

Social Calendar – Αναφορά

Αλεβιζόπουλος Άγγελος, Βελώνης Γεώργιος, Κασνάκης Αθανάσιος, Τζιμτζίμης Ελευθέριος

1.Εισαγωγή

Η εφαρμογή που αναπτύχθηκε από την ομάδα μας αφορά ένα Διαδικτυακό ημερολόγιο με την δυνατότητα δημιουργίας συμβάντων και κοινοποίησης αυτών στην πλατφόρμα κοινωνικής δικτύωσης Facebook, προσθέτοντας έναν κοινωνικό χαρακτήρα στην εφαρμογή. Ο χρήστης μπορεί να κάνει εγγραφή στο σύστημα και να εισέλθει με τα συνθηματικά που έχει επιλέξει. Στην συνέχεια μπορεί να επιλέξει μία ημερομηνία του τρέχοντος μήνα και να δημιουργήσει συμβάντα για όλες τις ώρες της ημέρας, όπως ένα ραντεβού, μία συνάντηση, μία νυχτερινή έξοδος κ.ο.κ. Η εφαρμογή του δίνει την δυνατότητα να επιλέξει αν το συμβάν αυτό θα εμφανίζεται δημόσια σε άλλους χρήστες του ημερολογίου. Επίσης μπορεί να επιλέξει να μοιραστεί αυτό το συμβάν με χρήστες από την ομάδα φίλων του στο Facebook.

Η πιο δημοφιλής αντίστοιχη υπηρεσία είναι το Google calendar του οποίου ο κάθε χρήστης θα πρέπει να διαθέτει προσωπικό λογαριασμό Google. Ο χρήστης μπορεί να δημιουργεί συμβάντα για όλες τις ώρες της ημέρας, να επιλέγει την εμφάνιση του ημερολογίου εστιάζοντας διαδοχικά σε ημέρα, τέσσερις ημέρες, εβδομάδα, μήνα και ατζέντα. Δίνεται η δυνατότητα στον χρήστη να δημιουργήσει όσα ημερολόγια επιθυμεί ώστε να τα ταξινομεί σύμφωνα με τις ανάγκες του π.χ. γενέθλια, εργασίες, αργίες στην Ελλάδα κτλ. Ο χρήστης μπορεί να μοιράζεται τα ημερολόγια που έχει δημιουργήσει είτε δημόσια στο Διαδίκτυο, είτε με συγκεκριμένα άτομα μέσω ηλεκτρονικού ταχυδρομείου, ακόμα και με συναδέλφους στην εταιρεία που εργάζεται. Το ημερολόγιο Google δίνει την δυνατότητα στον χρήστη να δημοσιεύσει το συμβάν που έχει δημιουργήσει σε έναν ιστότοπο παράγοντας αυτόματα τον κώδικα html, ώστε να τοποθετηθεί σε έναν ιστότοπο και οι επισκέπτες να μπορούν να προσθέσουν αυτό το συμβάν στο δικό τους ημερολόγιο.

2. CodeIgniter framework

Για την δημιουργία του project μας κάναμε χρήση του προγραμματιστικού πλαισίου CodeIgniter. Είναι ένα framework ανοιχτού κώδικα για γρήγορη ανάπτυξη και κατασκευή δυναμικών web εφαρμογών σε γλώσσα PHP. Στην προσπάθεια μας να διατηρήσουμε τον κώδικα μας απλό και ξεκάθαρο δίνοντας την δυνατότητα για μελλοντική επέκταση της εφαρμογής μας, χρησιμοποιήσαμε τις έτοιμες βιβλιοθήκες του framework. Οι βιβλιοθήκες δίνουν την δυνατότητα στον προγραμματιστή να συντάξει πολύπλοκο κώδικα παρέχοντας του ορισμένες χρήσιμες συναρτήσεις για την εκτέλεση περίπλοκων αναζητήσεων και λειτουργικότητας. Χωρίς την χρήση ενός framework σαν το CodeIgniter ο καθορισμός των βιβλιοθηκών θα έπρεπε να γίνει από τον προγραμματιστή.

