός συνεργάτης), μία σύντομη περιγραφή και ένα banner της συμμετοχής, την ημερομηνία εγγραφής και το αν είναι η πρώτη φορά που συμμετέχει.

Επίσης κάθε λογαριασμός, αποτελείται από μέλη, τα στοιχεία των οποίων διαφοροποιούνται ανάλογα με τον τύπο του λογαριασμού. Συγκεκριμένα για λογαριασμούς σχολείων, διατηρούνται εκτός από τα στοιχεία του υπεύθυνου και τα στοιχεία του σχολείου (ονομασία, εκπαιδευτική βαθμίδα, περιοχή κ.λπ.), σε αντίθεση με τους υπόλοιπους τύπους λογαριασμών (μαθητής, εξωτερικός συνεργάτης) όπου διατηρούνται αποκλειστικά τα στοιχεία των φυσικών προσώπων.

Για όλους τους χρήστες των λογαριασμών, διατηρούνται γενικά δεδομένα, όπως το ονοματεπώνυμο τους, στοιχεία επικοινωνίας, ο τύπος πιστοποίησης που έχουν επιλέξει (τοπική ή μέσω CAS που παρέχεται από το Π.Σ.Δ.), το όνομα χρήστη, και τέλος, για χρήστες που έχουν επιλέξει τοπική πιστοποίηση, ο κωδικός πρόσβασης. Παράλληλα για κάθε χρήστη, ανάλογα με τον τύπο του λογαριασμού του, διατηρούνται επιπλέον στοιχεία, όπως για παράδειγμα το σχολείο στο οποίο ανήκει ο εκάστοτε υπεύθυνος ή μαθητής, ο ρόλος των υπεύθυνων (διευθυντής σχολείου, καθηγητής ή άλλος ρόλος) και άλλα.

Οι διαχειριστές έχουν πλήρη έλεγχο των λογαριασμών, με δυνατότητα άμεσης προβολής των στοιχείων τους (πλην κωδικών πρόσβασης και άλλων τυχόν ευαίσθητων δεδομένων), επίβλεψης των παραγωγών που αναρτούν, προβολή των μελών του εκάστοτε λογαριασμού και του ιστορικού ενεργειών του κάθε μέλους. Ακόμη, οι διαχειριστές έχουν δυνατότητα παύσης ενός ολόκληρου λογαριασμού ή συγκεκριμένων μελών ενός λογαριασμού, όπου αυτό κρίνεται απαραίτητο, όπως για παράδειγμα, σε περιπτώσεις μη συμμόρφωσης με τον κανονισμό του *European School Radio* ή σε περιπτώσεις μη επαρκούς ποιοτικού επιπέδου.

Διαδικασία εγγραφής

Νέοι χρήστες της εφαρμογής μπορούν να εγγραφούν και να προχωρήσουν στην δημιουργία ενός νέου λογαριασμού. Η εφαρμογή διαχείρισης, έχει προοπτική χρήσης από σχολεία, μαθητές και εξωτερικούς συνεργάτες, και έχει δομηθεί με τέτοιο τρόπο ώστε κάθε τμήμα της, να προβλέπει χρήση από μέλη με διαφορετικό τύπο λογαριασμού, ωστόσο στα πλαίσια αυτής της Πτυχιακής Εργασίας, έχει ολοκληρωθεί αποκλειστικά το κομμάτι χρήσης από σχολεία.

Όπως πολλά άλλα τμήματα αυτής της εφαρμογής, έτσι και η διαδικασία της εγγραφής είναι χωρισμένη σε επιμέρους διακριτά στάδια, ούτως ώστε να είναι πιο εύκολη για τον τελικό χρήστη η κατανόηση της ροής της όλης διαδικασίας, και να μην υπάρχει σύγχυση σχετικά με τα στοιχεία που πρέπει να εισαχθούν.

Η διαδικασία δημιουργίας ενός νέου λογαριασμού σχολείου, ξεκινάει με την φόρμα εισαγωγής γενικών στοιχείων για το σχολείο, συνεχίζει με αντίστοιχα στοιχεία για τον υπεύθυνο του λογαριασμού και τα στοιχεία σύνδεσής του, και τέλος έχουμε την ρύθμιση διαφόρων παραμέτρων της συμμετοχής και την συνολική επισκόπηση.

Πιο αναλυτικά, στο πρώτο στάδιο της εγγραφής, που αφορά τα στοιχεία του σχολείου, ο χρήστης καλείται να συμπληρώσει το όνομα του σχολείου, την εκπαιδευτική του βαθμίδα και μπορεί να προσθέσει προαιρετικά σχόλια. Στη συνέχεια, πρέπει να συμπληρωθούν τα στοιχεία διεύθυνσης και επικοινωνίας, και πιο συγκεκριμένα, η χώρα, η πόλη, η διεύθυνση και ο ταχυδρομικός κωδικός του σχολείου, ενώ απαιτείται η εισαγωγή ενός τηλέφωνου επικοινωνίας και μίας διεύθυνσης ηλεκτρονικού ταχυδρομείου (email).

Η συμπλήρωση της ονομασίας του σχολείου, εκκινεί μία ασύγχρονη κλήση AJAX, η οποία ελέγχει την ύπαρξη ήδη εγγεγραμμένων σχολείων με την ίδια ή παρόμοια ονομασία. Σε περίπτωση που βρεθούν αντιστοιχίσεις, εμφανίζεται στον χρήστη ένα ενημερωτικό μήνυμα, και η λίστα των αντιστοιχιζόμενων σχολείων. Αν ο χρήστης ανήκει σε κάποιο από τα αναγραφόμενα σχολεία, τότε θα πρέπει να επιλέξει το σχολείο που επιθυμεί και στην συνέχεια ανακατευθύνεται αυτόματα στο δεύτερο στάδιο της εγγραφής. Οι πολλαπλές εγγραφές και δημιουργία λογαριασμών για το ίδιο σχολείο,

μπορούν να οδηγήσουν σε απρόβλεπτη συμπεριφορά της εφαρμογής και γι αυτό ακολουθείται αυτή η διαδικασία.

Η συμπλήρωση των στοιχείων διεύθυνσης, έχει ως αποτέλεσμα την εκκίνηση επίσης μίας ασύγχρονης κλήσης AJAX για τον γεωγραφικό εντοπισμό (geolocation), της εν λόγω διεύθυνσης και την μετακίνηση του δείκτη του χάρτη στο σημείο που την αντιπροσωπεύει. Σε περίπτωση που η διαδικασία γεωγραφικού εντοπισμού δεν βρει ακριβή αντιστοίχιση των στοιχείων διεύθυνσης, επιστρέφεται μία λίστα με το σύνολο των διευθύνσεων που έχουν αντιστοιχηθεί, και ο χρήστης καλείται να επιλέξει την σωστή. Επίσης σε περίπτωση, αδυναμίας ολοκλήρωσης του γεωγραφικού εντοπισμού, είτε λόγω λανθασμένων στοιχείων διεύθυνσης είτε λόγω μη επαρκών στοιχείων στον πάροχο της υπηρεσίας γεωγραφικού εντοπισμού (π.χ. απομακρυσμένες περιοχές), ο χρήστης έχει την δυνατότητα να μετακινήσει χειροκίνητα τον δείκτη του χάρτη στο σωστό σημείο.

Στο δεύτερο στάδιο της εγγραφής, ο χρήστης καλείται να συμπληρώσει τα προσωπικά του στοιχεία, εφόσον θα είναι ο υπεύθυνος του σχολείου, δηλαδή το ονοματεπώνυμο του, τον ρόλο που κατέχει μέσα στην σχολική κοινότητα (διευθυντής, υποδιευθυντής, δάσκαλος/καθηγητής, σχολικός σύμβουλος ή άλλο), την ειδικότητά του, και στοιχεία επικοινωνίας (τηλέφωνο και email), ενώ τέλος μπορεί να προσθέσει κάποια προαιρετικά σχόλια.

Στο τρίτο στάδιο, υπάρχει η επιλογή μεθόδου πιστοποίησης και η αντίστοιχη δημιουργία του λογαριασμού. Η επιλογή μεθόδου πιστοποίησης επιτρέπει στον χρήστη να διαλέξει αν επιθυμεί η πιστοποίηση του να γίνετε μέσω της υπηρεσίας CAS του Πανελλήνιου Σχολικού Δικτύου, ή μέσω του τοπικού μηχανισμού πιστοποίησης.

Η επιλογή μεθόδου πιστοποίησης είναι διαθέσιμη αποκλειστικά για μέλη, που έχουν δηλώσει ως χώρα παρουσίας την Ελλάδα (βλ. πρώτο στάδιο εγγραφής), αφού για μέλη σχολείων του εξωτερικού η πιστοποίηση μέσω της υπηρεσίας CAS του Πανελλήνιου Σχολικού Δικτύου, δεν θα ήταν δυνατή.

Στην περίπτωση επιλογής της υπηρεσίας CAS του Πανελλήνιου Σχολικού Δικτύου, ως μέθοδο πιστοποίησης, ο χρήστης μεταφέρεται (redirect) στην υπηρεσία CAS, όπου και του ζητείται να κάνει είσοδο (login) με τα στοιχεία του λογαριασμού του. Με την επιτυχή σύνδεση, ο χρήστης μεταφέρεται (redirect) και πάλι πίσω στην εφαρμογή διαχείρισης και πιο συγκεκριμένα στο στάδιο της εγγραφής όπου βρισκόταν και πριν την μεταφορά στην υπηρεσία CAS. Η διαδικασία αυτή γίνεται, ούτως ώστε να επιβεβαιωθεί ότι ο νέος χρήστης είναι όντως μέλος της εκπαιδευτικής κοινότητας, και για να καταχωρηθεί το username του στην βάση δεδομένων και συγκεκριμένα στον πίνακα `User`. Αντίθετα, αν ο χρήστης συνεχίσει με την επιλογή τοπικής πιστοποίησης, τότε του ζητείται να πληκτρολογήσει το επιθυμητό όνομα χρήστη και κωδικό πρόσβασης.

Συνεχίζοντας, στο τέταρτο στάδιο της εγγραφής, ο χρήστης πρέπει να συμπληρώσει κάποια στοιχεία για την συμμετοχή στο *European School Radio*, και πιο συγκεκριμένα, αν είναι η πρώτη φορά που συμμετέχει, μια σύντομη περιγραφή, και στη συνέχεια, δύο αρχεία. Το πρώτο αρχείο, είναι ένα εικονίδιο (banner), και λειτουργεί ως το λογότυπο της συμμετοχής. Το δεύτερο, είναι αρχείο ήχου (spot) και ουσιαστικά είναι μια σύντομη παρουσίαση της συμμετοχής.

Στο τελευταίο στάδιο της εγγραφής, υπάρχει η επισκόπηση, όπου εμφανίζονται όλα τα στοιχεία που έχει μέχρι στιγμής καταχωρήσει ο χρήστης στην εφαρμογή. Αν εντοπίσει σφάλματα, μπορεί να επιστρέψει στα προηγούμενα στάδια, και να διορθώσει ότι επιθυμεί, διαφορετικά προχωρά στην αποθήκευση των στοιχείων που έχει εισάγει και στην δημιουργία του λογαριασμού του.

Η διαδικασία δημιουργίας λογαριασμού απαρτίζεται από την αποθήκευση γενικών δεδομένων στον πίνακα `Account`, των βασικών στοιχείων του χρήστη καθώς και τον στοιχείων σύνδεσης (username/password) στον πίνακα `User`, των στοιχείων επικοινωνίας στον πίνακα `Contact`, των δεδομένων του σχολείου στον πίνακα `School` και τα υπόλοιπα στοιχεία του υπεύθυνου χρήστη στον πίνακα `Supervisor` και `SupervisorSector`.

Τύποι παραγωγών, ζώνες & θεματικές ενότητες

Το *European School Radio*, στο πλαίσιο λειτουργίας του, παρέχει δυνατότητα δημιουργίας και μετάδοσης διαφορετικών παραγωγών. Ο δύο κατηγορίες παραγωγών είναι οι εκπομπές και τα σποτ. Ο κύριος άξονας διαφοροποίησης μεταξύ εκπομπών και σποτ είναι η μέγιστη επιτρεπόμενη διάρκεια τους, με τις εκπομπές να διαρκούν από 15 λεπτά μέχρι και 2 ώρες, ενώ τα σποτ έχουν διάρκεια που συνήθως δεν ξεπερνά το 1 λεπτό. Επιπλέον, οι εκπομπές συνήθως είναι τακτικές, και οι περισσότερες, απαιτούν προηγούμενη δήλωση και προγραμματισμό.

Στα πλαίσια της απαίτησης πλήρους παραμετροποιήσιμης λειτουργίας, υπάρχει δυνατότητα δημιουργίας τύπων παραγωγών οι οποίες είναι θεματικές ή ειδικές - αφιερώματα.

Οι θεματικές παραγωγές είναι συνήθως μικρές σε διάρκεια (~ 15 λεπτά) και το περιεχόμενό τους ανήκει σε κάποια συγκεκριμένη θεματική ενότητα (κοινωνία, σχολικά νέα, αθλητισμός κ.λπ.).

Οι ειδικές παραγωγές είναι συνηθέστερα κάποιου είδους αφιερώματα, και πραγματεύονται διάφορα θέματα, όπως για παράδειγμα μία παγκόσμια ημέρα (π.χ. ημέρα κατά του AIDS), κάποιο δημοφιλές γεγονός (π.χ. έναρξη Ολυμπιακών Αγώνων) ή άλλες περιστάσεις (π.χ. ένας χρόνος λειτουργίας *European School Radio*) και έχουν

επίσης συγκεκριμένη περίοδο ισχύος, η οποία ορίζεται από τους διαχειριστές της εφαρμογής.

