



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη πλατφόρμας συνεργατικότητας πολιτών
Title	An e-petition platform for assisting the management of local communities
Όνοματεπώνυμο Φοιτητή	Γεώργιος Περρέας
Πατρώνυμο	Δημήτριος
Αριθμός Μητρώου	ΜΠΣΠ/ 14074
Επιβλέπων	Δημήτρης Αποστόλου, Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης **Ιούνιος 2016**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Κωνσταντίνος Πατσάκης
Επίκουρος Καθηγητής

Κωνσταντίνος Μεταξιώτης
Αναπληρωτής Καθηγητής

Δημήτρης Αποστόλου
Αναπληρωτής Καθηγητής

Περίληψη

Με την εξέλιξη των Τεχνολογιών Πληροφορικής και Επικοινωνίας, αναπτύχθηκε παγκοσμίως μια τάση στους Δήμους και τις τοπικές κοινωνίες να χρησιμοποιήσουν τις δυνατότητες του Διαδικτύου για την αποτελεσματικότερη διαχείριση της διακυβέρνησης των πόλεων. Στην παρούσα εργασία εξετάζουμε το ζητούμενο των «Έξυπνων Πόλεων» δημιουργώντας μία πλατφόρμα συνεργατικότητας μεταξύ των πολιτών και της τοπικής αυτοδιοίκησης και άλλων μη κυβερνητικών οργανισμών. Επιπλέον γίνεται σύγκριση της πλατφόρμας συνεργατικότητας με άλλες σύγχρονες εφαρμογές καθώς και η ανάλυση της αρχιτεκτονικής και παρουσίαση της.

Abstract

Due to the development of Information and Communication Technologies (ICT), there is a growing tendency for municipalities and local communities globally to use this potential for more effective management of city governance. In this paper, we are looking at the aim of 'Smart Cities', by creating a collaborative platform among citizens, local government and other non-governmental organizations. Furthermore, we are comparing the collaborative platform with other modern applications and provide an analysis of its architecture as well as its presentation.

Πίνακας περιεχομένων

1	Εισαγωγή	1
1.1	Έξυπνες Πόλεις.....	1
1.2	Σκοπός της εργασίας	2
2	Πλατφόρμες συνεργατικότητας πολιτών	2
3	Ανάλυση και σχεδιασμός έξυπνης πλατφόρμας συνεργατικότητας.....	4
4	Αρχιτεκτονική πλατφόρμας	5
4.1	Ασφάλεια και Διαχείριση Χρηστών.....	6
5	Υλοποίηση πλατφόρμας και Παρουσίαση πλατφόρμας	8
5.1	Επίπεδο Δεδομένων	8
5.2	Επίπεδο Λειτουργιών	14
5.3	Επίπεδο Παρουσίασης.....	18
6	Συμπεράσματα.....	26
7	Βιβλιογραφία	27

1 Εισαγωγή

Με την εξέλιξη των Τεχνολογιών Πληροφορικής και Επικοινωνίας, αναπτύχθηκε παγκοσμίως μια τάση στους Δήμους και τις τοπικές κοινωνίες να χρησιμοποιήσουν τις δυνατότητες του Διαδικτύου για την αποτελεσματικότερη διαχείριση της διακυβέρνησης των πόλεων. Οι πόλεις μπορούν να χρησιμοποιήσουν το Διαδίκτυο για να προσφέρουν υπηρεσίες πληροφοριών, καθώς και για να διασυνδεθούν καλύτερα με τους δημότες τους. Μια ιστοσελίδα μπορεί να θεωρηθεί ως μια τοπική πύλη για την προσφορά κοινοτικών υπηρεσιών (1). Προσφέρονται έτσι στους Δημότες, νέες ευκαιρίες εμπλοκής τους στην διακυβέρνηση σε τοπικό, περιφερειακό και κεντρικό επίπεδο. Αυτή η εμπλοκή, μπορεί να ορισθεί ως η συμμετοχή σε συλλογικές ή ατομικές δραστηριότητες για την εγκατάσταση και τροφοδότηση μιας κοινότητας ενεργών πολιτών και μπορεί να προωθηθεί και να εφαρμοστεί μέσω Τεχνολογιών Πληροφορικής και Επικοινωνίας (ΤΠΕ). Η κοινότητα ενεργών πολιτών (κοινωνικό κεφάλαιο) έχει ενσωματώσει πηγές πληροφορίας, κανόνες και κοινωνικές σχέσεις, που επιτρέπουν στους πολίτες να συνεργάζονται με συλλογική δράση ώστε να πετυχαίνουν κοινούς στόχους (2). Πολλοί Δήμοι παγκοσμίως, πανευρωπαϊκά και πανελλαδικά, έχουν προχωρήσει σε υιοθέτηση των ΤΠΕ μέσω των προγραμμάτων «έξυπνη πόλη» και «έξυπνος οικισμός» (3).

Η αποδοχή και η διάδοση των νέων τεχνολογιών, αποτελούν το κλειδί για την επιτυχία τέτοιων προγραμμάτων. Αυτή η διάδοση και αποδοχή των ΤΠΕ είναι μια μάλλον αργή και επίπονη διαδικασία. Έτσι, η μελέτη και διερεύνηση της διαδικασίας αυτής, είναι απαραίτητη για την κατανόηση του τρόπου με τον οποίο ενσωματώνονται στις τοπικές κοινωνίες οι τεχνολογικές αλλαγές και των λόγων για τους οποίους μπορεί αυτή η διαδικασία να καθυστερεί κατά περιόδους και κατά τόπους (4).