Ο τρόπος λειτουργίας του CodeIgniter βασίζεται στην τμηματική προσέγγιση μέσω του δημοφιλούς προτύπου σχεδιασμού λογισμικού Model-view-controller (MVC). Το πρότυπο αυτό χρησιμοποιείται κοινός για την υλοποίηση γραφικών διεπαφών χρήστη (GUIs) διαιρώντας την εφαρμογή λογισμικού σε τρία αλληλοσυνδεδεμένα μέρη (components):

- Το model, το οποίο διαχειρίζεται απευθείας τα δεδομένα που σχετίζονται με την βάση δεδομένων, την λογική, και τους κανόνες της εφαρμογής.
- Ο controller, δέχεται εισόδους και τις μετατρέπει σε εντολές για το model και ενημερώνει την κατάσταση του. Επίσης στέλνει εντολές στο view ώστε να γίνει η αντίστοιχη αναπαράσταση των δεδομένων του model μέσω του view.

- Το view, προβολή μπορεί να είναι οποιαδήποτε αναπαράσταση εξόδου των πληροφοριών, όπως ο κώδικας html, το frond-end της ιστοσελίδας.

Κάθε προγραμματιστής διαθέτει προσωπικό στυλ γραφής και δίνει διαφορετικές προτεραιότητες στην επιλογή των εργαλείων που θα χρησιμοποιήσει στην ανάπτυξη Διαδικτυακών εφαρμογών. Οι λόγοι επιλογής του CodeIgniter για την δημιουργία της εφαρμογής μας είναι οι εξής:

- Εύκολος τρόπος εγκατάστασης και ρύθμισης
- Κομψός τρόπος υλοποίησης του MVC
- Καλή διαχείριση του κώδικα και αντιμετώπιση των λαθών
- Βελτιωμένη συνολική απόδοση
- Χρήση του μοτίβου Active Record για πρόσβαση στην βάση δεδομένων
- Πλούσιες βιβλιοθήκες που παρέχουν όλα τα απαραίτητα εργαλεία με εξαιρετική τεκμηρίωση
- Δυνατότητα μελλοντικής επέκτασης της εφαρμογής
- Χωρίς σύγκρουση σε διαφορετικές εκδόσεις της PHP

3. Social Calendar

Η δικτυακή εφαρμογή που αναπτύξαμε εκτελείται από την πλευρά του διακομιστή εφαρμογών (server-side application) και εμφανίζει το περιεχόμενο της δυναμικά, ανάλογα με το τι ζητάει ο κάθε επισκέπτης, ενώ ταυτόχρονα ανταποκρίνεται σε οποιαδήποτε διάταξη μίας οθόνης προβολής. Η χρήση μίας βάσης δεδομένων μας επιτρέπει την εύκολη προσθαφαίρεση περιεχομένου και η διαχείριση του γίνεται από έναν εύχρηστο μηχανισμό Content Management System (CMS), το framework CodeIgniter που αναφέραμε νωρίτερα. Η λειτουργία των κλάσεων και των συναρτήσεων του CodeIgniter είναι να αναζητούν περιεχόμενο στον εκάστοτε πίνακα της βάσης δεδομένων και να προβάλουν το κάθε αντικείμενο ξεχωριστά στην ιστοσελίδα. Για τον λόγο αυτό έχουμε δημιουργήσει τον Controller, το Model, το View, όπως επίσης και την βάση δεδομένων στην οποία θα αποθηκεύεται η πληροφορία.

Ένα χαρακτηριστικό της εφαρμογής μας είναι ότι δίνει την δυνατότητα στους επισκέπτες να πραγματοποιήσουν εγγραφή ως μέλη της ιστοσελίδας. Ο επισκέπτης μπορεί να κάνει χρήση του συστήματος εγγραφής, να καταχωρίσει τα στοιχεία του, τον επιθυμητό κωδικό πρόσβασης και να εισέλθει στην ιστοσελίδα ως μέλος. Το σύστημα εγγραφής (registration module) επεξεργάζεται τα στοιχεία ως εξής:

1. Ζητάει από τον επισκέπτη να συμπληρώσει τα βασικά στοιχεία και να καταχωρίσει την διεύθυνση email που θα χρησιμοποιεί ως όνομα χρήστη
2. Αφού ο επισκέπτης υποβάλλει τα στοιχεία της φόρμας τα δεδομένα καταχωρούνται στην βάση δεδομένων
3. Στην συνέχεια ο χρήστης μπορεί να εισέλθει στην ιστοσελίδα ως μέλος με αναβαθμισμένα δικαιώματα