Μετά τον ορισμό των επιθυμητών τύπων παραγωγών, ακολουθεί ο ορισμός ζώνης, όπου δηλώνεται η συσχέτιση ανάμεσα σε τύπους παραγωγών και συγκεκριμένες ζώνες.

Οι ζώνες, ουσιαστικά σχηματίζουν το συνολικό εβδομαδιαίο ραδιοφωνικό πρόγραμμα, και χωρίζονται σε δύο μεγάλες κατηγορίες, τις ζώνες μουσικής και τις ζώνες παραγωγών. Οι ζώνες μουσικής, αποτελούνται αποκλειστικά από μουσικό περιεχόμενο, το οποίο ορίζεται από την συντονιστική ομάδα του *European School Radio*. Η διαχείριση του περιεχομένου των ζωνών μουσικής είναι κάτι που δεν επαφίεται στα πλαίσια αυτής της εφαρμογής διαχείρισης. Οι ζώνες παραγωγών, αποτελούνται από τις παραγωγές που αναρτούν οι χρήστες. Κατά την διάρκεια δημιουργίας μίας νέας ζώνης ή επεξεργασίας μίας ήδη υπάρχουσας, οι διαχειριστές καλούνται να σχηματίσουν την συσχέτιση μεταξύ των παραγωγών που απαιτούν προγραμματισμό και της εκάστοτε ζώνης. Το αποτέλεσμα αυτής της διαδικασίας είναι να «γνωρίζει» το σύστημα για κάθε τύπο παραγωγής τις ζώνες στο πλαίσιο των οποίων θα μπορούσε να αναρτηθεί και, φυσικά, να μεταδωθεί.

Ένα σημαντικό κομμάτι της διαχείρισης των ζωνών, είναι ο προγραμματισμός τους. Οι διαχειριστές έχουν στην διάθεσή τους, ένα εβδομαδιαίο 24-ωρο πρόγραμμα (σε μορφή πίνακα), χωρισμένο σε χρονοθυρίδες (time slots) των 30 λεπτών, και έχουν δυνατότητα να επιλέξουν σε ποια time slots (συνεχόμενα ή μη) θα μεταδίδεται η εκάστοτε ζώνη. Ο προγραμματισμός αυτός γίνεται με τον περιορισμό πως ένα time slot, μπορεί να περιέχει το πολύ μία ζώνη μουσικής και μία ζώνη παραγωγών.

Επιπλέον, για τις παραγωγές που έχουν ορισθεί ως θεματικές, κατά την δημιουργία ή επεξεργασία μίας ζώνης, υπάρχει δυνατότητα επιλογής και των θεματικών ενοτήτων που η παραγωγή αυτή θα καλύπτει, ή καλύτερα, επιτρέπεται να έχει ως περιεχόμενο.

Τέλος, οι θεματικές ενότητες, είναι μία λίστα, πλήρως επεξεργάσιμη από τα μέλη της διαχειριστικής ομάδας, όπου αναφέρονται όλα τα διαθέσιμα θέματα τα οποία μπορούν να αναπτυχθούν στα πλαίσια μίας παραγωγής.

Ρυθμίσεις

Διάφορες παράμετροι που αφορούν την λειτουργία της εφαρμογής, υφίστανται εντός του τομέα (σελίδας) των ρυθμίσεων. Το συνόλο των ρυθμίσεων της εφαρμογής αποθηκεύονται σε XML αρχείο, δεδομένου ότι δεν υπάρχει διεπαφή διαχείρισης των διαθέσιμων ρυθμίσεων αφού κάτι τέτοιο θα ήταν υπερβολικά περίπλοκο για τον τελικό χρήστη, ενώ ταυτόχρονα ο ελάχιστος αριθμός αλλαγών που αφορούν τις διαθέσιμες ρυθμίσεις δεν θα μπορούσε να το δικαιολογήσει, και φυσικά η επεξεργασία ενός

αρχείου XML είναι πολύ πιο απλή και εύκολη για έναν χρήστη συγκριτικά με την διαχείριση μίας βάσης δεδομένων, είτε αυτή γίνεται μέσω γραμμής εντολών ή μέσω κάποιου γραφικού περιβάλλοντος.

Οι ρυθμίσεις αφορούν διάφορα τμήματα της εφαρμογής, όπως για παράδειγμα την πιστοποίηση χρηστών, την μεταφόρτωση εκπομπών, την εξ ορισμού γλώσσα, την ζώνη ώρας του εξυπηρετητή, ρυθμίσεις σελιδοποίησης (pagination) και άλλα.

Το αρχείο ρυθμίσεων, είναι οργανωμένο σε groups και subgroups (όπου αυτό κρίνεται απαραίτητο), και κάθε group, ή subgroup αντίστοιχα, περιέχει ένα ή περισσότερα settings. Κάθε setting με την σειρά του, έχει διάφορες επιμέρους παραμέτρους, που ορίζουν το μοναδικό αναγνωριστικό του κάθε setting (ID), τον τύπο του (κείμενο, αριθμός, λίστα επιλογών κ.ο.κ.), κάποια πιθανή προκαθορισμένη τιμή, και υπάρχει επίσης δυνατότητα ορισμού κανόνων επαλήθευσης (validation) των τιμών που θα πληκτρολογήσει ο χρήστης. Ένα υπόδειγμα, του αρχείου XML με τις ρυθμίσεις της εφαρμογής, είναι διαθέσιμο στο Παράρτημα Β’.

Ασφάλεια

Για να επιτευχθεί η μέγιστη ασφάλεια της εφαρμογής και η σωστή λειτουργία της, έχει αναπτυχθεί ένας μηχανισμός ομάδων χρηστών (user groups), δικαιωμάτων χρήσης (permissions) και λιστών ελέγχου πρόσβασης (ACL – Access Control List).

Κάθε εγγεγραμένος χρήστης της εφαρμογής ανήκει σε μία ομάδα χρηστών. Σε αρχικό στάδιο, υφίστανται μόνο δύο ομάδες, οι Διαχειριστές, και οι Χρήστες, αλλά υπάρχει δυνατότητα πλήρους διαχείρισης και ενεργειών προσθήκης, επεξεργασίας ή διαγραφής των ομάδων χρηστών, για να υφίσταται πάντα απόλυτη προσαρμογή στις ανάγκες ασφάλειας του *European School Radio*.

Επιπλέον, ορίζονται δικαιώματα χρήσης, βάσει των οποίων επιτρέπεται ή απαγορεύεται η πρόσβαση, η εκτέλεση συγκεκριμένων ενεργειών ή η προβολή δεδομένων. Τα δικαιώματα χρήσης, ορίζονται από έναν τομέα (π.χ. administration), μία σελίδα (π.χ. accounts), μία ενέργεια (π.χ. delete) και τέλος την εξ όρισμου (default) επιλογή για το αν επιτρέπεται ή όχι, η οριζόμενη ενέργεια.

Η συσχέτιση μεταξύ δικαιωμάτων χρήσης και ομάδων χρηστών, γίνεται μέσω των λιστών ελέγχου πρόσβασης (ACLs). Στις λίστες ελέγχου πρόσβασης, αρχικά επιλέγεται μία ομάδα χρηστών ή ένας συγκεκριμένος χρήστης, και όλα τα διαθέσιμα δικαιώματα χρήσης συνδέονται με την επιλεγμένη ομάδα ή χρήστη. Στην συνέχεια, ακολουθείται μία διαδικασία αντικατάστασης (overwrite) των δικαιωμάτων χρήσης, όπου υπάρχει δυνατότητα, αλλαγής της εξ ορισμού επιλογής (επιτρέπεται/απαγορεύεται) που

είχε επιλεχθεί για το εκάστοτε permission, ούτως ώστε να αντικατοπτρίζει την τελική απόφαση χρήσης (επιτρέπεται/απαγορεύεται) για την επιλεγμένη ομάδα ή χρήστη.

Φυσικά υπάρχει και δυνατότητα επαναφοράς ενός δικαιώματος χρήσης στην αρχική του κατάσταση, δηλαδή στην εξ ορισμού επιλογή (διαδικασία revert).

Εκτός της ασφάλειας που ορίζεται σε επίπεδο χρηστών, έχουν εφαρμοσθεί επιπλέον μηχανισμοί και έχουν ληφθεί κάποιες ενέργειες για την περαιτέρω ασφάλεια της εφαρμογής.

Σε επίπεδο δεδομένων, και πιο συγκεκριμένα σε επίπεδο αποθήκευσης, τα αναγνωριστικά των εγγραφών (IDs), σε κανέναν από τους πίνακες που έχουν υλοποιηθεί, δεν χρησιμοποιείται η επιλογή AUTO INCREMENT της MySQL, αλλά τυχαία αναγνωριστικά που δημιουργούνται από την εφαρμογή. Αυτά τα αναγνωριστικά είναι παράγωγα της μεθόδου `uniqid()` η οποία είναι ενσωματωμένη στην PHP. Με αυτό τον τρόπο, η σάρωση (scanning) των αποθηκευμένων δεδομένων, γίνεται πολύ πιο δύσκολη, έως και αδύνατη.

Η σάρωση δεδομένων είναι μία διαδικασία, όπου ένας χρήστης της εφαρμογής, έχοντας αναλύσει την δομή των URLs, μπορεί να εντοπίσει το, συνήθως αριθμητικό, ID μίας οντότητας (π.χ. λογαριασμός) και στη συνέχεια τροποποιώντας το συγκεκριμένο ID, να αποκτήσει πρόσβαση σε δεδομένα άλλων χρηστών. Για παράδειγμα, αν εξετάσουμε το URL

```
http://www.esr.gr/user/productions/list/25
```

μπορούμε να καταλήξουμε στο συμπέρασμα, πως το συγκεκριμένο URL είναι «υπεύθυνο» για την εμφάνιση όλων των παραγωγών του λογαριασμού με ID 25, που είναι το ID του λογαριασμού του τρέχοντος χρήστη. Σε περίπτωση που ο χρήστης αυτός, αλλάξει τα στοιχεία του παραπάνω URL και αντικαταστήσει τον αριθμό 25 με κάποιο άλλο αριθμό (π.χ. 26), τότε, δεδομένου ότι υφίσταται λογαριασμός με ID 26, θα μπορέσει να δει το σύνολο των παραγωγών αυτού του λογαριασμού.

Αυτή η συμπεριφορά θα μπορούσε να αποτραπεί ακολουθώντας ενέργειες εξουσιοδότησης (authorization), όπου για κάθε αίτηση (request) των χρηστών, θα έπρεπε να ελέγχεται και να απαιτείται κάποιου είδους ταύτιση ή «εμπιστοσύνη», αντίστοιχη των δικαιωμάτων χρήσης, μεταξύ του τρέχοντος χρήστη και του δημιουργού/συντάκτη του ζητούμενου πόρου. Ακολουθώντας μία τέτοια υλοποιήση, ωστόσο, προστίθεται στην εφαρμογή επιπλέον overhead, το οποίο μάλιστα σε ορισμένες περιπτώσεις, όπου απαιτείται ο αναδρομικός έλεγχος πολλαπλών, αλληλοεξαρτόμενων ή πολυεπίπεδων πόρων, είναι σημαντικό, οδηγώντας σε αργή απόκριση και ενδεχομένως καθιστώντας την εφαρμογή μη αποδοτική ή ακόμη και μη λειτουργική.

Τέλος, αναφορικά με την διατήρηση των κωδικών πρόσβασης, αυτοί αποθηκεύονται σε μονόδρομα κατακερματισμένη (hashed) μορφή και συγκεκριμένα με χρήση του αλγορίθμου SHA-512 με επιπλέον salt.

Ως salt, χαρακτηρίζεται ένα μικρό σύνολο, τυχαίων χαρακτήρων που προσαρτώνται στον κωδικό πρόσβασης που δίδεται στην συνάρτηση κατακερματισμού, και έχει ως σκοπό την αύξηση της τυχαιότητας (randomness) της παραγώμενης συμβολοσειράς (output string).

Μηχανισμός πιστοποίησης χρηστών

Ένα από τα κυριότερα τμήματα της ασφάλειας της εφαρμογής, είναι αυτό της πιστοποίησης χρηστών. Η απόφαση για απόρριψη της χρήσης της υπηρεσίας CAS, οδήγησε στην ανάπτυξη ενός τοπικού μηχανισμού πιστοποιήσης, ο οποίος βασίζεται στην χρήση ονόματος χρήστη (username) και κωδικού πρόσβασης (password).

Οποιοδήποτε τμήμα της εφαρμογής, δηλαδή οποιοσδήποτε controller, επεκτείνει τον `EsrAuthController`, ουσιαστικά απαιτεί την πιστοποίηση του χρήστη που πρόκειται να χρησιμοποιήσει το εκάστοτε τμήμα. Ο `EsrAuthController`, βασίζεται στην λειτουργικότητα της βιβλιοθήκης `Authentication`. Αρχικά για κάθε αίτηση (request) καλείται η μέθοδος `isLoggedIn()` η οποία ελέγχει αν ο χρήστης που χρησιμοποιεί την εφαρμογή είναι ήδη πιστοποιημένος ή όχι. Ο έλεγχος αυτός βασίζεται σε τιμές της συνόδου σύνδεσης (session), και σε ορισμένες περιπτώσεις σε ειδικά cookies που είναι υπεύθυνα για την διατήρηση της σύνδεσης ενός χρήστη, κατ' απαίτησή του. Εάν η μέθοδος `isLoggedIn()` επιστρέψει την τιμή `false`, τότε ο χρήστης θα πρέπει να πιστοποιηθεί, και ανακατευθείνεται αυτόματα στην σελίδα εισόδου (log in).