1.1 Έξυπνες Πόλεις

Υπάρχουν πολλοί ορισμοί για τις έξυπνες πόλεις. Μια σειρά από εννοιολογικές παραλλαγές συχνά επιτυγχάνονται με την αντικατάσταση του «έξυπνες» με εναλλακτικά επίθετα, για παράδειγμα, «ευφυής» ή «ψηφιακή» (5).

Έργα που αφορούν τις έξυπνες πόλεις έχουν αντίκτυπο στην ποιότητα ζωής των πολιτών και σκοπεύουν να προαγάγουν περισσότερο την ενημέρωση και την συμμετοχή των πολιτών. Τέτοιες πρωτοβουλίες επιτρέπουν στα μέλη της πόλης να συμμετάσχουν στη διακυβέρνηση και τη διαχείριση της πόλης και να γίνουν ενεργοί χρήστες. Η βελτίωση της κοινωνικής και πολιτιστικής ταυτότητας κάθε πόλης επιτυγχάνεται με την συνεισφορά κάθε ατόμου. Αν είναι βασικοί παίκτες έχουν την ευκαιρία να ασχοληθούν με τις πρωτοβουλίες, στο βαθμό που μπορούν να επηρεάσουν την προσπάθεια να είναι μια επιτυχία ή μια αποτυχία. Οι πολίτες έχοντας ηλεκτρονική συμμετοχή τους επιτρέπεται να σχολιάσουν και να αξιολογήσουν τις προτάσεις και τα προγράμματα στο πλαίσιο της διοίκησης και της πολιτικής. Επιπλέον, παρέχουν στους πολίτες τη δυνατότητα να κάνουν τις δικές τους προτάσεις και να προσθέσουν τις γνώσεις και την εμπειρία. Ως εκ τούτου, έχουν απευθείας σύνδεση στις διαβουλεύσεις όπου έμμεσα είναι δημοκρατικές διαδικασίες που δεν αμφισβητούν την εξουσία λήψης αποφάσεων της πολιτικής. Αυτή η μορφή της συμμετοχής παρέχει στους συμμετέχοντες μια νέα ευκαιρία για να εκφράσουν τις ανησυχίες και την πρότασή τους στο θέμα της συζήτησης. Σε κάθε περίπτωση, υπάρχει μια μετατόπιση των ορίων: η πολιτικό-διοικητική διαδικασία γίνεται πιο προσιτή μέσω της ηλεκτρονικής συμμετοχής, αυξάνοντας την πίεση στην πολιτική και στην διοίκηση (6).

Σήμερα, υπάρχει αυξανόμενη προσοχή για τα πιθανά οφέλη της τεχνολογίας για την προώθηση των σχέσεων πολίτη-με-κυβέρνηση (C2G – Citizen2Government). Όλο και περισσότεροι άνθρωποι συμφωνούν ότι απαιτείται ένα νέο πρότυπο διακυβέρνησης, δηλαδή μία που εξελίσσεται γύρω από την ενδυνάμωση του πολίτη και την αυξημένη εμπλοκή του σε διαδικασίες της διακυβέρνησης, η-κυβέρνηση (e-government) αναδεικνύεται ως μια πιθανή λύση για την αποκατάσταση συμμετοχή των πολιτών και την διευκόλυνση της ανάπτυξης των πραγματικά συμμετοχικών κοινοτήτων σε όλα τα επίπεδα της κυβέρνησης, που παραδοσιακά ήταν κυρίως ένα προνόμιο των μικρών κοινωνιών (7).

1.2 Σκοπός της εργασίας

Σκοπός της εργασίας είναι να αναδείξει και να παρουσιάσει μία πλέον διαδεδομένη ιδέα που αφορά τις έξυπνες πόλεις, την διάδραση των πολιτών με τους αρμόδιους οργανισμούς της εκάστοτε περιοχή τους μέσω της δημοσιοποίησης προβλημάτων και δημοψηφισμάτων.

Λέξεις **Κλειδιά**: SmartCity Project, Java, MongoDB, e-petition, e-participation

2 Πλατφόρμες συνεργατικότητας πολιτών

Πληθώρα εφαρμογών έχουν ως αντικείμενο τις έξυπνες πόλεις, παρακάτω περιγράφονται τρεις εφαρμογές, το FixMyStreet, το Novonville και το Improve My City όπου έρχονται σε σύγκριση με την παρούσα εφαρμογή (Smart City Project).

Το FixMyStreet (8) καθιστά εύκολο για οποιονδήποτε να αναφέρει ένα πρόβλημα χωρίς να ανησυχεί για το ποια είναι η σωστή αρχή. Επιπλέον δεν είναι μόνο για να λαμβάνει αναφορές προβλημάτων αλλά και να καθιστά τις αναφορές ορατές σε όλους. Ο καθένας μπορεί να δει τι έχει ήδη αναφερθεί, να αφήνει ενημερώσεις, ή να εγγράφεται για τις ειδοποιήσεις. Προσφέρει επιπλέον χαρακτηριστικά για τους χρήστες του προσωπικού που εργάζεται στις αρχές που επιδιορθώνουν τα προβλήματα.