Η μονάδας εγγραφής χρήστη διαχειρίζεται από τα απαραίτητα αρχεία του CodeIgniter. Το αρχείο user_model.php χειρίζεται το τμήμα **Model** της εφαρμογής, κάνοντας ανάκτηση των πληροφοριών email και password που εισάγει ο χρήστης με την μέθοδο GET. Ανακτά το αναγνωριστικό (id) χρήστη επίσης με την μέθοδο GET και στην συνέχεια εισάγει τις πληροφορίες στην βάση δεδομένων, στον ήδη δημιουργημένο πίνακα user κάνοντας χρήση των παρακάτω συναρτήσεων:

1. `get_user()` εκτελεί την αναζήτηση και επιστρέφει το αποτέλεσμα
2. `insert()` εγγραφή των δεδομένων στον πίνακα `user`
3. `get_user_id_from_username()` επιστρέφει το αναγνωριστικό χρήστη

Ο **Controller** καθορίζει τον τρόπο που αντιμετωπίζονται τα αιτήματα HTTP. Είναι απλά ένα αρχείο που περιέχει μία κλάση η οποία ονομάζεται με τέτοιο τρόπο ώστε να μπορεί να συσχετιστεί με τον Uniform Resource Identifier (URI) της εφαρμογής. Η συνάρτηση `index()` δηλώνεται εντός του αρχείου `Event.php` το οποίο, είναι ένας ακόμη controller της εφαρμογής μας που διαχειρίζεται την δημιουργία συμβάντων στο ημερολόγιο. Η συνάρτηση αυτή δεν επιστρέφει κανένα αποτέλεσμα, είναι όμως η προεπιλεγμένη συνάρτηση που φορτώνεται όταν το δεύτερο τμήμα σε ένα URI είναι κενό.

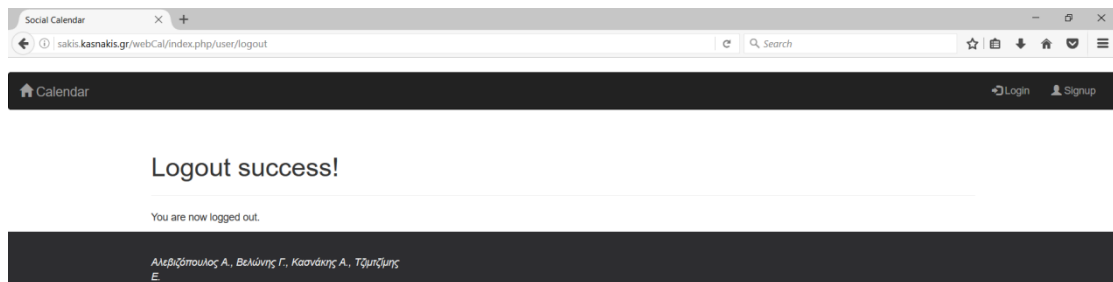
Η μέθοδος `construct()` αρχικοποιεί τους URL Helpers οι οποίοι βοηθούν στην δημιουργία συνδέσμων, τις βιβλιοθήκες για την δημιουργία session και την επικοινωνία με το Model. Στην συνάρτηση `add_event()` αρχικά θέτουμε τους κανόνες επικύρωσης και στην συνέχεια με την εντολή `if ($this->form_validation->run() == FALSE)` ελέγχουμε αν πληρούνται οι κανόνες δημιουργίας συμβάντων στο ημερολόγιο, διαφορετικά εμφανίζουμε ένα μήνυμα σφάλματος.

Το **View** περιέχει τον κώδικα διεπαφής χρήστη (user interface) όπως την φόρμα εγγραφής χρήστη με όλα τα απαιτούμενα πεδία, την κεντρική σελίδα κ.α. Για την εμφάνιση της ιστοσελίδας έχουμε χρησιμοποιήσει διάφορα στυλ Cascading Style Sheets (CSS). Για την σχεδίαση κάναμε χρήση του Bootstrap framework το οποίο μας επιτρέπει την ανάπτυξη του front-end της εφαρμογής μας με τέτοιο τρόπο ώστε να ανταποκρίνεται σε οποιοδήποτε μέγεθος οθόνης προβάλλεται.