Στην σελίδα εισόδου, υπάρχουν δύο επιλογές για σύνδεση, μέσω της υπηρεσίας CAS του Πανελλήνιου Σχολικού Δικτύου, ή εισάγοντας τα στοιχεία σύνδεσης (όνομα χρήστη και κωδικό πρόσβασης) στην διαθέσιμη φόρμα.

Σε περίπτωση που χρησιμοποιηθεί η υπηρεσία CAS, τότε ο χρήστης ανακατευθείνεται στην σελίδα πιστοποίησης της υπηρεσίας CAS, όπου και παραμένει μέχρι την επιτυχή πιστοποίηση του. Ο χρήστης ανακατευθείνεται και πάλι στην εφαρμογή διαχείρισης, ενώ καταχωρούνται στο session διάφορα στοιχεία τα οποία έχουν ανακτηθεί βάσει του username που επιστρέφεται από την διαδικασία πιστοποίησης της υπηρεσίας CAS.

Σε περίπτωση που χρησιμοποιηθεί η τοπική πιστοποίηση, αρχικά γίνεται έλεγχος για το αν ο χρήστης έχει καταχωρήσει τιμές και στα δύο πεδία της σχετικής φόρμας, και στη συνέχεια καλείται η μέθοδος `logIn()` της βιβλιοθήκης `Authentication`, όπου και αρχικοποιούνται δύο μεταβλητές. Η πρώτη μεταβλητή αφορά το μέγιστο όριο

επιτρεπόμενων προσπαθειών σύνδεσης (`allowedAttempts`) και η δεύτερη το χρόνο που θα πρέπει να περιμένει ένας χρήστης πριν μπορέσει να προσπαθήσει εκ νέου μετά από επαναλαμβανόμενες αποτυχημένες προσπάθειες (`attemptsWindow`). Στη συνέχεια για κάθε αίτηση σύνδεσης, αυξάνεται μία τρίτη μεταβλητή η οποία αποθηκεύεται στο session, ενώ ταυτόχρονα αποθηκεύεται και η χρονική στιγμή (timestamp) της τελευταίας αίτησης σύνδεσης. Εφόσον η τρίτη μεταβλητή είναι μικρότερη ή ίση του επιτρεπόμενου ορίου προσπαθειών, η διαδικασία συνεχίζεται, ενώ σε αντίθετη περίπτωση, διακόπτεται με ένα μήνυμα σφάλματος, το οποίο εμφανίζεται για χρονικό διάστημα ίσο με την χρόνο που ορίζεται από την μεταβλητή `attemptsWindow`.

Από το μοντέλο User, γίνεται προσπάθεια ανάκτησης των στοιχείων ενός χρήστη το username του οποίου θα πρέπει να αντιστοιχεί στο όνομα χρήστη, που πληκτρολογήθηκε στην φόρμα σύνδεσης. Σε περίπτωση που δεν βρεθούν εγγραφές στην βάση δεδομένων, δεν επιτρέπεται η πρόσβαση στα μέλη του λογαριασμού ή στον συγκεκριμένο χρήστη (blocked account/blocked user), ή ο κωδικός πρόσβασης δεν αντιστοιχεί με τον αποθηκευμένο, τότε εμφανίζονται κατάλληλα μηνύματα σφάλματος και η διαδικασία πιστοποίησης διακόπτεται. Σε διαφορετική περίπτωση, οι προαναφερόμενες μεταβλητές επαναφέρονται στην αρχική τους κατάσταση, ενώ καταχωρούνται στο session διάφορα στοιχεία για τον πιστοποιημένο πλέον χρήστη.

Τα στοιχεία που καταχωρούνται στο session, χρησιμοποιούνται σε όλο το εύρος της εφαρμογής για την αναγνώριση του χρήστη και είναι τα εξής:

- `accountId`
  το αναγνωριστικό του λογαριασμού του συγκεκριμένου χρήστη
- `accountType`
  ο τύπος του λογαριασμού (σχολείο, μαθητής ή εξωτερικός συνεργάτης)
- `isActive`
  δηλώνει αν ο λογαριασμός είναι ενεργός, δηλαδή αν είναι συμπληρωμένα όλα τα στοιχεία της συμμετοχής (περιγραφή, banner, spot και θεματολογία)
- `userId`
  το αναγνωριστικό του συγκεκριμένου χρήστη

και υπάρχουν επιπλέον στοιχεία τα οποία διαφοροποιούνται ανάλογα με τον τύπο του λογαριασμού. Συγκεκριμένα, για τα σχολεία που είναι και ο μοναδικός τύπος που χρησιμοποιείται στα πλαίσια αυτής της Πτυχιακής Εργασίας, διατηρούνται επίσης τα εξής στοιχεία:

- `schoolId`
  το αναγνωριστικό του σχολείου που εκπροσωπείται από τον τρέχον λογαριασμό

- `supervisorId`
  το ανανγωριστικό του υπεύθυνου του σχολείου

Επαναφορά κωδικών πρόσβασης

Ένα επιπλέον κομμάτι που αφορά την ασφάλεια της εφαρμογής και πιο συγκεκριμένα την πιστοποίηση χρηστών, είναι το πολύ συχνό φαινόμενο της απώλειας ενός κωδικού πρόσβασης και η ανάγκη υπενθύμισής του στον χρήστη. Υπάρχουν ορισμένες εφαρμογές οι οποίες προσφέροντας λειτουργίες υπενθύμισης κωδικών πρόσβασης, ζητούν το email ή το όνομα χρήση του μέλους, και στη συνέχεια αποστέλουν τον κωδικό πρόσβασης αυτόματα. Αυτή η διαδικασία κρύβει κινδύνους αποκάλυψης του κωδικού πρόσβαση σε μη εξουσιοδοτημένα άτομα, αφού μεταφέρεται μέσω μη ασφαλών δικτύων (internet) και σε μη κρυπτογραφημένη μορφή (plain text), ενώ ταυτόχρονα μπορεί να υποδεικνύει ότι οι κωδικοί πρόσβασης αποθηκεύονται σε μη κατακερματισμένη (hashed) μορφή.

Για την επίλυση τέτοιου είδους προβλημάτων, σε αυτή την εφαρμογή διαχείρισης, χρησιμοποιείται ένα σύστημα επαναφοράς του κωδικού πρόσβασης. Σε πρώτο στάδιο, ο χρήστης ζητείται να συμπληρώσει το όνομα χρήστη του λογαριασμού του, και στην συνέχεια του αποστέλλεται ένα μήνυμα ηλεκτρονικού ταχυδρομείου (email) στην διεύθυνση επικοινωνίας που είχε ορίσει στο προφίλ του, το οποίο περιέχει έναν σύνδεσμο (link) με ένα μοναδικό αναγνωριστικό επαναφοράς (reset token), το οποίο έχει καταχωρηθεί σε σχετικό πεδίο (field) στον πίνακα `User` της βάσης δεδομένων.

Αφού ο χρήστης ακολουθήσει τον σύνδεσμο, μεταφέρεται σε κατάλληλη σελίδα της εφαρμογής, όπου και ελέγχεται εάν το αναγνωριστικό επαναφοράς υφίσταται και είναι έγκυρο. Σε αυτή την περίπτωση, παρουσιάζεται στον χρήστη μία φόρμα, όπου του επιτρέπεται να καταχωρήσει έναν νέο κωδικό πρόσβασης, και με αυτό τον τρόπο επιτυγχάνεται η επαναφορά του κωδικού πρόσβασης. Να σημειωθεί ότι μετά την επιτυχή επαναφορά του κωδικού πρόσβασης, το σχετικό αναγνωριστικό (token) διαγράφεται από την βάση δεδομένων, ούτως ώστε να μην είναι εφικτή η εκ νέου επαναφορά του κωδικού πρόσβασης με την χρήση του ίδιου αναγνωριστικού.

Σε περιπτώσεις μη ύπαρξης ή μη έγκυρων αναγνωριστικών επαναφοράς, εμφανίζεται σχετικό μήνυμα σφάλματος, και η διαδικασία επαναφοράς του κωδικού πρόσβασης διακόπτεται.

Δήλωση & προγραμματισμός παραγωγών

Όπως αναφέρθηκε και σε προηγούμενη ενότητα, στα πλαίσια της πλήρους παραμετροποιήσιμης λειτουργίας της εφαρμογής διαχείρισης, οι κάτοχοι διαχειριστικών δικαιωμάτων, μπορούν να ορίσουν διάφορους τύπους παραγωγών. Ορισμένοι από

τους διαθέσιμους τύπους παραγωγών, απαιτούν δήλωση και προγραμματισμό, πριν την μεταφόρτωση και μετάδοση τους.

Η δήλωση & ο προγραμματισμός, είναι μία διαδικασία που αφορά συνήθως τακτικές ή έκτακτες εκπομπές, και ολοκληρώνεται σε τέσσερα στάδια.

Γενικά στοιχεία παραγωγής

Αρχικά ο χρήστης, ζητείται να επιλέξει και να εισάγει διάφορα γενικά στοιχεία για την παραγωγή. Τα δύο πιο βασικά στοιχεία, σε αυτό το στάδιο, είναι ο τύπος της παραγωγής, και η ζώνη. Βάσει αυτών των δύο επιλογών είναι στη συνέχεια δυνατή η δημιουργία του προγράμματος παραγωγών.

Όπως προαναφέρθηκε, η κάθε ζώνη μπορεί να «δεχθεί» συγκεκριμένους τύπους παραγωγών. Έτσι η επιλογή ενός συγκεκριμένου τύπου παραγωγής, εκκινεί μία ασύγχρονη κλήση AJAX, η οποία έχει ως αποτέλεσμα την ενημέρωση της λίστας επιλογών ζώνης. Με αυτό τον τρόπο, ο χρήστης δεν χρειάζεται να ξέρει σε ποια ή ποιες ζώνες μπορεί να δηλώσει την παραγωγή του, μειώνεται η περίπτωση σφάλματος από επιλογή λανθασμένης ζώνης, καθώς επίσης μειώνεται και η συμμετοχή των διαχειριστών στην παρακολούθηση και διόρθωση λανθασμένων δηλώσεων.

Σε αυτό το αρχικό στάδιο, ο χρήστης καλείται επίσης να συμπληρώσει τον τίτλο της παραγωγής του, μία σύντομη περιγραφή καθώς και τις θεματικές ενότητες που καλύπτει η παραγωγή. Επίσης, μπορεί να δηλώσει αν μία παραγωγή είναι ζωντανή ή ηχογραφημένη και την γλώσσα που χρησιμοποιείται.

Τέλος, για ορισμένους τύπους παραγωγών και συγκεκριμένα για τις τακτικές παραγωγές, ο χρήστης επιλέγει την συχνότητα μετάδοσης (εβδομαδιαία, 15/ημερη, μηνιαία) και την επιθυμητή ημερομηνία πρώτης μετάδοσης.

Συνεργασίες

Ένας τομέας όπου το *European School Radio* πρωτοπόρισε και ξεχώρισε από άλλα διαδικτυακά ραδιόφωνα, ήταν αυτός των συνεργασιών. Τα συμμετέχοντα μέλη, έχουν δυνατότητα σύναψης συνεργασιών με μέλη άλλων λογαριασμών, πάνω σε συγκεκριμένες θεματικές ενότητες.

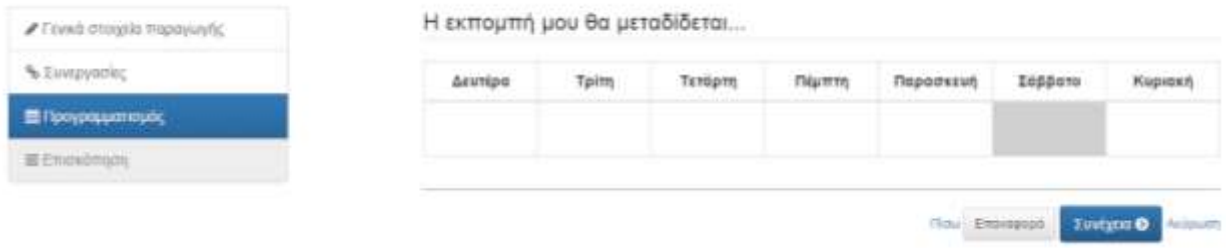
Σε αυτό το στάδιο, δήλωσης μίας παραγωγής, ο χρήστης έχει δυνατότητα επιλογής μίας εκ των συνεργασιών που έχουν ήδη συναφθεί και αφορούν τις θεματικές ενότητες που έχουν επιλεγεί, για την συγκεκριμένη παραγωγή, στο πρώτο στάδιο δήλωσης. Σε περίπτωση που δεν υπάρχουν ενεργές συνεργασίες, το στάδιο αυτό είναι απενεργοποιημένο και παρακάμπτεται.

Η διαδικασία σύναψης συνεργασιών αναλύεται σε επόμενη ενότητα.

Προγραμματισμός

Στο στάδιο του προγραμματισμού, ο χρήστης καλείται να επιλέξει την ημέρα και ώρα που θα μεταδίδεται η συγκεκριμένη παραγωγή.

