**Project 2020**

**Μηχανικη Λογισμικου για Διαδικτυακες Εφαρμογες**



**Δημιουργια διαδικτυακης εφαρμογης με χρηση δεδομενων απο εξωτερικες πηγες**

***ΚΑΤΑΓΡΑΦΗ ΑΠΑΙΤΗΣΕΩΝ***

Ομάδα Υλοποίησης του Project

Ηλιάδης Αλέξιος, Κόκκινου Ξανθή, Μήσιος Δημήτριος, Ταουκτσής Βασίλης

Περιεχόμενα

[1. Τι υλοποιεί η εφαρμογή; 2](#_Toc57584023)

[2. Πως θα υλοποιηθεί 2](#_Toc57584024)

[3. Ανάλυση απαιτήσεων 3](#_Toc57584025)

[4. Τι κάνει η εφαρμογή σε επίπεδο χρηστών 4](#_Toc57584026)

[Κλήσεις προς το API SongKick 4](#_Toc57584027)

[5. Σχεδιάγραμμα της αρχιτεκτονικής του συστήματος 7](#_Toc57584028)

# Τι υλοποιεί η εφαρμογή;

Το ***musiCity*** είναι μια διαδικτυακή εφαρμογή η οποία θα υλοποιηθεί στα πλαίσια της ομαδικής εργασίας στο μάθημα «Μηχανική Λογισμικού για Διαδικτυακές Εφαρμογές, του μεταπτυχιακού προγράμματος «Ευφυείς Τεχνολογίες Διαδικτύου» του Τμήματος Μηχανικών Πληροφορικής του Διεθνούς Πανεπιστημίου Ελλάδας.

Μέσω της εφαρμογής αυτής θα παρέχουμε τη δυνατότητα στον χρήστη να πληροφορείται για μουσικά γεγονότα (event) που θα γίνονται σε μια πόλη. Η πόλη θα είναι προκαθορισμένη από τον προγραμματιστή και θα είναι η «Θεσσαλονίκη». Στη συνέχεια θα δίνεται η δυνατότητα στο χρήστη να επιλέξει μια άλλη πόλη. Ο χρήστης θα μπορεί να δημιουργεί λογαριασμό στην υπηρεσία μας. Αν έχει ξεχάσει τον κωδικό του θα υπάρχει λειτουργία επανέκδοσής του. Ο συνδεμένος χρήστης θα μπορεί να επιλέγει μουσικά event που τον ενδιαφέρουν. Το σύστημα θα τον πληροφορεί ποιες εκδηλώσεις έχει χαρακτηρίσει ότι τον ενδιαφέρουν.

Η πορεία εξέλιξης του project φαίνεται στο περιβάλλον του GitHub στον παρακάτω σύνδεσμο: <https://github.com/iee-ihu-gr-coursem102/WebEng20_group9_CSearch>

Η ιστοσελίδα θα εμφανιστεί στην συνέχεια στο παρακάτω URL: <https://nireas.it.teithe.gr/webeng20g9>

# Πως θα υλοποιηθεί

* Για το Backend θα χρησιμοποιηθεί η **PHP** η οποία θα αναλάβει την διαχείριση των αιτημάτων (κλήσεων).
* Θα γίνει χρήση της **MySQL** βάσης δεδομένων στην οποία θα αποθηκεύονται τα στοιχεία των χρηστών και το ενδιαφέρον τους για μια εκδήλωση.
* Θα γίνει χρήση του **Api Songkick** απ’ όπου θα αντλούνται οι πληροφορίες για τα μουσικά γεγονότα που γίνονται σε μια πόλη.
* Για το Frontend Θα χρησιμοποιηθεί η **bootstrap** βιβλιοθήκη και ο χειρισμός των γεγονότων θα γίνει με **jQuery**.
* Οι κλήσεις από client 🡺server θα γίνουν με τεχνολογία **ΑJΑΧ**.