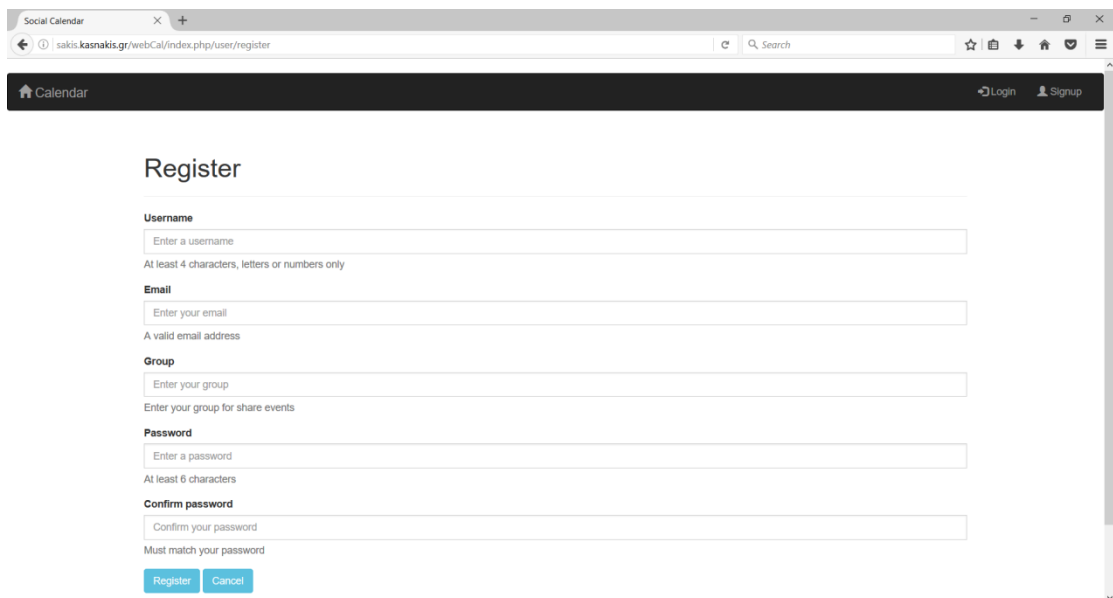
Κατά την σύνδεση του χρήστη στο σύστημα δημιουργείτε μία μεταβλητή τύπου session στην οποία αποθηκεύονται προσωρινά δεδομένα, που περιέχουν ορισμένα στοιχεία για την αναγνώριση του συγκεκριμένου χρήστη σε όλες τις σελίδες της εφαρμογής. Για την αποσύνδεση του χρήστη θα χρειαστεί να απομακρύνουμε τα δεδομένα το τρέχοντος session με την συνάρτηση `unset()` και ο χρήστης θα ανακατευθυνθεί στην αρχική σελίδα μέσω της συνάρτησης `redirect()`.

Ο χρήστης μπορεί να αλλάξει το password και το group στο οποίο ανήκει με επιβεβαίωση παλιού και νέου password. Το σύστημα κάνει χρήση του User controller ο οποίος με την σειρά του καλεί την συνάρτηση `user_update()`. Στην συνέχεια το Model με την συνάρτηση `user_update_password()` ενημερώνει την βάση δεδομένων.

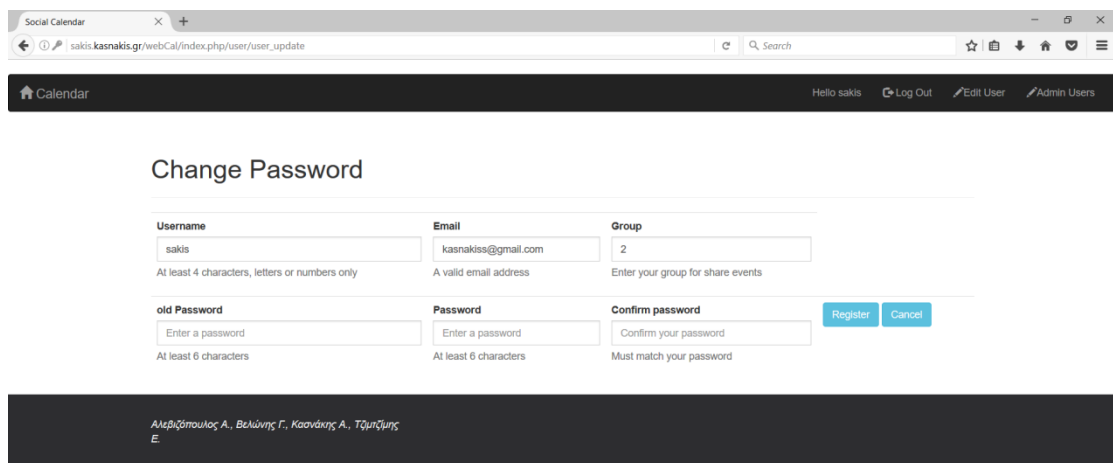
Εικόνα 1. Η φόρμα εισαγωγής συνθηματικών χρήστη