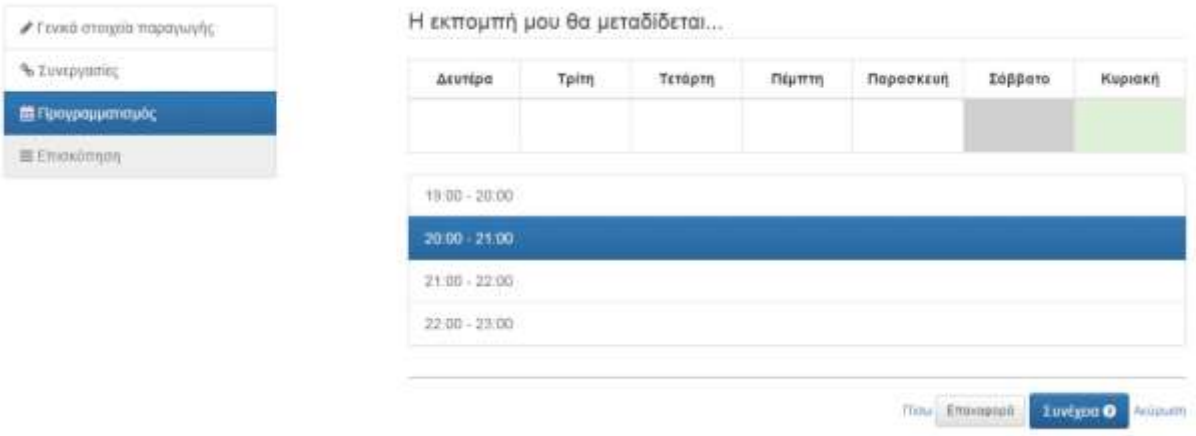
Η διαδικασία ξεκινά με την εμφάνιση ενός πίνακα όπου φαίνονται οι ημέρες όπου «παίζει» η ζώνη που έχει επιλέξει ο χρήστης στο πρώτο στάδιο. Αφού ο χρήστης επιλέξει μία από τις διαθέσιμες ημέρες, εκκινείται μία ασύγχρονη κλήση AJAX, η οποία έχει ως αποτέλεσμα την εμφάνιση ενός δεύτερου πίνακα, όπου εμφανίζονται οι ώρες που σχετίζονται με την επιλεγμένη ζώνη και ημέρα, σε μορφή χρονοθυρίδων (time slots).



Εικόνα 5 - Επιλογή ημέρας μετάδοσης

Σε αυτό το σημείο, να υπενθυμίσουμε ότι το ωρολόγιο πρόγραμμα των ζωνών, προκύπτει από την επιλογή θυρίδων (slots) διάρκειας 30 λεπτών έκαστη. Η επιλογή συγκεκριμένου τύπου παραγωγής από τον χρήστη, στο πρώτο στάδιο, μας δίνει τη δυνατότητα να γνωρίζουμε την επιτρεπόμενη μέγιστη διάρκεια της παραγωγής. Με αυτό τον τρόπο, οι 30λεπτες χρονοθυρίδες της επιλεγμένης ζώνης, μετατρέπονται προγραμματιστικά, σε χρονοθυρίδες επαρκούς διάρκειας, ούτως ώστε να καλύπτουν την διάρκεια του επιλεγμένου τύπου παραγωγής, με αποτέλεσμα ο χρήστης να μην είναι υποχρεωμένος να επιλέξει πολλές χρονοθυρίδες των 30 λεπτών.

Για παράδειγμα, αν ο επιλεγμένος τύπος παραγωγής έχει μέγιστη διάρκεια 60 λεπτών, και η επιλεγμένη ζώνη έχει διάρκεια τεσσάρων ωρών, δηλαδή οκτώ χρονοθυρίδες των 30 λεπτών, τότε οι χρονοθυρίδες αυτές θα μετατραπούν σε τέσσερις χρονοθυρίδες των 60 λεπτών.



Εικόνα 6 - Επιλογή ώρας μετάδοσης

Σε περίπτωση που κάποια από τις χρονοθυρίδες της ζώνης είναι κατηλλειμένη από κάποια άλλη παραγωγή, τότε αυτή αυτόματα απενεργοποιείται και σημειώνεται με κόκκινο χρώμα, ούτως ώστε να μην είναι δυνατή η επιλογή της.

Επισκόπηση

Το τελευταίο στάδιο της δήλωσης και προγραμματισμού παραγωγών, είναι μία επισκόπηση (overview) όλων των επιλογών και των στοιχείων που έχει επιλέξει και εισάγει ο χρήστης. Εδώ μπορεί να ελέγξει όλα τα στοιχεία, να εντοπίσει τυχόν σφάλματα και να επιστρέψει σε προηγούμενα στάδια για να τα διορθώσει ή να συνεχίσει με την αποθήκευση της συγκεκριμένης δήλωσης.

Η αποθήκευση εκκινεί μία διαδικασία, όπου καταχωρούνται όλα τα τα στοιχεία που έχει επιλέξει ή εισάγει ο χρήστης, στην βάση δεδομένων και πιο συγκεκριμένα στον πίνακα `Production`, ενώ ταυτόχρονα γίνεται μία μαζική εισαγωγή (batch insert) στον πίνακα `Booking`, όπου έχουμε ουσιαστικά την κράτηση (reservation) συγκεκριμένης ημέρας και ώρας για την τρέχουσα παραγωγή. Βάσει των δεδομένων του πίνακα `Booking` έχουμε δυνατότητα να γνωρίζουμε ποιες χρονοθυρίδες είναι ελεύθερες και ποιες κατηλειμμένες.

Μεταφόρτωση παραγωγών

Η μεταφόρτωση παραγωγών είναι η διαδικασία που επιτρέπει στους χρήστες της εφαρμογής να αναρτήσουν τις παραγωγές τους.

Σε πρώτο στάδιο, ο χρήστης καλείται να επιλέξει τον τύπο της παραγωγής που πρόκειται να μεταφορτώσει, και συγκεκριμένα, αν πρόκειται για εκπομπή ή σποτ. Βάσει αυτής της επιλογής, εμφανίζεται η κατάλληλη λίστα επιλογών που περιέχει όλους τους διαθέσιμους τύπους εκπομπών και σποτ αντίστοιχα.

Η επιλογή συγκεκριμένου τύπου παραγωγής (εκπομπή ή σποτ) οδηγεί στην εμφάνιση επιπλέον επιλογών, οι οποίες ωστόσο διαφοροποιούνται αναλόγως. Πιο συγκεκριμένα, για τις προγραμματισμένες εκπομπές, ο χρήστης καλείται να επιλέξει μία από τις διαθέσιμες εκπομπές που προηγουμένως έχει δηλώσει μέσω της διαδικασίας δήλωσης και προγραμματισμού, ενώ για μη προγραμματισμένες παραγωγές, πρέπει να πληκτρολογήσει τον επιθυμητό τίτλο της παραγωγής. Αντίστοιχα για τις θεματικές εκπομπές, πλην του τίτλου που θα πρέπει να πληκτρολογήσει, ο χρήστης, πρέπει να επιλέξει και την θεματική ενότητα που αντιστοιχεί στην παραγωγή που πρόκειται να μεταφορτώσει. Ακόμη, για τις μη προγραμματισμένες παραγωγές, υπάρχει δυνατότητα επιλογής της γλώσσας της παραγωγής.

Τέλος, ανεξάρτητα από τον επιλεγμένο τύπο παραγωγής, ο χρήστης έχει στην διάθεσή του, μία περιοχή όπου μπορεί να εναποθέσει (drag ‘n drop) ή να επιλέξει τα αρχεία που επιθυμεί. Τα επιτρεπόμενα αρχεία στην διαδικασία μεταφόρτωσης παραγωγών είναι ένα αρχείο ήχου, και μία προαιρετική εικόνα η οποία χρησιμοποιείται στην προβολή του αρχείου των παραγωγών (podcast).

Όταν ο χρήστης συμπληρώσει όλα τα απαιτούμενα στοιχεία, και προβεί στην αποθήκευση της παραγωγής, τα αρχεία του μεταφορτώνονται, αρχικά σε μία προσωρινή τοποθεσία (`uploads/tmp`) και στη συνέχεια υπόκεινται σε μία διαδικασία επιβεβαίωσης του τύπου τους, βάσει της επέκτασης του αρχείου (extension) αλλά και βάσει του τύπου MIME, ενώ τα αρχεία ήχου ελέγχονται επιπλέον για την μέγιστη διάρκειά τους, με χρήση της βιβλιοθήκης getID. Τέλος τα αρχεία που έχουν μεταφορτωθεί μετονομάζονται ανάλογα με τον τύπο της επιλεγμένης παραγωγής και βάσει των στοιχείων που έχει εισάγει ο χρήστης. Αφού ολοκληρωθεί η διαδικασία ελέγχου επιτυχώς, τα αρχεία μετακινούνται στον τελικό φάκελο αποθήκευσης, ο οποίος επίσης διαφοροποιείται ανάλογα με τον επιλεγμένο τύπο παραγωγής.

Στην ακόλουθη ενότητα γίνεται εκτενής ανάλυση του ασύγχρονου μηχανισμού μεταφόρτωσης στην πλευρά του χρήστη (client side).

Μηχανισμός μεταφόρτωσης (upload)

Η απαίτηση για εκσυγχρονισμό της εφαρμογής διαχείρισης, καθώς και η ανάγκη διαχείρισης αρχείων μεγάλου όγκου (αρχεία ήχου με μέγεθος εκατοντάδες MiB), οδήγησαν στην ανάπτυξη ενός μηχανισμού ασύγχρονης μεταφόρτωσης, υλοποιημένου εξ ολοκλήρου με javascript και το framework jQuery, κάνοντας παράλληλα χρήση των API File, FileList και Drag and Drop.

Ο μηχανισμός μεταφόρτωσης (`uploader.class.js`), υλοποιημένος ως πρόσθετο (plugin) του jQuery, προσφέρει ένα σύνολο επιλογών, το οποίο το καθιστούν

αρκετά ευέλικτο, και προσαρμόσιμο σε ένα σύνολο διαφορετικών υλοποιήσεων. Το πρόσθετο αυτό, αναλαμβάνει την διαχείριση όλων των events επιλογής αρχείων (drag and drop ή επιλογή αρχείου), τον έλεγχό τους, αναφορικά με τον τύπο και το μέγεθός τους και την ασύγχρονη μεταφόρτωσή τους.

Η ασύγχρονη μεταφόρτωση είναι μία διαδικασία που επιτρέπει στον χρήστη να συνεχίσει την εργασία του στην εφαρμογή, χωρίς να χρειάζεται να περιμένει για την ολοκλήρωση της μεταφόρτωσης, η οποία ανάλογα με το μέγεθος του αρχείου και την ταχύτητας της σύνδεσης που έχει ο χρήστης στο διαδίκτυο, μπορεί να διαρκέσει αρκετά λεπτά. Παράλληλα μέσω του αντικειμένου `XMLHttpRequest`, το οποίο διαχειρίζεται τις ασύγχρονες κλήσεις προς τον εξυπηρετητή της εφαρμογής αλλά και το επιστρεφόμενο σύνολο δεδομένων (response), υπάρχει δυνατότητα παρακολούθησης της πρόοδου της μεταφόρτωσης, με την διαχείριση του event `progress`.

Ο μηχανισμός μεταφόρτωσης, αρχικά στην μέθοδο `init()`, ορίζει τις μεθόδους που θα παρακολουθούν και θα διαχειρίζονται τα events επιλογής αρχείων. Συγκεκριμένα μας ενδιαφέρουν το event `onDrop` στην περιοχή που έχουμε ορίσει για την εναπόθεση αρχείων (`_dropzone`) και το event `onChange` στο στοιχείο επιλογής αρχείων (`input[type=file]`). Επίσης υπάρχει δυνατότητα παρακολούθησης ενός επιπλέον event, το οποίο ορίζεται παραμετρικά μέσω των ρυθμίσεων του πρόσθετου, και αποσκοπεί στην αυτόματη εκκίνηση της διαδικασίας μεταφόρτωσης.

Τα events `onDrop` και `onChange` με την σειρά τους καλούν την μέθοδο `_collectFiles()`, η οποία εκκινεί μία επανάληψη στο σύνολο των αρχείων, ασχέτως της προέλευσής τους (drag ‘n drop ή επιλογή), ελέγχει αν υπάρχουν ήδη στην λίστα επιλεγμένων αρχείων, οπότε και τα απορρίπτει αυτόματα, επιβεβαιώνει πως επιτρέπεται η επιλογή επιπλέον αρχείων (`maxFilesAllowed`) και τέλος καλεί την μέθοδο `_verify()` ή εμφανίζει σχετικό μήνυμα σφάλματος, σε περίπτωση υπέρβασης του μέγιστου ορίου επιτρεπτού αριθμού αρχείων.

Η μέθοδος `_verify()` αναλαμβάνει τον έλεγχο του εκάστοτε αρχείου, χρησιμοποιώντας το File API. Με την χρήση του File API η μέθοδος `_verify()` έχει πρόσβαση σε βασικά στοιχεία του αρχείου, όπως ο τύπος του και το μέγεθός του. Ο έλεγχος πραγματοποιείται με χρήση κανόνων αντιστοίχισης απέναντι σε παραμετροποιήσιμες ρυθμίσεις λίστας επιτρεπόμενων τύπων αρχείων και μέγιστου μεγέθους αρχείου. Ακόμη η μέθοδος `_verify()` μπορεί να ρυθμιστεί ώστε να επιτρέπει συγκεκριμένο αριθμό αρχείων ανάλογα με τον τύπο τους, περίπτωση όπου ο περιορισμός του συνολικού αριθμού αρχείων δεν θα επαρκούσε. Αν οποιοσδήποτε έλεγχος αποτύχει, εμφανίζεται σχετικό μήνυμα σφάλματος, και το σχετικό αρχείο απορρίπτεται, διαφορετικά προστίθεται στην λίστα επιλεγμένων αρχείων (`var _files`).

Η ολοκλήρωση της επανάληψης στη μέθοδο `_collectFiles()`, οδηγεί στην κλήση της `_updateFilesList()`, η οποία είναι υπεύθυνη για την εμφάνιση των επιλεγμένων αρχείων.