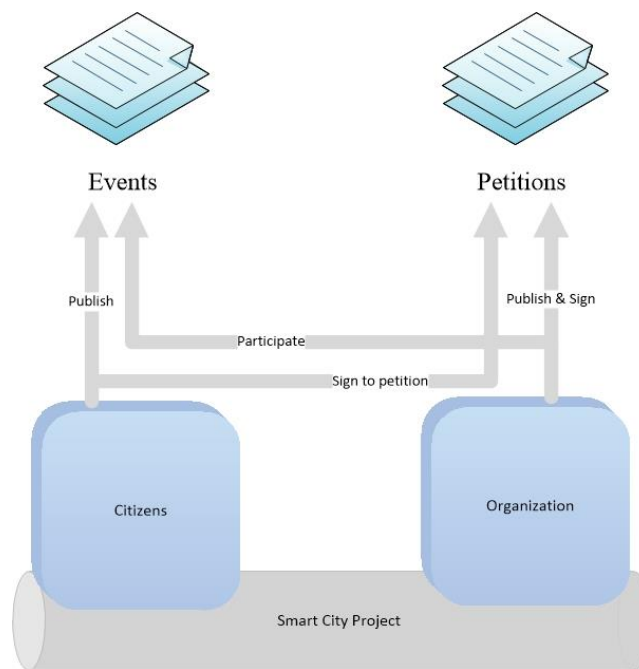
Το Novonville (9) είναι μια ηλεκτρονική πλατφόρμα καταγραφής και διαχείρισης προβλημάτων δημοσίου ενδιαφέροντος. Οι χρήστες μέσω των κινητών τους τηλεφώνων μπορούν να δηλώνουν με εύκολο και γρήγορο τρόπο προβλήματα δημοσίου ενδιαφέροντος. Παράλληλα, οι δημόσιοι φορείς όπως οι Δήμοι, χρησιμοποιούν ένα online διαχειριστικό σύστημα με διαδραστικούς χάρτες ώστε να βλέπουν σε πραγματικό χρόνο τα προβλήματα που δηλώνονται από τους πολίτες και να δρομολογούν την επίλυση τους με βέλτιστο τρόπο, κρατώντας ταυτόχρονα ενήμερους τους πολίτες για το στάδιο εξέλιξης.

Το Improve My City (10) είναι ακόμη μία πλατφόρμα διαχείρισης προβλημάτων – αναφορών που έχει υλοποιηθεί με τα πλέον διαδεδομένα CMS (Content Management System) Joomla/Wordpress. Προσφέρει την δυνατότητα δημοσιοποίησης προβλημάτων μέσω των κινητών τηλεφώνων και των φυλλομετρητών. Ακόμη επιτρέπεται ο σχολιασμός και η εκδήλωση ενδιαφέροντος καθώς όλες η αναφορές είναι ανοιχτές προς τους πολίτες. Η πλατφόρμα εγκαθιστάτε ξεχωριστά για κάθε φορέα με την δυνατότητα να διαχειρίζονται τα προβλήματα αλλά και να παρακολουθούν σχετικά στατιστικά δεδομένα.

<i>Χαρακτηριστικά</i>	<i>Novoville</i>	<i>FixMyStreet</i>	<i>Improve My City (IMCity)</i>	<i>Smart City Project (SCP)</i>
<i>Υποστήριξη πολλαπλών οργανισμών - φορέων</i>	x	✓	x	✓
<i>Υποστήριξη προβλημάτων</i>	✓	✓	✓	✓
<i>Υποστήριξη δημοψηφισμάτων</i>	x	x	x	✓
<i>Υποστήριξη σε web browsers</i>	✓	✓	✓	✓
<i>Υποστήριξη σε mobile (smartphones)</i>	✓	✓	✓	✓
<i>Ανοιχτό Λογισμικό</i>	x	✓	✓	✓

3 Ανάλυση και σχεδιασμός έξυπνης πλατφόρμας συνεργατικότητας

Το Smart City – Project (SCP) αποτελεί μία πλατφόρμα συνεργατικότητας και επικοινωνίας μεταξύ πολιτών και οργανισμών. Επικεντρώνεται στην διευθυνσιοδότηση των προβλημάτων που αφορά γεωγραφικές περιοχές και αρμόδιους εγγεγραμμένους οργανισμούς. Ακόμη δίνει έμφαση στη έννοια των δημοψηφισμάτων και των ηλεκτρονικών υπογραφών έτσι ώστε να δημιουργούνται δράσεις από τους οργανισμούς. Επιπλέον στην διαχείριση του περιεχομένου δίνεται την δυνατότητα στους χρήστες να προσθέτουν τα σχόλια τους, να ψηφίζουν και να αναφέρουν πόσο κατάλληλο είναι το περιεχόμενο που είναι δημοσιευμένο από άλλους χρήστες και οργανισμούς.