Εικόνα 2. Το μήνυμα επιτυχής αποσύνδεσης



Εικόνα 3. Η φόρμα καταχώρησης στοιχείων χρήστη



Εικόνα 4. Η φόρμα αλλαγής κωδικού πρόσβασης

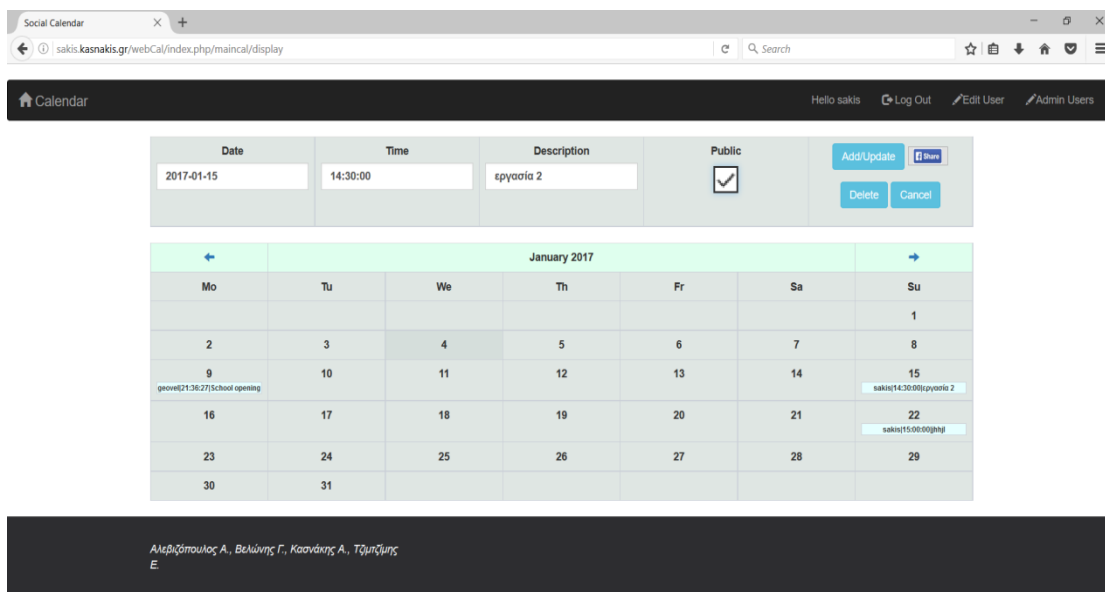
3.1 Κυρίως εφαρμογή

Τα επίπεδα πρόσβασης είναι δύο. Απλός χρήστης ή administrator. Ανάλογα με το επίπεδο πρόσβασης αλλάζουν και οι διαθέσιμες επιλογές στο μενού. Το επίπεδο πρόσβασης ελέγχεται από το περιεχόμενο της μεταβλητής συνόδου `$_SESSION('is_Admin')`.

Αν ο χρήστης αποκτήσει δικαιώματα απλού χρήστη τότε γίνεται κλήση του βασικού controller της εφαρμογής `Maincal` και πιο συγκεκριμένα της μεθόδου `Maincal/display()`, η

οποία φορτώνει τη βασική φόρμα maincal και στη συνέχεια αναλαμβάνει με κλήση κατάλληλης μεθόδου του model Maincal_model:

- να παρουσιάσει σε μορφή ημερολογίου το περιεχόμενο χρήστη,
- να εισάγει ή να ενημερώσει εγγραφές συμβάντων και να ενημερώσει τους εμπλεκόμενους χρήστες με email,
- να διαγράψει εγγραφές συμβάντων
- να κάνει login σε προφίλ του χρήστη στο Facebook και στη συνέχεια Post στο τρέχων συμβάν.