Τέλος η μέθοδος `_upload()`, κάνοντας χρήση του FormData API, εκκινεί μία ασύγχρονη κλήση AJAX, η οποία μεταφορτώνει τα αρχεία μαζί με άλλα δεδομένα που πιθανά έχουν ορισθεί, χρησιμοποιώντας το script που έχει δηλωθεί μέσω των ρυθμίσεων του πρόσθετου, ενώ παράλληλα προβάλει την πρόοδο της όλης διαδικασίας μέσω της παρακολούθησης του event `progress`. Τυχόν σφάλματα από την μεταφόρτωση των αρχείων, εμφανίζονται και πρέπει να επιλυθούν πριν επιτραπεί η εκ νέου εκκίνηση της μεταφόρτωσης.

Ιστορικό παραγωγών

Στο ιστορικό παραγωγών, αναρτώνται όλες οι παραγωγές που έχουν μεταφορτωθεί στα πλαίσια ενός συγκεκριμένου λογαριασμού και υπάρχει δυνατότητα λήψης (download) του αρχείου ήχου ή αναπαραγωγής του. Η τεχνολογία και ο τρόπος με τον οποίο αυτή χρησιμοποιείται για την αναπαραγωγή των αρχείων, αναλύεται σε επόμενη ενότητα.

Συνεργασίες

Όπως αναφέρθηκε και σε προηγούμενη ενότητα, ένα από τα στοιχεία που διαφοροποιούν το *European School Radio* από άλλα διαδικτυακά ραδιόφωνα είναι η δυνατότητα σύναψης συνεργασιών.

Μία συνεργασία μπορεί να συναφθεί μεταξύ αρκετών συμμετεχόντων μελών και αφορά κάποια συγκεκριμένη θεματική ενότητα. Οι συνεργασίες συνάπτονται σε επίπεδο λογαριασμού.

Σε αυτό το σημείο θα πρέπει να υπενθυμίσουμε, ότι κάθε λογαριασμός στα πλαίσια της συμμετοχής του, έχει ένα επιλεγμένο σύνολο από θεματικές ενότητες θέματα των οποίων αναπτύσσονται στις παραγωγές του εν λόγω λογαριασμού. Για αυτές τις επιλεγμένες θεματικές ενότητες τα μέλη του λογαριασμού μπορούν να δηλώσουν επίσης, πρόθεση συνεργασίας για την κάθε μία ξεχωριστά, γνωστοποιόντας με αυτό τον τρόπο ότι είναι διατεθημένοι να προχωρήσουν στην σύναψη μίας συνεργασίας για την εκάστοτε θεματική ενότητα.

Ο χρήστης που θα εκκινήσει την διαδικασία σύναψης μίας νέας συνεργασίας, επιλέγει την επιθυμητή θεματική ενότητα, και στη συνέχεια μπορεί να ορίσει επιπλέον φίλτρα όπως για παράδειγμα την περιοχή ή την εκπαιδευτική βαθμίδα των λογαριασμών με τους οποίους επιθυμεί να συνεργαστεί.

Η διαδικασία ολοκληρώνεται με αναζήτηση στο σύνολο των λογαριασμών που έχουν στην θεματολογία τους την επιλεγμένη θεματική ενότητα και επιπλέον έχουν δηλώσει πρόθεση συνεργασίας. Ο χρήστης στην συνέχεια μπορεί να αιτηθεί συνεργασίας, σε κάποιον από τους αναγραφόμενους λογαριασμούς που εμφανίζονται στην λίστα με τα αποτελέσματα της αναζήτησης.

Η διαδικασία αίτησης συνεργασίας, περιλαμβάνει την αυτόματη αποστολή ενός μηνύματος ηλεκτρονικού ταχυδρομείου (email) και την προβολή σχετικής ειδοποίησης στα προφίλ των χρηστών του λογαριασμού που έχει λάβει την αίτηση συνεργασίας.

Για να ολοκληρωθεί η σύναψη της συνεργασίας, πρέπει κάποιος από τους χρήστες του λογαριασμού, να αποδεχθεί την αίτηση συνεργασίας, ενώ ταυτόχρονα έχει την δυνατότητα να απορρίψει την αίτηση αυτή. Μέχρι να ολοκληρωθεί η σύναψη, είτε με την αποδοχή ή την απόρριψή της, η συνεργασία βρίσκεται σε κατάσταση αναμονής και δεν είναι δυνατή η χρήση της στην διαδικασία δήλωσης και προγραμματισμού.

Φυσικά οποιοδήποτε μέλος των λογαριασμών που μετέχουν σε μία συνεργασία, μπορεί να αποχωρήσει από την συνεργάσια και ο αρχικός δημιουργός της μπορεί να την τερματίσει εξ ολοκλήρου.

Διαχείριση λογαριασμού

Κάθε μέλος ενός λογαριασμού, μπορεί να προχωρήσει σε διάφορες ρυθμίσεις και διαχειριστικές ενέργειες που αφορούν το προσωπικό του προφίλ, την συμμετοχή στο *European School Radio*, τους άλλους χρήστες του ίδιου λογαριασμού και σε συγκεκριμένους τύπους λογαριασμού, μπορεί να έχει πρόσβαση και στα στοιχεία του σχολείου.

Προσωπικό προφίλ

Οι χρήστες των λογαριασμών, έχουν το δικό τους, προσωπικό προφίλ, όπου μπορούν να προβάλλουν και να διαχειριστούν τα προσωπικά τους στοιχεία, τα στοιχεία επικοινωνίας, ενώ χρήστες που έχουν επιλέξει ως μέθοδο πιστοποίησης την τοπική και όχι την υπηρεσία CAS, έχουν επίσης δυνατότητα μεταβολής του ονόματος χρήστη (username) και του κωδικού πρόσβασης (password).

Επιπλέον οι υπεύθυνοι σχολείων, μέσω του προσωπικού τους προφίλ, μπορούν να ανανεώσουν τους εκπαιδευτικούς κλάδους και τις ειδικότητες στις οποίες ανήκουν.

Συμμετοχή

Στις ρυθμίσεις της συμμετοχής, οι χρήστες του λογαριασμού, μπορούν να προσαρμόσουν την περιγραφή της συμμετοχής τους και να προχωρήσουν στην

μεταφόρτωση (upload) ενός εικονιδίου (banner), που λειτουργεί σαν αναγνωριστικό λογότυπο της συμμετοχής, και ενός ηχητικού σποτ για την συμμετοχή.

Επίσης οι χρήστες του λογαριασμού, μέσω των ρυθμίσεων της συμμετοχής, μπορούν να τροποποιήσουν την θεματολογία του λογαριασμού και να αλλάξουν τις προθέσεις συνεργασίας για κάθε θεματική ενότητα.

Σε αυτό το σημείο θα πρέπει να σημειωθεί, πως η σελίδα της συμμετοχής θα πρέπει να είναι συμπληρωμένη σε όλο της το φάσμα (περιγραφή, banner, spot και θεματολογία), διαφορετικά ο λογαριασμός θεωρείται ανενεργός και δεν είναι δυνατή η δήλωση και η μεταφόρτωση εκπομπών.

Χρήστες

Η λίστα των χρηστών, προβάλει το σύνολο των ατόμων που έχουν πρόσβαση στον κάθε λογαριασμό. Συγκεκριμένοι χρήστες του λογαριασμού, οι οποίοι κατέχουν τα κατάλληλα δικαιώματα, μπορούν να προσθέσουν νέους και να επεξεργαστούν ή να διαγράψουν ήδη υπάρχοντες χρήστες.

Σχολείο

Οι ρυθμίσεις σχολείου, είναι διαθέσιμες μόνο σε λογαριασμούς τύπου σχολείου, και όχι σε μαθητές ή εξωτερικούς συνεργάτες. Οι χρήστες του λογαριασμού έχουν δυνατότητα προβολής και τροποποίησης των στοιχείων του σχολείου, δηλαδή της ονομασίας, της εκπαιδευτικής του βαθμίδας καθώς και ορισμένων σχολίων. Επίσης υπάρχει δυνατότητα μεταβολής των στοιχείων διεύθυνσης και επικοινωνίας.

Βάσει της ταχυδρομικής διεύθυνσης που θα πληκτρολογήσει ο χρήστης, ο χάρτης τοποθεσίας ενημερώνεται αυτόματα, ενώ υπάρχει δυνατότητα μετατόπισης του δείκτη του χάρτη (map marker) σε διαφορετικό σημείο, σε περίπτωση που η αυτόματη τοποθέτηση αποτύχει ή δεν είναι απόλυτα ακριβής.

Αρχείο παραγωγών (podcast)

Το αρχείο παραγωγών είναι ένας κατάλογος του συνόλου των παραγωγών που έχουν αναρτηθεί στην εφαρμογή από τους χρήστες όλων των λογαριασμών. Αυτό είναι και το κύριο σημείο όπου το αρχείο παραγωγών (podcast) διαφοροποιείται από το ιστορικό, που αναφέρθηκε νωρίτερα. Να υπενθυμίσουμε ότι στο ιστορικό, εμφανίζονται μόνο οι παραγωγές του εκάστοτε λογαριασμού.

Το αρχείο παραγωγών, διαφέρει επίσης από το ιστορικό, σε θέματα παρουσίασης, αφού εμφανίζει περισσότερα στοιχεία για την κάθε παραγωγή.

Επίσης, στο αρχείο παραγωγών, οι χρήστες έχουν δυνατότητα αναζήτησης με χρήση ενός πλήθους διαφορετικών φίλτρων.

Αναζήτηση με χρήση φίλτρων

Η διαδικασία αναζήτησης επιτυγχάνεται με την χρήση μίας φόρμας όπου οι χρήστες έχουν στην διάθεσή τους ένα σύνολο από διάφορα φίλτρα όπως τίτλος, ζώνη ή τύπος παραγωγής, ημερομηνία μετάδοσης και άλλα που αφορούν την παραγωγή αλλά και την συμμετοχή.

Η αποστολή της φόρμας από τον χρήστη, έχει ως αποτέλεσμα την δημιουργία μίας πρότασης αναζήτησης (search clause), το οποίο και μεταφέρεται παραμετρικά στην αντίστοιχη μέθοδο του σχετικού model, και εκεί χρησιμοποιείται ως μέρος του τμήματος `WHERE` του συνολικού ερωτήματος (query).

Παράλληλα το σύνολο των χρησιμοποιημένων φίλτρων και των τιμών των αντίστοιχων πεδίων, αποθηκευόνται στην σύνοδο σύνδεσης (session) ούτως ώστε να είναι δυνατή η επαναδημιουργία της πρότασης αναζήτησης (search clause) και σε επόμενα αιτήματα. Επιπλέον, η δυνατότητα επαναδημιουργίας της πρότασης αναζήτησης, χωρίς να απαιτείται η εκ νέου συμπλήρωση και αποστολή της φόρμας από τους χρήστες, επιτρέπει και την σελιδοποίηση των αποτελεσμάτων (pagination).

Αναπαραγωγή αρχείων ήχου

Σε ορισμένα σύγχρονα προγράμματα περιήγησης (browsers) υποστηρίζεται η αναπαραγωγή αρχείων ήχου από ενσωματωμένους μηχανισμούς αναπαραγωγής, μέσω της χρήσης της ετικέτας (tag) `audio` που είναι διαθέσιμη στο πρότυπο HTML5 και έχει γίνει προσπάθεια εκμετάλευσης αυτής της νέας δυνατότητας, ούτως ώστε να μην είναι αναγκασμένοι οι χρήστες να κάνουν λήψη ενός αρχείου πριν προβούν στην αναπαραγωγή του. Ωστόσο, σε παλαιότερα πρόγραμματα περιήγησης ή σε κάποιες κινητές συσκευές, όπου η λειτουργία ενσωμματωμένης αναπαραγωγής δεν υποστηρίζεται, ο χρήστης παρακινείται να κάνει λήψη του αρχείου.

Να σημειωθεί, πως η ετικέτα `audio`, περιέχει τουλάχιστον μία ετικέτα `source`. Οι ετικέτες `source`, χρησιμοποιούνται για την δήλωση των αρχείων ήχου. Μία ετικέτα `audio`, μπορεί να περιέχει πολλαπλές ετικέτες `source`, επιτρέποντας με αυτό τον τρόπο, την φόρτωση του ίδιου αρχείου ήχου με διαφορετικούς τύπους κωδικοποίησης (MP3 / OGG / WAV κ.λπ.).

Η διαδικασία της αναπαραγωγής χρησιμοποιεί έναν συνδυασμό APIs του προτύπου HTML5, Javascript και PHP. Η επιλογή (κλικ) μίας συγκεκριμένης παραγωγής, εκκινεί μία σειρά από events, τα οποία και διαχειρίζονται μέσω JavaScript, και πιο συγκεκριμένα μέσω jQuery.

Ουσιαστικά η επιλογή του συνδέσμου (link) αναπαραγωγής, ανοίγει ένα modal παράθυρο, το οποίο περιέχει την ετικέτα `audio`, μέσω της οποίας το πρόγραμμα περιήγησης (browser), διαχειρίζεται την αναπαραγωγή του αρχείου ήχου που έχει οριστεί μέσω των ετικετών `source`. Η πηγή (src) του αρχείου ήχου που δηλώνεται στην ετικέτα source, αλλάζει δυναμικά, ούτως ώστε να αντιστοιχεί στην τιμή του χαρακτηριστικού (attribute) href του συνδέσμου.

Εν συνεχεία, καλούνται δύο μέθοδοι του audio API, που ανήκει στο πρότυπο HTML5. Η πρώτη μέθοδος είναι η `load()`, και είναι αυτή που αναλαμβάνει την φόρτωση των δεδομένων του αρχείου ήχου, και ακολούθως καλείται η `play()` η οποία είναι υπέυθυνη για την αναπαραγωγή του αρχείου ήχου.