Conceptual Architecture Schema

Οι γεωγραφικές περιοχές αποτελούν ολόκληρο τον χάρτη της Ελλάδας που είναι χωρισμένος σε δήμους και μας παρέχονται από τα ανοιχτά γεωγραφικά δεδομένα (11). Αυτό μας δίνει την δυνατότητα να δημιουργούμε περιορισμούς για την δήλωση προβλημάτων, για την συνεισφορά των οργανισμών ως προς αυτά και πιο γρήγορα αποτελέσματα αναζήτησης προβλημάτων και οργανισμών.

Ένας οργανισμός μπορεί να είναι κρατικός δηλαδή κάποιος Δήμος αλλά και κάποιος μη κυβερνητικός οργανισμός (ΜΚΟ). Αυτό γίνεται λόγω της ποικιλομορφίας των προβλημάτων που δημοσιεύονται καθώς δίνεται στους πολίτες η δυνατότητα να εκφραστούν ελεύθερα καθορίζοντας το πρόβλημα τους με ετικέτες (tags).

Οι εγγεγραμμένοι οργανισμοί δηλώνουν τις περιοχές που δραστηριοποιούνται και μπορούν να συνεισφέρουν σε προβλήματα που δημοσιεύονται από του πολίτες. Έτσι οι μπορούν να έχουν καλύτερα αποτελέσματα αναζήτησης προβλημάτων.

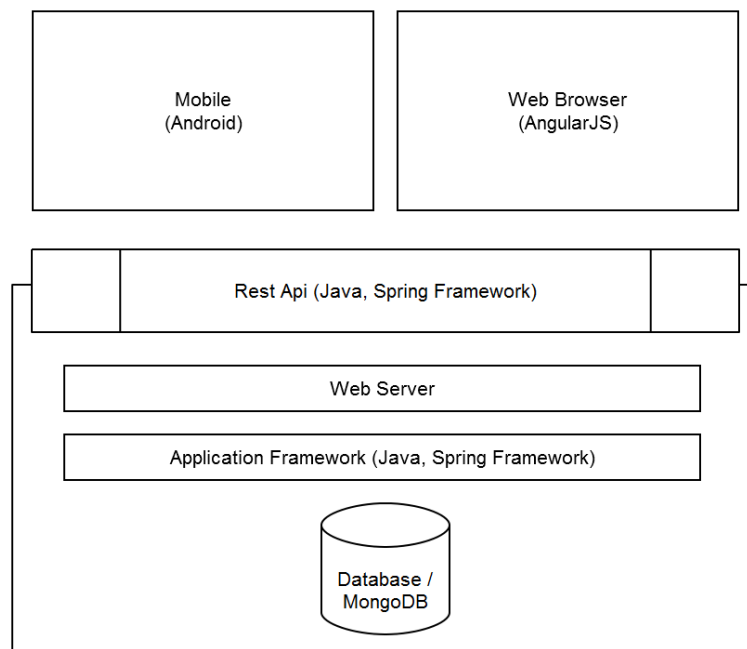
Πρέπει να τονίσουμε ότι η συγκεκριμένη πλατφόρμα αποτελεί κατά βάση ένα ενιαίο blog για εγγεγραμμένους χρήστες. Στόχος είναι η πληροφόρηση γύρω από κοινωνικά θέματα, προβλήματα που αντιστοιχούν σε γεωγραφικές περιοχές και ο ανοιχτός σχολιασμός από τους πολίτες προς τους οργανισμούς και το αντίστροφο. Όπως θα δούμε παρακάτω στο επίπεδο παρουσίασης η χρήστες έχουν προσωπική σελίδα (profile) που είναι προσβάσιμη απ' όλους τους χρήστες πολίτες ή οργανισμοί.

Στα επόμενα κεφάλαια που ακολουθούν θα αναλύσουμε την αρχιτεκτονική της πλατφόρμας, τα εργαλεία που χρησιμοποιήθηκαν και τις δυνατότητες της μέσω της ανάλυσης και του σχεδιασμού που έγινε.

4 Αρχιτεκτονική πλατφόρμας

Η πλατφόρμα έχει σχεδιαστεί χρησιμοποιώντας την κλιμακωτή προσέγγιση (3-Tier). Στο χαμηλότερο επίπεδο υπάρχει μια βάση δεδομένων - MongoDB που φιλοξενεί τα δεδομένα της πλατφόρμας. Η λογική υλοποιείται χρησιμοποιώντας την γλώσσα προγραμματισμού Java σε συνδυασμό με το Spring Framework. Τα δεδομένα και οι δράσεις που αφορούν το χρήστη εξυπηρετούνται από ένα web server όπου υλοποιείται με την stateless αρχιτεκτονική. Οι ανταλλαγές δεδομένων του Smart City Project με το UI (ιστοσελίδα ή κινητό) γίνεται μέσω RESTful API. Η web διεπαφή είναι χτισμένη χρησιμοποιώντας δυναμικά HTML με τη χρήση της AngularJS.