Εικόνα 5. Η παρουσίαση του ημερολογίου

Η εμφάνιση των συμβάντων και η δημοσίευσή τους σε κοινωνικά δίκτυα ή μέσω email ελέγχεται σε δύο επίπεδα.

- Ο χρήστης μπορεί να βλέπει τα συμβάντα που
 - ο έχει δημιουργήσει ο ίδιος
 - είναι δημόσια(public) από άλλους χρήστες που ανήκουν στην ίδια ομάδα(group).
- Η αποστολή για ενημέρωση με e-mail ακολουθεί την ίδια λογική όσο αναφορά τα δικαιώματα, αλλά πραγματοποιείται μόνο κατά την δημιουργία ενός νέου συμβάντος.
 - ο στέλνει email στον κάτοχο του συμβάντος
 - Αν είναι public στέλνει και στους χρήστες της ίδιας ομάδας.
- Για το Facebook σαν παράδειγμα κοινωνικού δικτύου ο χρήστης μπορεί να δημοσιεύσει με επιλογή όποιο συμβάν είναι διαθέσιμο σ' αυτόν.

3.2 Backend διαχείρισης

3.2.1 Επίπεδο πρόσβασης Administrator.

Στην περίπτωση που ο χρήστης αποκτήσει δικαιώματα administrator, στο μενού εμφανίζεται επιπλέον επιλογή.

Έγινε υλοποίηση της διαχείρισης χρηστών σε επίπεδο update. Για λόγους κυρίως ασφάλειας κατά τη διάρκεια των δοκιμών αφαιρέθηκε η δυνατότητα αλλαγής του Password για όλους τους χρήστες από τον administrator.

Ο διαχειριστής έχει την επιπλέον δυνατότητα

- να κάνει αναζήτηση χρηστών με βάση το username
- να κάνει αλλαγές στο username, email, user group, isConfirmed, isAdmin. Ουσιαστικά κυρίως να αλλάξει το Group χρηστών, το αν έχει γίνει επιβεβαίωση(μελλοντική χρήση), και το αν είναι διαχειριστής ή όχι.

The screenshot shows the 'Social Calendar' backend interface. At the top, there's a navigation bar with 'Hello sakis', 'Log Out', 'Edit User', and 'Admin Users'. Below this is a search form with fields for 'id', 'Username', 'Email', 'Group', 'Is Confirmed', and 'Is Admin'. A table below the search form lists users with columns: id, Username, email, group id, is Confirmed, and is Admin. The table contains 12 rows of user data.

id	Username	email	group id	is Confirmed	is Admin
1	george	gkasnak@sch.gr	2	1	1
2	nikos	nikos@gmail.com	1	0	0
3	sakis	kasnakiss@gmail.com	2	1	1
4	Angel55	angelonpc@gmail.com	2	1	1
6	geovel	geovel@freemail.gr	2	1	1
7	angelonpc	aggelos@freemail.gr	1	0	0
8	Extasia93	leytzm93@hotmail.com	1	0	0
9	aristantasios	aristantasios@gmail.com	0	0	0
10	karakousis	sakis.karakousis@gmail.com	0	0	0
11	gianniskarakous	gian.karakousis@gmail.com	0	0	0
12	trikalopoulos	geotrik520@gmail.com	0	0	0
13	hivickone	hivickone@freemail.ro	0	0	0

Εικόνα 6. Το backend διαχείρισης

3.2.2 Το Facebook API

Έχει δημιουργηθεί λογαριασμός στο Facebook για να πάρουμε API key και στη συνέχεια χρησιμοποιήθηκε JQuery για την επικοινωνία (λήψη script για login) στο Facebook και κατασκευάστηκε επίσης σε JQuery το κομμάτι που η εφαρμογή κάνει post ένα συμβάν.

The screenshot shows the 'Social Calendar' frontend interface. At the top, there's a navigation bar with 'Hello george', 'Log Out', 'Edit User', and 'Admin Users'. Below this is a form to add or update an event with fields for 'Date', 'Time', 'Description', and 'Public'. A calendar for January 2017 is displayed below the form. A Facebook login overlay is visible on the right side of the screen, prompting the user to log in to use their Facebook account with webCal.

Facebook - Mozilla Firefox
https://www.facebook.com/login.php?skip_api_login=1&api...