Η διαδικασία αυτή, με την δυναμική αλλαγή της πηγής του αρχείου ήχου, ακολουθείται ούτως ώστε, να μην είναι ορατή η πραγματική θέση του αρχείου ήχου, στο σύστημα αρχείων του εξυπηρετητή (server) της εφαρμογής. Πιο συγκεκριμένα, ο σύνδεσμος αναπαραγωγής, οδηγεί στη μέθοδο `listen()` του controller `Productions`, και παρέχει ως παράμετρο το μοναδικό αναγνωριστικό (ID) της εκάστοτε παραγωγής.

Στην μέθοδο `listen()`, γίνεται ανάκτηση των στοιχείων της παραγωγής, με χρήση του μοναδικού αναγνωριστικού (ID). Για λειτουργικά συστήματα Microsoft™ Windows, ακολουθείται μία διαδικασία μετατροπής της διαδρομής (path), όπου είναι αποθηκευμένο το αρχείο ήχου, σε κατάλληλη κωδικοποιήση, και πιο συγκεκριμένα από `UTF-8` σε `Windows-1253`. Στην συνέχεια, γίνεται φόρτωση και έξοδος (output) του αρχείου ήχου, με χρήση του URL wrapper `file:///`, που επιτρέπει την φόρτωση αρχείων από το τοπικό σύστημα αρχείων, και με χρήση διάφορων κεφαλίδων (headers), που χρησιμοποιούνται για να δηλώσουν τον τύπο και το μέγεθος των μεταφερόμενων δεδομένων (audio).

Το πρόγραμμα περιήγησης (browser), είναι αυτό που θα αναλάβει να ερμηνεύσει το σύνολο των δεδομένων που επιστρέφονται από τον εξυπηρετητή (server) της εφαρμογής, και να τα δρομολογήσει προς αναπαραγωγή, μέσω της ετικέτας `audio`.

```
header('Content-Type: audio/mpeg');

header('Content-Disposition:
        inline;filename="'.basename("file:///$link").'"');

header('Content-length: ' . filesize("file:///$link"));

header("Content-Transfer-Encoding: inline");
```

Εικόνα 7 - Κεφαλίδες (headers) για την αναπαραγωγή του αρχείου ήχου

Στοιχεία ακροαματικότητας

Η ακροαματικότητα είναι μία τιμή που υποδηλώνει το σύνολο των ακροατών, δηλαδή των ενεργών συνδέσεων στον server μετάδοσης (broadcast server), σε μία συγκεκριμένη χρονική στιγμή. Είναι ένας σημαντικός παράγοντας, που αν αναλυθεί στατιστικά, μπορεί να καθορίσει την δομή που θα έχουν οι ζώνες παραγωγών, τις ώρες που θα μεταδίδονται οι διάφοροι τύποι παραγωγών, ενώ σε εμπορικά περιβάλλοντα, μπορεί να καθορίσει για παράδειγμα την ώρα που είναι πιο αποδοτική η μετάδοση διαφημίσεων ή ανακοινώσεων σχετικά με κάποια διοργάνωση ή σχετικό event.

Το *European School Radio* έχει μία ιδιαίτεροτητα ως προς τον υπολογισμό της ακροαματικότητας. Οι ακροατές του ραδιοφώνου μπορούν να συντονιστούν και να ακούσουν το πρόγραμμα από τρεις (3) διαφορετικές πηγές, την υπηρεσία VOD του Πανελλήνιου Σχολικού Δικτύου, και δύο streams που μεταδίδονται από το δίκτυο του Αλεξάνδρειου ΤΕΙ Θεσσαλονίκης (64 και 128kbps αντίστοιχα).

Στον server όπου φιλοξενείται η εφαρμογή διαχείρισης, έχουν ρυθμιστεί προγραμματισμένες εργασίες (Scheduled Tasks σε λειτουργικά συστήματα Windows ή Cron Jobs αντίστοιχα σε λειτουργικά συστήματα UNIX), οι οποίες αναλαμβάνουν την εκτέλεση του controller `Ratings`, κάθε 2 λεπτά. Ο controller `Ratings`, είναι υπεύθυνος για την ανάκτηση στατιστικών δεδομένων από την υπηρεσία VOD αλλά και από τον server μετάδοσης (icecast), την εξαγωγή των απαιτούμενων δεδομένων, δηλαδή του αριθμού των ενεργών ακροατών, και στη συνέχεια την αποθήκευση του συνόλου των ακροατών στην βάση δεδομένων, και συγκεκριμένα στον πίνακα `rating`.

Το χρονικό περιθώριο των δύο λεπτών, αν και αρχικά φαίνεται πλήρως αυθαίρετο, προκύπτει από την γενικότερη δομή του ραδιοφώνου και συγκεκριμένα των παραγωγών του. Με βάση την ελάχιστη διάρκεια μίας παραγωγής, που συνήθως είναι ένα λεπτό, κρίθηκε υπερβολική η ανάκτηση δεδομένων σε διάστημα μικρότερο του ενός λεπτού, και εν τέλει αποφασίσθηκε η χρήση χρονικού περιθωρίου δύο λεπτών, το οποίο είναι και αυτό που προσφέρει το πιο χρήσιμο σύνολο πληροφορίας.

Ενοποίηση με WordPress & ανάκτηση περιεχομένου

Το WordPress είναι μία από τις πιο γνωστές πλατφόρμες ιστολογίου (blog) σε παγκόσμια κλίμακα. Η διαχειριστική ομάδα του *European School Radio,* στα πλαίσια λειτουργίας του, προχώρησε στην εγκατάσταση του WordPress για να καλύψει διάφορες ανάγκες δημοσίευσης ανακοινώσεων και διάφορων άλλων στοιχείων, όπως νομικά θέματα λειτουργίας ή σελίδες βοήθειας.

Το σύνολο αυτού του περιεχομένου, ζητήθηκε να είναι διαθέσιμο στις σελίδες της τρέχουσας εφαρμογής, αλλά η διαχείριση του να συνεχίσει να γίνεται μέσω του

WordPress. Για αυτό το λόγο, χρησιμοποιήθηκε μία γνωστή πρακτική, που προτείνεται από την κοινότητα χρηστών του WordPress, και έχει ως σκοπό την ανάκτηση και προβολή περιεχομένου εκτός της πλατφόρμας του Wordpress.

Για να επιτευχθεί η ενοποίηση με το WordPress, πρέπει να συμπεριληφθεί (include), στον κώδικα συγκεκριμένων σελίδων της εφαρμογής διαχείρισης, το αρχείο `wp-load.php`, το οποίο είναι υπεύθυνο για την φόρτωση και αρχικοποίηση άλλων αρχείων και διαφόρων παραμέτρων. Με την προσθήκη αυτού του αρχείου στις σελίδες που απαιτείται, είναι δυνατή η χρήση όλου του API της πλατφόρμας WordPress και συγκεκριμένα του loop.

WordPress loop

Η loop είναι μία δομή, ένα σύνολο κώδικα, που επιτρέπει στο WordPress να παρουσιάζει το περιεχομένο των δημοσιευμένων άρθρων. Πιο συγκεκριμένα η χρήση του loop, επιτρέπει την επεξεργασία του κάθε άρθρου μεμονωμένα και την μορφοποίησή του σύμφωνα με τυχόν κώδικα HTML ή PHP που έχουν προστεθεί και που εκτελείται για κάθε άρθρο.

Η χρήση του loop είναι πολύ απλή διαδικασία, και απαιτεί μόνο λίγες γραμμές κώδικα. Αρχικά γίνεται ένας έλεγχος για την ύπαρξη άρθρων. Εφόσον υπάρχουν δημοσιευμένα άρθρα, εκκινείται μία επανάληψη, στον κάθε βρόγχο της οποίας, καλείται η `the_post()`, η οποία είναι μία μέθοδος, υπεύθυνη για την ανάκτηση του επόμενου άρθρου και την ρύθμιση των εσωτερικών δεικτών (indexes) της επανάληψης.

```
if (have_posts()) {
      while (have_posts()) {
            the_post();
            the_title();
            the_content();
      }
}
```

Εικόνα 8 - Βασική χρήση του WordPress loop

Φυσικά με χρήση του πλούσιου API της πλατφόρμας WordPress μπορούμε να προχωρήσουμε στην ανάκτηση και μορφοποίηση διαφόρων τμημάτων και πληροφοριών για κάθε άρθρο. Για παράδειγμα, στην *Εικόνα 7* φαίνεται η χρήση δύο μεθόδων, της `the_title()` και της `the_content()`, οι οποίες είναι υπεύθυνες για την ανάκτηση του τίτλου και του περιεχομένου ενός άρθρου αντίστοιχα.

Η ενοποίηση του CodeIgniter με το WordPress δημιούργησε πρόβλημα «σύγκρουσης» της μεθόδου `site_url()`, η οποία είναι ορισμένη και στον πυρήνα του CodeIgniter, αλλά ταυτόχρονα είναι και μία από τις βασικές μεθόδους του WordPress.

Το πρόβλημα επαναπροσδιορισμού των δύο μεθόδων επιδιορθώθηκε με μία μικρή μεταβολή του αρχείου `link-template.php` στον φάκελο `wp-includes`. Στο σημείο που γίνεται η δήλωση της μεθόδου `site_url()` έχει προστεθεί μία συνθήκη (if block) όπου ελέγχεται αν η μέθοδος έχει ήδη ορισθεί `function_exists()`.

Σε περίπτωση όπου η μέθοδος υπάρχει ήδη, δηλαδή σε περιπτώσεις όπου γίνεται φόρτωση του wordpress εντός της εφαρμογής διαχείρισης, η συνθήκη αποτυγχάνει, και δεν ολοκληρώνεται η εκ νέου φόρτωση της `site_url()`, επομένως δεν δημιουργείται σφάλμα επαναπροσδιορισμού της μεθόδου. Αντίθετα σε περιπτώσεις όπου το WordPress λειτουργεί ως ξεχωριστή «οντότητα» και όχι ως μέρος της εφαρμογής διαχείρισης, η συνθήκη επαληθεύεται, και η μέθοδος `site_url()` δηλώνεται κανόνικα επιτρέποντας την απρόσκοπτη λειτουργία της πλατφόρμας.

Κεφάλαιο 4 – Μελλοντικές επεκτάσεις

Ενσωμάτωση μέσων κοινωνικής δικτύωσης

Υπάρχει μία γενικότερη κατεύθυνση προς το Web 2.0, δηλαδή εφαρμογές και ιστοσελίδες, οι οποίες δίνουν ιδιαίτερη έμφαση στο περιεχόμενο που δημιουργείται από τους χρήστες, καθώς και στην κοινωνική διάσταση που έχει αυτό το περιεχόμενο (social).

Η εφαρμογή διαχείρισης που έχει αναπτυχθεί στα πλαίσια αυτής της Πτυχιακής Εργασίας, θα πρέπει σε επόμενο στάδιο να ενσωματώσει λειτουργίες που αφορούν την αλληλεπίδραση με μέσα κοινωνικής δικτύωσης (social media), ώστε να προσφέρει ολοένα και πιο αρεστό επίπεδο λειτουργικότητας σε άτομα νεότερης ηλικίας (κύριοι χρήστες μέσων κοινωνικής δικτύωσης), ενώ παράλληλα θα αυξήσει το σύνολο του στοχευμένου κοινού (target audience) και αναφορικά με τους ενεργούς χρήστες, αλλά και με το σύνολο των ακροατών.

Η ενοποίηση με μέσα κοινωνικής δικτύωσης (π.χ. Facebook/Twitter κ.λπ.) θα μπορούσε να αφορά σε πρώτη φάση, την δυνατότητα κοινοποίησης των επερχόμενων εκπομπών, και σε επόμενη φάση, την δυνατότητα αποστολής σχολίων (comments) ή ακόμη και την χρήση μηχανισμών πιστοποίησης (oAuth) που προσφέρονται από τα μέσα κοινωνικής δικτύωσης.

Η κοινοποίηση επερχόμενων εκπομπών θα μπορούσε να λειτουργήσει θετικά στην αύξηση του κοινού ακροατών των εκπομπών που μεταδίδονται στο *European School Radio*. Παράλληλα η αύξηση των σημείων (πηγών) όπου παρουσίαζεται το *European School Radio*, θα είχε ως έμμεσο αποτέλεσμα, και την αύξηση των συμμετεχόντων μελών, αφού θα γνωστοποιούνταν σε μεγαλύτερο κοινό η ύπαρξη και ο σκοπός του. Σε παρόμοια κατεύθυνση και με ίδια ή παρόμοια αποτελέσματα, θα λειτουργούσε και η ένταξη λειτουργίας αποστολής σχολίων.

Η χρήση της λειτουργίας πιστοποίησης που παρέχεται από τα μέσα κοινωνικής δικτύωσης, μέσω του μηχανισμού oAuth ή άλλων παρόμοιων μηχανισμών, θα μπορούσε να οδηγήσει, στην αναίρεση της ανάγκης χρήσης ενός τοπικού μηχανισμού πιστοποίησης επομένως και στην διακοπή διατήρησης στοιχείων σύνδεσης των χρηστών. Με αυτό τον τρόπο, αποφεύγεται η χρήση πολύπλοκων και χρονοβόρων μηχανισμών τοπικής πιστοποίησης, και ελαχιστοποιείται η ανάγκη ύπαρξης άλλων, γενικότερων μηχανισμών ασφαλείας. Τέλος, οι χρήστες δεν είναι υποχρεωμένοι να απομνημονεύουν τα στοιχεία σύνδεσης ενός ακόμη λογαριασμού.