Η ουσία της «έλλειψης κατάστασης» (stateless) είναι ότι οποιαδήποτε κλήση σε μια υπηρεσία REST δε θα πρέπει να αναφέρεται σε άλλη προγενέστερη κλήση. Όλες οι κλήσεις πρέπει να είναι ανεξάρτητες. Ο server δε θα πρέπει να γνωρίζει το αν έγινε με κάποια προηγούμενη κλήση τη στιγμή που επεξεργάζεται την τρέχουσα κλήση. Η κατάσταση «συνδεδεμένος» διατηρείται από την υπηρεσία και ο πελάτης δεν χρειάζεται να παρουσιάσει τα πιστοποιητικά του σε περίπτωση μιας δεύτερης ή τρίτης κλήσης της υπηρεσίας. Μια τέτοια διατήρηση καταστάσεων απαγορεύεται σε δικτυακές υπηρεσίες τύπου REST.



Architecture Schema

Η MongoDB είναι μία βάση αρχείων η οποία ξεκίνησε να δημιουργείται το 2007 από την εταιρία 10gen (πλέον MongoDB) σαν μέρος ενός προϊόντος PaaS (Platform as a Service). Αργότερα η εταιρία επικέντρωσε τις προσπάθειές της μόνο στην βάση, βλέποντας τις προοπτικές που αυτή είχε. Είναι ανοιχτού λογισμικού γραμμένη σε γλώσσα C++ και σχεδιάστηκε για να καλύπτει αποτελεσματικά τις ανάγκες web εφαρμογών και με την προοπτική να μπορεί να κλιμακωθεί οριζόντια ανάλογα με τις ανάγκες, συνδυάζοντας δυνατά χαρακτηριστικά σχεσιακών βάσεων και NoSQL συστημάτων. Τα δεδομένα που αποθηκεύονται έχουν μορφή αρχείων BSON (Binary JSON).

Πρέπει να επισημάνουμε ότι για την διαχείριση της γεωγραφικής πληροφορίας χρησιμοποιούμε το GeoJSON πρότυπο όπου υποστηρίζεται από την MongoDB. Το GeoJSON (12) είναι ένα πρότυπο για την παρουσίαση γεωγραφικών δομών δεδομένων. Ένα αντικείμενο GeoJSON μπορεί να αντιπροσωπεύει μια γεωμετρία, ένα χαρακτηριστικό ή μια συλλογή χαρακτηριστικών. Το GeoJSON υποστηρίζει τους ακόλουθους τύπους γεωμετρίας: Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon και GeometryCollection.

Ακόμη η αρχιτεκτονική λογισμικού κατανεμημένων δικτύων τύπου REST, όντας η κυρίαρχη αρχιτεκτονική στο διαδίκτυο αυτή τη στιγμή πάνω στην οποία βασίζεται το WorldWideWeb, είναι ουσιαστικά ένας τρόπος υλοποίησης και καταμερισμού διεργασιών για την δημιουργία διαδικτυακών εφαρμογών και υπηρεσιών.

Τα κύρια χαρακτηριστικά της αρχιτεκτονικής αυτού του τύπου είναι η ευκολία πρόσβασης στις υπηρεσίες και τις εφαρμογές που παρέχει απομακρυσμένα σε όποιον τις ζητήσει με κατάλληλο προκαθορισμένο τρόπο, η επεκτασιμότητα που προσφέρει με τον τρόπο που δομείται το σύστημα και τέλος η ανεξάρτητη ανάπτυξη, καθώς και η γενίκευση των επιμέρους υπηρεσιών που παρέχονται με σκοπό την ευρύτερη χρήση τους.

Το REST API της εφαρμογής προσφέρει στους εγγεγραμμένους χρήστες την δυνατότητα καταχώρησης, επεξεργασίας και αναζήτησης δεδομένων.

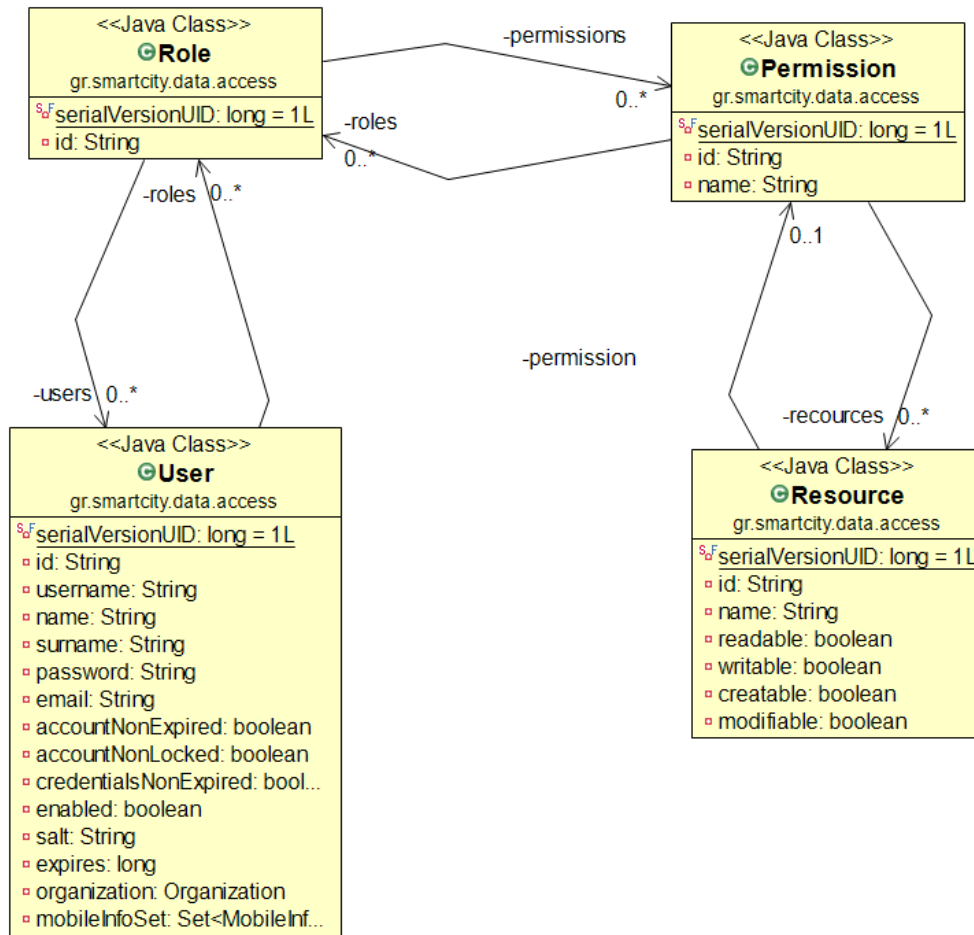
Τέλος το AngularJS είναι ένα MVC(Model View Controller) Framework για δυναμικές εφαρμογές web. Αυτό μας επιτρέπει να χρησιμοποιήσουμε την HTML ως γλώσσα πρότυπό σας και μας επιτρέπει να επεκτείνουμε την HTML σύνταξη για να εκφράσουμε τα στοιχεία της εφαρμογής μας.