Log in to use your Facebook account with webCal

Email or Phone:

Password:

Log In

Forgot account?

Create New Account

Εικόνα 7. Το Facebook API

4. Application Programming Interface (API) implementation

Για επικοινωνία με άλλες εφαρμογές – κατασκευή API το codeigniter υποστηρίζει την τεχνολογία XMLRPC. Από τη μεριά της εφαρμογής μας μπορεί να λειτουργεί ένας XMLRPC server ενώ από τη μεριά του πελάτη ένας XMLRPC client. Ο πελάτης παράδειγμα θα μπορούσε να είναι ένα ANDROID κινητό.

Η εφαρμογή XMLRPC client στέλνει μορφοποιημένα σε XML τα δεδομένα και τη μέθοδο που θέλουμε να εκτελεστεί. Ο XMLRPC server εκτελεί τη μέθοδο με τα δεδομένα σαν είσοδο και τα αποτελέσματα τα επιστρέφει σε μορφή XML σαν request στον client.

Θα μπορούσε να υλοποιηθεί σε επόμενη φάση.

Η εφαρμογή μας σε οποιαδήποτε στιγμή μπορεί να επεκταθεί όσον αφορά την επικοινωνία με άλλες εφαρμογές χωρίς αλλαγές στον πυρήνα της.

5. Αξιολόγηση με την χρήση web usability test

Η αναφορά των αποτελεσμάτων του πραγματοποιηθέντος usability testing παρατίθεται σε ξεχωριστό αρχείο με όνομα: Αναφορά αποτελεσμάτων Testing.

6. Διαδικτυακές πηγές και παραδείγματα

CodeIgniter

1. https://www.codeigniter.com/user_guide/
2. https://codeigniter.com/user_guide/libraries/calendar.html
3. https://www.codeigniter.com/userguide2/database/active_record.html
4. <http://easyappointments.org/>
5. <https://code.tutsplus.com/tutorials/codeigniter-from-scratch-the-calendar-library--net-9570>
6. <https://www.eduonix.com/blog/web-programming-tutorials/calendar-codeigniter/>
7. <http://www.w3programmers.com/build-a-full-featured-login-and-registration-system-with-codeigniter/>
8. <http://www.kodingmadesimple.com/2016/06/codeigniter-login-and-registration-tutorial-source-code.html>
9. <https://www.formget.com/codeigniter-calendar/>
10. <https://github.com/hedii/Codeigniter-login-logout-register>

HTML/CSS/Javascript

1. <http://www.w3schools.com/html/default.asp>
2. <http://www.w3schools.com/cssdefault.asp>
3. <http://www.w3schools.com/js/default.asp>
4. <https://www.acyba.com/el/acymailing/248-acymailing-external-subscription-form.html>
5. <http://www.wlearn.gr/>
6. <http://eclass.teiion.gr/modules/document/file.php/EPDO133/HTML%20CE%B5%CE%BA%CE%BC%CE%AC%CE%B8%CE%B7%CF%83%CE%B7/html/les9/input.html>
7. Γιώργος Λιακέας - Η γλώσσα Javascript

Bootstrap

1. <http://getbootstrap.com/css/>
2. <https://v4-alpha.getbootstrap.com/layout/grid/>
3. <https://v4-alpha.getbootstrap.com/components/navbar/>
4. <https://getbootstrap.com/examples/grid/>
5. <http://getbootstrap.com/2.3.2/scaffolding.html>
6. <http://www.w3schools.com/bootstrap/default.asp>
7. <http://engineering.como.com/using-bootstrap-for-responsive-sites/>
8. <http://bootsnipp.com/tags/registration>
9. <http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-forms.php>

Usability testing

1. <https://www.usability.gov/how-to-and-tools/methods/reporting-usability-test-results.html>
2. <http://www.trymyui.com/usability-testing-benefits>
3. <http://www.blastam.com/benefits-of-usability-testing>
4. <https://econsultancy.com/blog/5932-ten-free-usability-testing-tools/>
5. <http://www.usefulusability.com/24-usability-testing-tools/>
6. <http://www.jenniferslegg.com/2007/09/24/list-of-questions-to-ask-website-usability-testers/>
7. <http://www.legaltechdesign.com/LegalDesignToolbox/2015/04/29/userfocus-web-usability-checklist-guide/>
8. <http://jrjw.com/cadbury-website-usability-testing/>