Διεπαφή χρήστη (UI)

Ένας ακόμη τομέας όπου θα μπορούσε να βελτιωθεί ή ακόμη και να εκσυγχρονιστεί περισσότερο η τρέχουσα εφαρμογή διαχείρισης, είναι η διεπαφή χρήσης (UI). Κάποια τμήματα της εφαρμογής, και κατά κύριο λόγο οι φόρμες εισαγωγής στοιχείων, θα πρέπει να ελαχιστοποιηθούν σε μέγεθος, με εισαγωγή ακόμη μεγαλύτερου επιπέδου αυτοματισμού και επακόλουθη χρήση λιγότερων πεδίων, ούτως ώστε να είναι πιο εύκολη, περισσότερο κατανοητή και ταχύτερη η εισαγωγή και επεξεργασία στοιχείων από τους χρήστες.

Επιπλέον κάποια λεκτικά, όπως ετικέτες πεδίων, κείμενα βοήθειας και μηνύματα, πληροφοριακά ή ειδοποιήσεις σφαλμάτων, πιθανώς να χρειαστεί να μεταβληθούν, ούτως ώστε να γίνει πιο κατανοητή η σημασία τους και το μήνυμα που προσπαθούν να διοχετεύσουν προς τους χρήστες της εφαρμογής. Μία τέτοια εργασία ωστόσο απαιτεί την προηγούμενη λειτουργία της εφαρμογής, έστω και σε δοκιμαστικό περιβάλλον, με ένα δείγμα χρηστών, με σκοπό να σημειωθούν τα σημεία όπου οι χρήστες εντοπίζουν προβλήματα ή ξοδεύουν περισσότερο χρόνο, και στη συνέχεια να διορθωθούν.

Παράλληλα η συνολική δομή (layout) της εφαρμογής θα μπορούσε να μεταβληθεί με σκοπό να προσφέρει μία πιο φιλική διεπαφή χρήσης, δεδομένου και του συνόλου μελών που την χρησιμοποιούν (μαθητές και σχολεία).

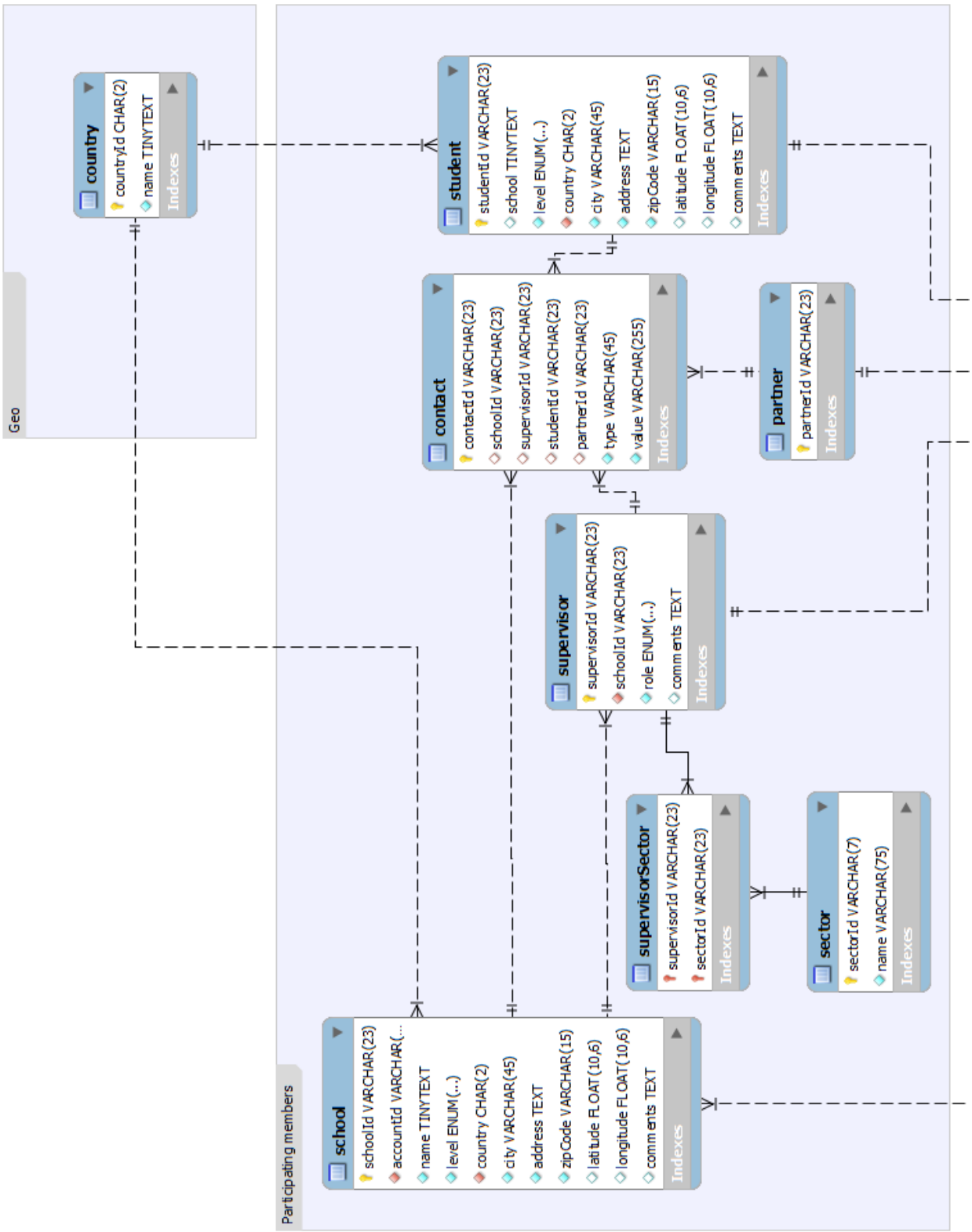
Προτιμήσεις χρηστών (preferences)

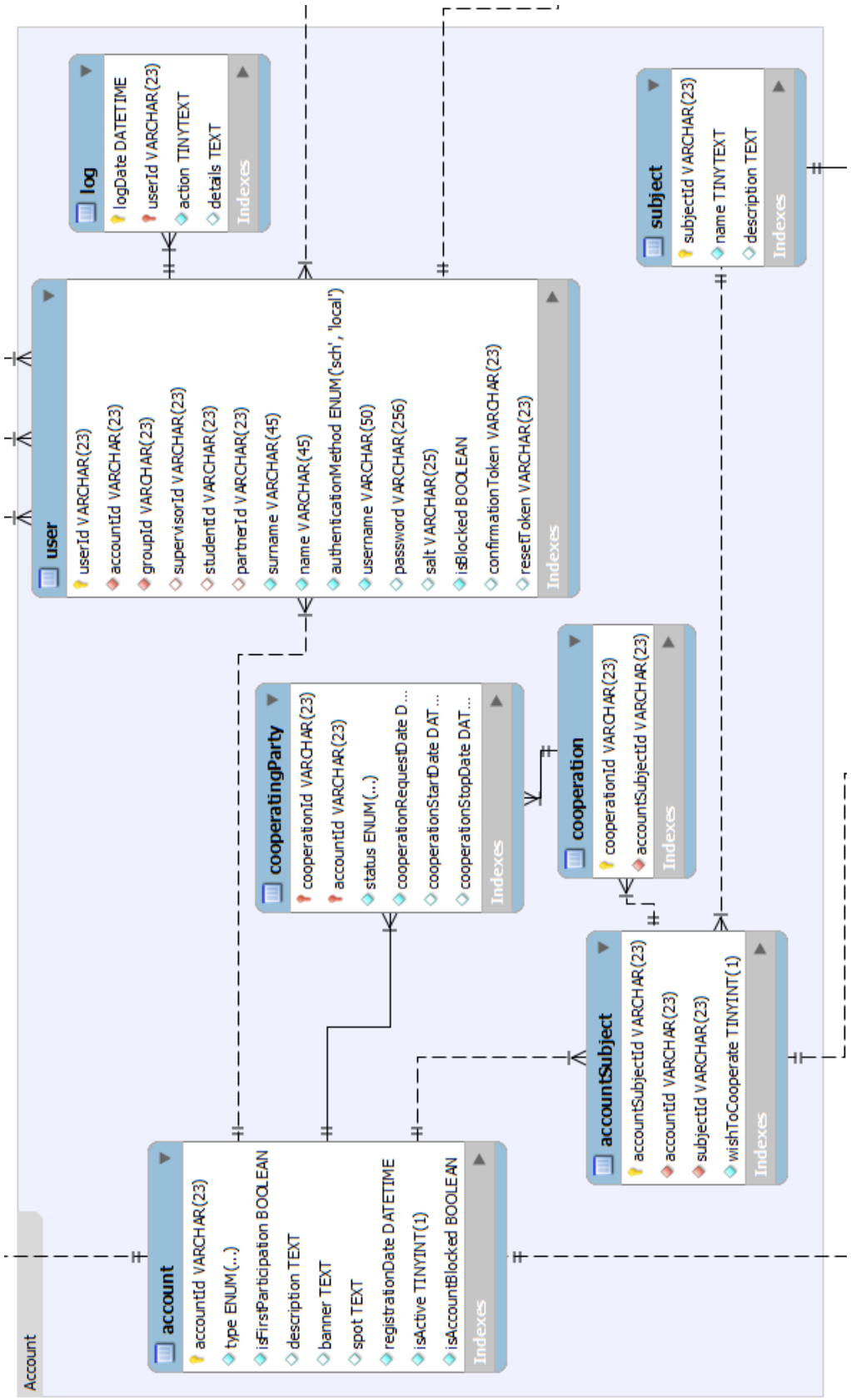
Η εφαρμογή διαχείρισης σε αυτή τη φάση, έχει ένα αρκετά υψηλό επίπεδο ευελιξίας και προσαρμοστικότητας. Ωστόσο η δυνατότητα ρύθμισης και μεταβολής διαφόρων επιλογών, περιορίζεται αποκλειστικά στην ομάδα διαχείρισης, και αφορά κατά κύριο λόγο, βασικά τμήματα της εφαρμογής.

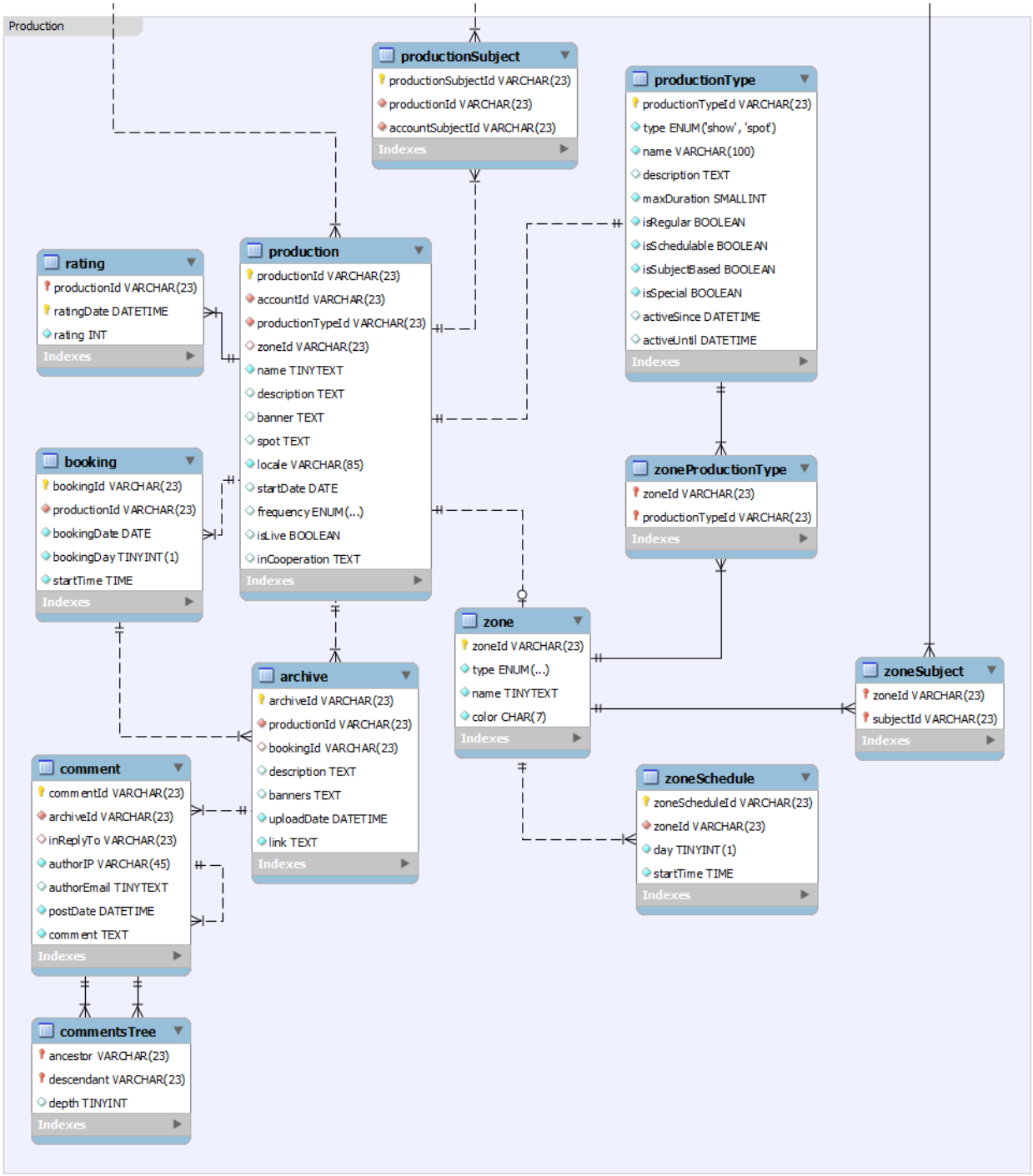
Σε επόμενο στάδιο, θα πρέπει να αναπτυχθεί η δυνατότητας ρύθμισης επιλογών και προτιμήσεων από τους χρήστες της εφαρμογής, ούτως ώστε να παραμετροποιούν άμεσα και χωρίς ιδιαίτερη προσπάθεια, τα στοιχεία και τον τρόπο προβολής των στοιχείων των λογαριασμών και των προσωπικών προφίλ τους. Θα μπορούσε, για παράδειγμα, να αναπτυχθεί δυνατότητα πλήρους προσαρμογής της εμφάνισης του προφίλ, με τις προσωπικές προτιμήσεις χρώματος φόντου ή άλλα στοιχεία μορφοποίησης (εικόνες).