4.1 Ασφάλεια και Διαχείριση Χρηστών

Η πλατφόρμα Smart City Project χρησιμοποιεί την λογική του token για τον έλεγχο ταυτοποίησης και της εξουσιοδότησης για το API όπου είναι αναγκαίο. Μερικές από τις περιπτώσεις που είναι αναγκαία η εξουσιοδότηση είναι η καταγραφή προβλημάτων στις εκάστοτε γεωγραφικές περιοχές, η ψηφοφορία, ο σχολιασμός, η επεξεργασία του δημοσιευμένου περιεχομένου, την προβολή των προφίλ. Όταν ένας χρήστης είναι ήδη συνδεδεμένος, τα αιτήματα προς το API εκτελούνται με τέτοιο τρόπο ώστε το token να παρέχεται με κάθε αίτημα για τους σκοπούς της εξουσιοδότησης και της καταγραφής πληροφοριών στο σύστημα.

Βασική λειτουργία της εφαρμογής είναι η υποστήριξη χρηστών και ο έλεγχος πρόσβασης ακολουθώντας την προσέγγιση του RBAC (Role Based Access Control) (13). Ο σχεδιασμός απεικονίζεται στο παρακάτω σχήμα. Σύμφωνα με αυτές ορίζονται η έννοια του χρήστη (**User**). Κάθε χρήστης μπορεί να έχει έναν ή περισσότερους ρόλους (**Role**). Κάθε ρόλος έχει ένα ή περισσότερα δικαιώματα (**Permission**).

Κάθε δικαίωμα έχει ορίσει μία ή περισσότερες προσβάσεις σε οντότητες του της εφαρμογής (**Resources**). Συγκεκριμένα η πρόσβαση (Resource) περιέχει το όνομα μιας οντότητας (Resource), όπως αυτές που θα δούμε στο επίπεδο δεδομένων (π.χ. Event) και το επίπεδο πρόσβασης (readable, writable, creatable, modifiable).



Class Diagram 1

Μέσω αυτής της πληροφορίας για τους χρήστες και τους ρόλους τους είναι δυνατόν να γίνει έλεγχος πρόσβασης ανά χρήστη και ρόλο σε επίπεδο οντότητας του συστήματος. Τέλος στην εφαρμογή έχει οριστεί ασφάλεια με το RBAC στις οντότητες: Event, Petition, ReportDocument.

Οι ρόλοι που έχουν ενσωματωθεί στη πλατφόρμα είναι δύο, συγκεκριμένα ο ρόλος του πολίτη (ROLE_USER) και ο ρόλος του οργανισμού (ROLE_ADMIN). Οι δύο αυτοί ρόλοι διαφέρουν ως προς τα δικαιώματα και τις δυνατότητες στην διαχείριση των δεδομένων.

5 Υλοποίηση πλατφόρμας και Παρουσίαση πλατφόρμας

Ο σχεδιασμός της εφαρμογής χωρίζεται σε 3 επίπεδα, σε αυτό το κεφάλαιο θα δούμε το επίπεδο δεδομένων, το επίπεδο λειτουργιών και το επίπεδο παρουσίασης. Η αποθήκευση εικόνων δεν υποστηρίζεται από την εφαρμογή σε επίπεδο αρχείων αλλά σε επίπεδο συνδέσμων (links), αυτό γίνεται για να έχουμε μεγαλύτερη ταχύτητα στις απαντήσεις των αιτημάτων που γίνονται στο σύστημα.