# Ανάλυση απαιτήσεων

Η ανάλυση των απαιτήσεων είναι η δραστηριότητα με την οποία συλλέγει η ομάδα ανάπτυξης, οποιαδήποτε διαθέσιμη πηγή για τη λειτουργικότητα που πρέπει να παρέχει το σύστημα στους μελλοντικούς χρήστες. Η διαδικασία διερεύνησης απαιτήσεων μπορεί να είναι περίπλοκη.

Για να σχεδιαστεί η λειτουργικότητα της εφαρμογής μας, επιλέξαμε να κάνουμε χρήση της τεχνικής JAD (Joint Application Development). Είναι μια ομαδική τεχνική που απαιτεί τη συμμετοχή όλων των ενδιαφερομένων ενός έργου, δηλ. αναλυτές, σχεδιαστές, χρήστες, διαχειριστές συστημάτων και πελάτες.

Καταγράφηκαν ένα σύνολο απαιτήσεων σε ένα σύνολο συνεδριών αρκετών ημερών. Σε κάθε συνεδρία, οι απαιτήσεις αναλύθηκαν και δημιουργήθηκαν το πεδίο του προβλήματος και η τεκμηρίωση του. Κατά τη διάρκεια κάθε συνεδρίας η ομάδα συζήτησε για τα διάφορα θέματα, καταλήγοντας σε ένα σύνολο τεκμηριωμένων συμπερασμάτων. Τα συμπεράσματα της ομάδας οδήγησαν στις προδιαγραφές του συστήματος.

# Τι κάνει η εφαρμογή σε επίπεδο χρηστών

### Κλήσεις προς το API SongKick

**Metro area’s upcoming events (calendar)**

Εδώ βρίσκουμε επερχόμενες εκδηλώσεις για μια περιοχή (metro). Η περιοχή του metro είναι μια πόλη ή μια συλλογή πόλεων που χρησιμοποιεί το Songkick για να ειδοποιεί τους χρήστες για συναυλίες κοντά τους. Χρησιμοποιούμε την αναζήτηση τοποθεσίας μας για να βρούμε ένα αναγνωριστικό περιοχής metro.

**Request**

[https://api.songkick.com/api/3.0/metro\_areas/{metro\_area\_id}/calendar.json?apikey={your\_api\_key}](https://api.songkick.com/api/3.0/metro_areas/%7bmetro_area_id%7d/calendar.json?apikey=%7byour_api_key%7d)

Μας επιστρέφει :

{

"resultsPage": {

"results": {

"event": [

{

"id":11129128,

"type":"Concert",

"uri":"http://www.songkick.com/concerts/11129128-wild-flag-at-fillmore?utm\_source=PARTNER\_ID&utm\_medium=partner",

"displayName":"Wild Flag at The Fillmore (April 18, 2012)",

"start": {

"time":"20:00:00",

"date":"2012-04-18",

"datetime":"2012-04-18T20:00:00-0800"

},

"performance": [

{

"artist": {

"id":29835,

"uri":"http://www.songkick.com/artists/29835-wild-flag?utm\_source=PARTNER\_ID&utm\_medium=partner",

"displayName":"Wild Flag",

"identifier": []

},

"id":21579303,

"displayName":"Wild Flag",

"billingIndex":1,

"billing":"headline"

}

],

"location": {

"city":"San Francisco, CA, US",

"lng":-122.4332937,

"lat":37.7842398

},

"venue": {

"id":6239,

"displayName":"The Fillmore",

"uri":"http://www.songkick.com/venues/6239-fillmore?utm\_source=PARTNER\_ID&utm\_medium=partner",

"lng":-122.4332937,

"lat":37.7842398,

"metroArea": {

"id":26330,

"uri":"http://www.songkick.com/metro-areas/26330-us-sf-bay-area?utm\_source=PARTNER\_ID&utm\_medium=partner",

"displayName":"SF Bay Area",

"country": { "displayName":"US" },

"state": { "displayName":"CA" }

}

},

"status":"ok",

"popularity":0.012763

}, ....