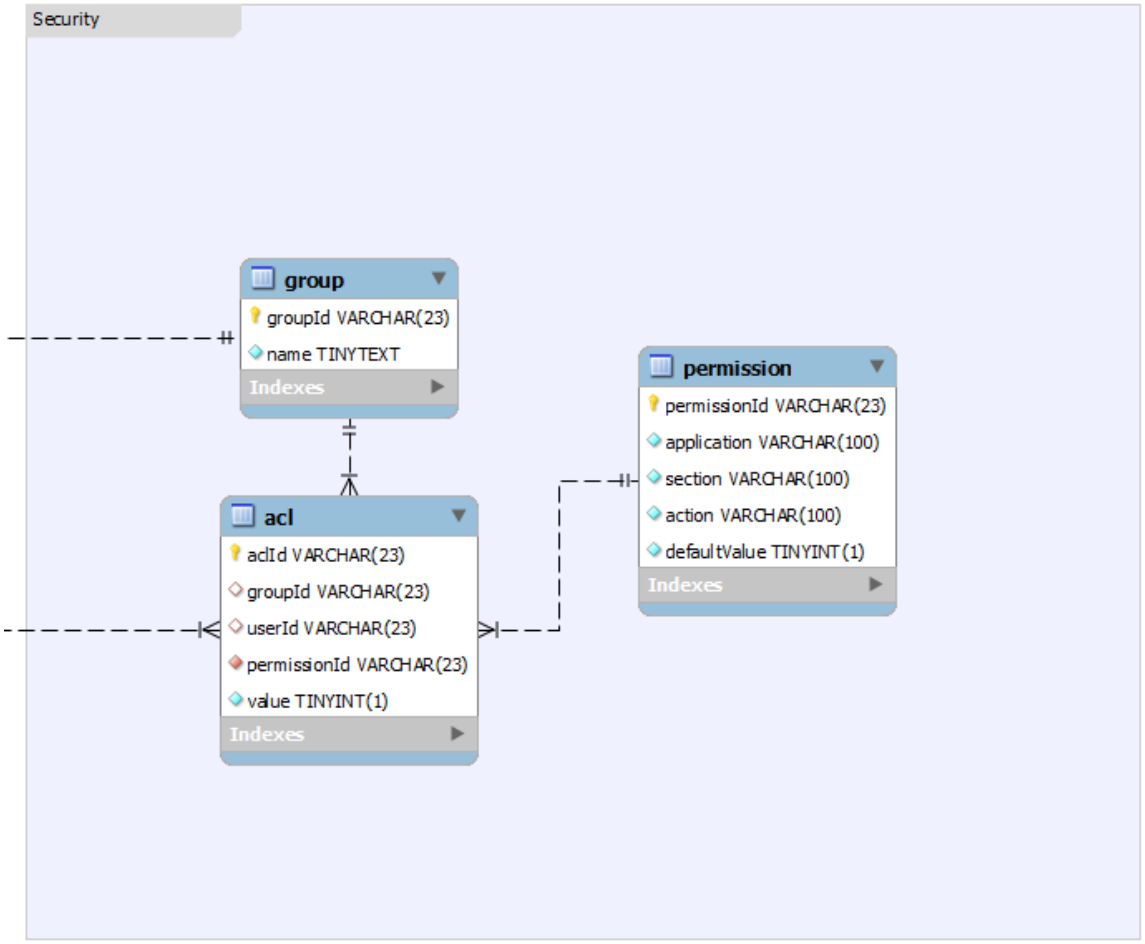
Επίσης θα ήταν δυνατή η μεταφορά, σε επίπεδο προτιμήσεων, του συνόλου των επιλογών που αφορούν ρυθμίσεις λογαριασμού, όπως για παράδειγμα οτιδήποτε αφορά μεθόδους πιστοποίησης.

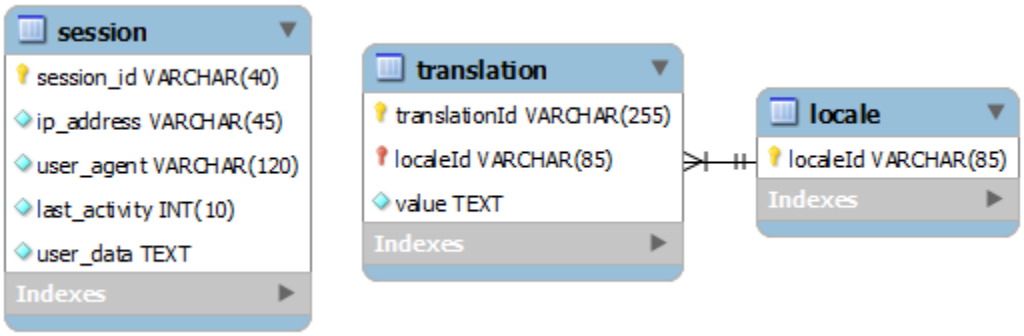
Παράρτημα Α – Δομή Βάσης Δεδομένων (Schema)




Account
log
logDate DATETIME
userId VARCHAR(23)
action TINYTEXT
details TEXT
Indexes
subject
subjectId VARCHAR(23)
name TINYTEXT
description TEXT
Indexes
user
userId VARCHAR(23)
accountId VARCHAR(23)
groupId VARCHAR(23)
supervisorId VARCHAR(23)
studentId VARCHAR(23)
partnerId VARCHAR(23)
surname VARCHAR(45)
name VARCHAR(45)
authenticationMethod ENUM('sch', 'local')
username VARCHAR(50)
password VARCHAR(256)
salt VARCHAR(25)
isBlocked BOOLEAN
confirmationToken VARCHAR(23)
resetToken VARCHAR(23)
Indexes
cooperatingParty
cooperationId VARCHAR(23)
accountId VARCHAR(23)
status ENUM(...)
cooperationRequestDate D...
cooperationStartDate DAT...
cooperationStopDate DAT...
Indexes
cooperation
cooperationId VARCHAR(23)
accountSubjectId VARCHAR(23)
Indexes
accountSubject
accountSubjectId VARCHAR(23)
accountId VARCHAR(23)
subjectId VARCHAR(23)
wishToCooperate TINYINT(1)
Indexes
account
accountId VARCHAR(23)
type ENUM(...)
isFirstParticipation BOOLEAN
description TEXT
banner TEXT
spot TEXT
registrationDate DATETIME
isActive TINYINT(1)
isAccountBlocked BOOLEAN
Indexes


Production
productionSubject
productionSubjectId VARCHAR(23)
productionId VARCHAR(23)
accountSubjectId VARCHAR(23)
Indexes
productionType
productionTypeId VARCHAR(23)
type ENUM('show', 'spot')
name VARCHAR(100)
description TEXT
maxDuration SMALLINT
isRegular BOOLEAN
isSchedulable BOOLEAN
isSubjectBased BOOLEAN
isSpecial BOOLEAN
activeSince DATETIME
activeUntil DATETIME
Indexes
rating
productionId VARCHAR(23)
ratingDate DATETIME
rating INT
Indexes
production
productionId VARCHAR(23)
accountId VARCHAR(23)
productionTypeId VARCHAR(23)
zoneId VARCHAR(23)
name TINYTEXT
description TEXT
banner TEXT
spot TEXT
locale VARCHAR(85)
startDate DATE
frequency ENUM(...)
isLive BOOLEAN
inCooperation TEXT
Indexes
booking
bookingId VARCHAR(23)
productionId VARCHAR(23)
bookingDate DATE
bookingDay TINYINT(1)
startTime TIME
Indexes
zoneProductionType
zoneId VARCHAR(23)
productionTypeId VARCHAR(23)
Indexes
zone
zoneId VARCHAR(23)
type ENUM(...)
name TINYTEXT
color CHAR(7)
Indexes
zoneSubject
zoneId VARCHAR(23)
subjectId VARCHAR(23)
Indexes
archive
archiveId VARCHAR(23)
productionId VARCHAR(23)
bookingId VARCHAR(23)
description TEXT
banners TEXT
uploadDate DATETIME
link TEXT
Indexes
comment
commentId VARCHAR(23)
archiveId VARCHAR(23)
inReplyTo VARCHAR(23)
authorIP VARCHAR(45)
authorEmail TINYTEXT
postDate DATETIME
comment TEXT
Indexes
zoneSchedule
zoneScheduleId VARCHAR(23)
zoneId VARCHAR(23)
day TINYINT(1)
startTime TIME
Indexes
commentsTree
ancestor VARCHAR(23)
descendant VARCHAR(23)
depth TINYINT
Indexes


Security
group
groupId VARCHAR(23)
name TINYTEXT
Indexes
permission
permissionId VARCHAR(23)
application VARCHAR(100)
section VARCHAR(100)
action VARCHAR(100)
defaultValue TINYINT(1)
Indexes
acl
aclId VARCHAR(23)
groupId VARCHAR(23)
userId VARCHAR(23)
permissionId VARCHAR(23)
value TINYINT(1)
Indexes


session
session_id VARCHAR(40)
ip_address VARCHAR(45)
user_agent VARCHAR(120)
last_activity INT(10)
user_data TEXT
Indexes
translation
translationId VARCHAR(255)
localeId VARCHAR(85)
value TEXT
Indexes
locale
localeId VARCHAR(85)
Indexes

Παράρτημα Β – Υπόδειγμα αρχείου ρυθμίσεων XML

```
<?xml version="1.1" encoding="utf-8"?>
<settings>
    <group name="system" icon="fa fa-desktop" isSystemGroup="true">
        <subgroup name="login" icon="fa fa-lock">
            <setting required="true">
                <id>max.allowed.login.attempts</id>
                <type>numeric</type>
                <value>3</value>
                <validation>
                    <range min="1" max="5"/>
                </validation>
            </setting>
            <setting required="true">
                <id>login.attempts.time.window</id>
                <type>numeric</type>
                <value>1</value>
                <validation>
                    <range min="1" max="10"/>
                </validation>
            </setting>
        </subgroup>
    </group>
</settings>
```

Παράρτημα Γ – Εργαλεία ανάπτυξης

Τα εργαλεία και τα πακέτα λογισμικού που χρησιμοποιήθηκαν κατά την διάρκεια ανάπτυξης αυτής της εφαρμογής ήταν τα ακόλουθα:

- **NetBeans IDE**
  Περιβάλλον ανάπτυξης κώδικα (IDE), με πολλές δυνατότητες, όπως έλεγχο εκδόσεων (version control), refactoring, auto formatting, κ.λπ.

- **MySQL Workbench**
  Εφαρμογή διαχείρισης βάσης δεδομένων και σχεδιασμού διαγράμματος οντοτήτων (ER). Παρέχει δυνατότητες συγχρονισμού (forward/backward engineering) μεταξύ του διαγράμματος οντοτήτων και της βάσης δεδομένων.

- **phpMyAdmin**
  Διαδικτυακή εφαρμογή διαχείρισης βάσης δεδομένων.

- **jQuery**
  Javascript framework ανοικτού κώδικα. Προτυποποιεί λειτουργίες javascript ούτως ώστε να έχουν κοινή χρήση και συμπεριφορά σε όλα τα προγράμματα περιήγησης (browser)

- **jQuery UI**
  Javascript framework βασισμένο στο jQuery. Προσθέτει λειτουργικότητα στην εφαρμογή, με την εισαγωγή επιπλέον εφέ κίνησης και πρόσθετων, όπως ημερολογίο, accordion lists, sliders, λειτουργία αυτόματης συμπλήρωσης (autocomplete) και άλλα.

- **Google Maps**
  Javascript API για την χρήση της εφαρμογής Google Maps, για προβολή χάρτη καθώς και λειτουργίες γεωγραφικού εντοπισμού (geolocation)

- **Apache HTTP server**
  Εξυπηρετητής (server) HTTP, ανοικτού κώδικα (open source)

- **Apache Maven**
  Εφαρμογή διαχείρισης και συντονισμού ανάπτυξης (build - deploy) πακέτων προγραμμάτων.

- **Apache Tomcat server**
  Εξυπηρετητής (server) HTTP, ανοικτού κώδικα (open source) με δυνατότητα διαχείρισης Java Servlets.

- **TeamViewer**
  Εφαρμογή απομακρυσμένου ελέγχου υπολογιστών. Χρησιμοποιείται για την παραμετροποίηση και διαχείριση του εξυπηρετητή (server) της εφαρμογής καθώς και την γενικότερη εποπτεία και έλεγχο του συστήματος.

- **Google Chrome / Mozilla Firefox / Opera**
  Προγράμματα περιήγησης (browsers), που χρησιμοποιήθηκαν για τον έλεγχο καλής λειτουργίας της εφαρμογής. Ο Internet Explorer δεν χρησιμοποιηθήκε καθώς δεν παρέχει την απαραίτητη υποστηρίξη σε βασικές λειτουργίες της εφαρμογής.