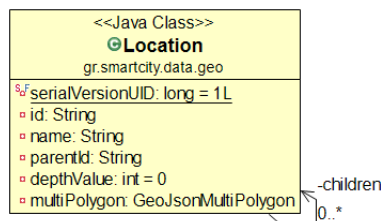
5.1 Επίπεδο Δεδομένων

Οι γεωγραφικές περιοχές έχουν παρθεί από τα ανοιχτά δεδομένα, συγκεκριμένα είναι ένα αρχείο τύπου KML όπου και ενσωματώσαμε στο σύστημα μας για να μπορούμε να καθορίσουμε σε ποια γεωγραφική περιοχή βρίσκεται ένα πρόβλημα πέρα του γεωγραφικού στίγματος. Ακόμη μπορούμε να ορίσουμε τις περιοχές όπου δραστηριοποιούνται οι οργανισμοί. Παρακάτω βλέπουμε ένα στιγμιότυπο από το αρχείο με τις γεωγραφικές περιοχές.

Η KML (14) είναι μια μορφή αρχείου που χρησιμοποιείται για την εμφάνιση των γεωγραφικών δεδομένων σε ένα πρόγραμμα περιήγησης. Η KML χρησιμοποιεί μια δομή βασισμένη σε ετικέτες ένθετα στοιχεία και χαρακτηριστικά και βασίζεται στο πρότυπο XML.

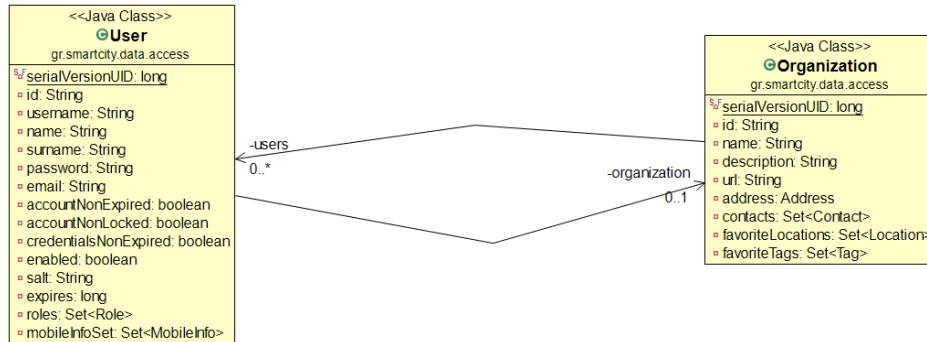
```
<Placemark>
  <name>Αθως</name>
  <Style><LineStyle><color>ff0000ff</color></LineStyle>
<PolyStyle><fill>0</fill></PolyStyle></Style>
  <ExtendedData><SchemaData schemaUrl="#oria_dhmnw_kallikraths">
    <SimpleData name="Name">Αθως</SimpleData>
    <SimpleData name="gid">2</SimpleData>
  </SchemaData></ExtendedData>
  <MultiGeometry>
    <Polygon>
      <outerBoundaryIs>
        <LinearRing>
          <coordinates></coordinates>
        </LinearRing>
      </outerBoundaryIs>
    </Polygon>
  </MultiGeometry>
</Placemark>
```

Τα στοιχεία που αποθηκεύουμε είναι τα: name, gid, multi geometry, στο σχήμα που ακολουθεί.



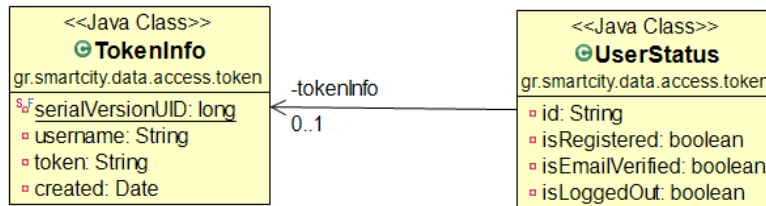
Class Diagram: Geographic location data model to support a map

Στα σχήματα που ακολουθούν βλέπουμε την σχέση μεταξύ του οργανισμού με του χρήστες, το μοντέλο υποστηρίζει πολλούς χρήστες ανά οργανισμό με σκοπό την διαχείριση από διάφορα άτομα που ανήκουν στον οργανισμό.



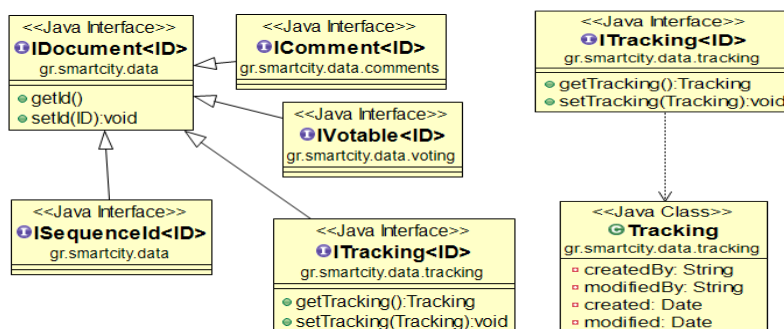
Class Diagram: User-Organization data model

Για να έχουμε γνώση της κατάστασης του χρήστη έχουμε δημιουργήσει τη συλλογή – πίνακα UserStatus όπου αποθηκεύουμε ένα αντικείμενο τύπου TokenInfo, ένα id όπου είναι το id του χρήστη και κάποιες επιπλέον βασικές πληροφορίες.