]

},

"totalEntries":24,

"perPage":50,

"page":1,

"status":"ok" }

}

Για να βρούμε τα στοιχεία μιας πόλης κάνουμε μια κλήση:

<https://api.songkick.com/api/3.0/search/locations.json?query=thessaloniki&apikey=RmJpmyc8RlY74n1I>

Στην παρούσα φάση δεν κάνουμε αναζήτηση του ID της πόλης, το έχουμε βρει εκ των προτέρων με κλήση:

<https://api.songkick.com/api/3.0/metro_areas/ΚΩΔΙΚΟΣ.ΠΟΛΗΣ/calendar.json?apikey=APIKEY&page=1&per_page=20>

Η εφαρμογή μας επιστρέφει αποτελέσματα για μια συγκεκριμένη πόλη οπότε και ο κωδικός της πόλης είναι εκ των προτέρων γνωστός. Σκοπός αυτής της προσέγγισης είναι να ελαχιστοποιήσουμε τις κλήσεις.

Εάν επεκταθεί η εφαρμογή στην επιλογή μιας πόλης τότε θα πρέπει να αποφασιστεί εάν ο κωδικός της πόλης θα αποκτάται δυναμικά μέσω κλήσεων στο API ή αν ο χρήστης θα έχει τη δυνατότητα να επιλέξει μέσα από έναν προκαθορισμένο κατάλογο πόλεων. Ο product owner δεν έχει αποφασίσει ακόμη πως θέλει σε αυτό το σημείο να λειτουργεί η εφαρμογή.

**Παράμετροι κλήσεων:**

apikey Το κλειδί API

metro\_area\_id Το αναγνωριστικό της περιοχής του μετρό για την επιστροφή συμβάντων

min\_date Μια ημερομηνία με τη μορφή ΕΕΕΕ-ΜΜ-ΗΗ

max\_date Μια ημερομηνία με τη μορφή ΕΕΕΕ-ΜΜ-ΗΗ

page Τα αποτελέσματα είναι σελιδοποιημένα. Αυτό καθορίζει τον αριθμό σελίδας αποτελεσμάτων. (Πρώτη σελίδα = 1)

per\_page Ο αριθμός των αποτελεσμάτων που θα επιστραφούν σε κάθε σελίδα. (Μέγιστο 50)

Όταν ο χρήστης κάνει μια αναζήτηση δεν επιστρέφουν σαν αποτέλεσμα όλες τις πληροφορίες που μπορεί να στείλει το API. Φιλτράρεται το αποτέλεσμα και στον χρήστη εμφανίζονται:

* Όνομα event με το αντίστοιχο Link
* Όνομα καλλιτέχνη με το αντίστοιχο Link
* Το όνομα του χώρου διεξαγωγής με το αντίστοιχο Link
* Ημερομηνία
* Ώρα
* Τι είδους εκδήλωση είναι (concert/festival)
* Κατάσταση (Αν θα πραγματοποιηθεί ή όχι)

Επίσης επιστρέφουν οι παρακάτω πληροφορίες που δεν είναι άμεσα ορατές στον χρήστη αλλά χρησιμοποιούνται από την εφαρμογή

* Κωδικός event
* Πληροφορία αν ο χρήστης έχει εκδηλώσει ενδιαφέρον για το event ( 1 ναι 0 όχι).

# Σχεδιάγραμμα της αρχιτεκτονικής του συστήματος

Χρήστης μη συνδεδεμένος

**client**

**server**

**SongKick API**

***External DB SongKick***

Son

S

S

Χρήστης συνδεδεμένος

Προτιμήσεις χρήστη

***DB musiCity***

Son

S

S

client

Server

SongKick API

***External DB SongKick***

Son

S

S

Ερώτημα για προτιμήσεις χρήστη