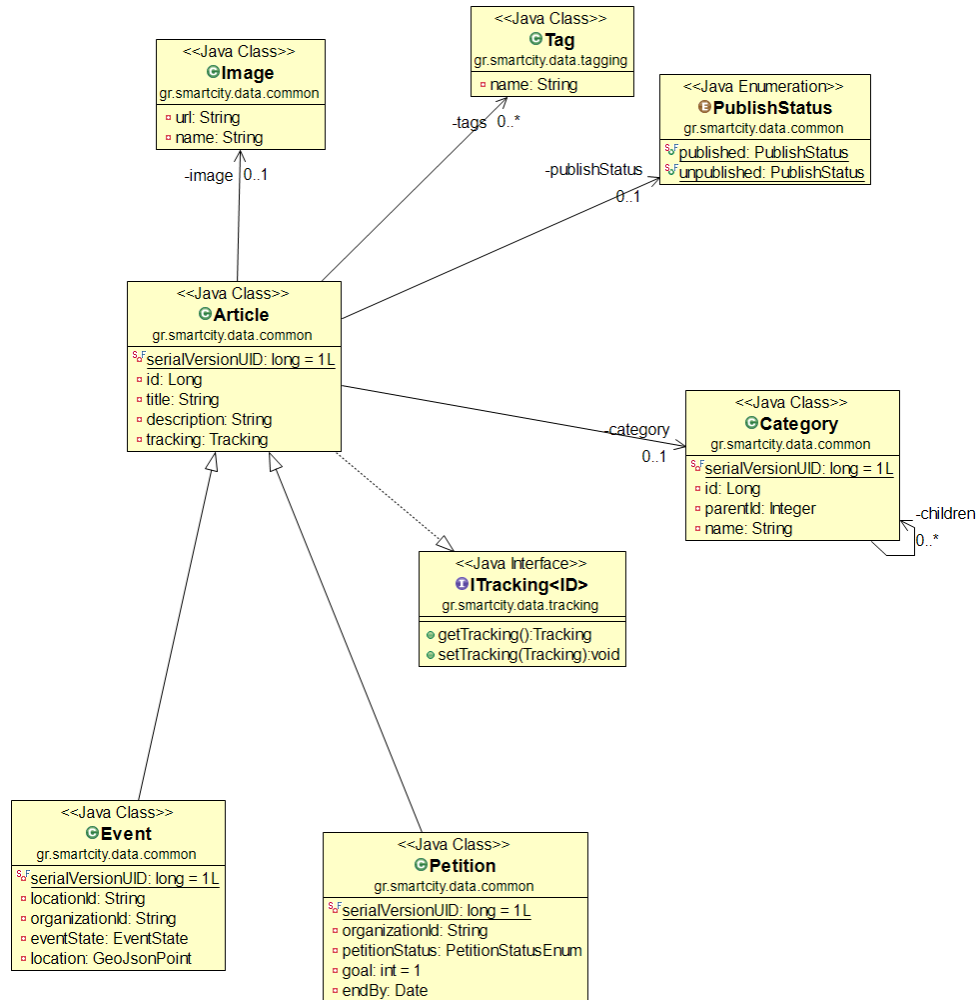
Class Diagram: User-Token data model

Όλες οι οντότητες που αντιπροσωπεύουν μία συλλογή – πίνακα στην MongoDB κατ'άγονται από την IDocument, οι οντότητες που κληρονομούν τις ιδιότητες της IDocument όπως φαίνετε στο παρακάτω σχήμα δηλώνουν την υποστήριξη που μπορούν να έχουν κάποιες κλάσεις (Events, Petitions). Για να μπορούμε να κρατάμε βασικές πληροφορίες δημιουργίας και τροποποίησης (createdBy, created, modifiedBy, modified) κάποιων αντικειμένων χρησιμοποιούμε την ITracking.



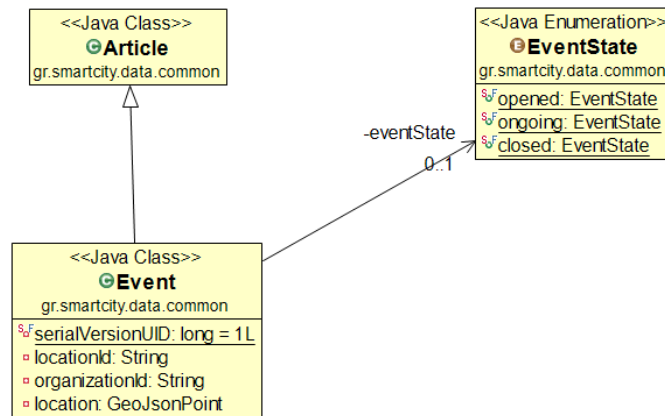
Class Diagram: Base repository data model

Οι οντότητες των Events και των Petitions είναι δομημένα ως προς την οντότητα του Article λόγω του ότι έχουν την έννοια μιας δημοσίευσης σχήμα που περιέχει έναν τίτλο, μία περιγραφή, μία συλλογή από ετικέτες, την κατάσταση δημοσίευσης (published, unpublished), ένα αντικείμενο τύπου Image που περιγράφει μία εικόνα με ένα όνομα και ένα σύνδεσμο url.



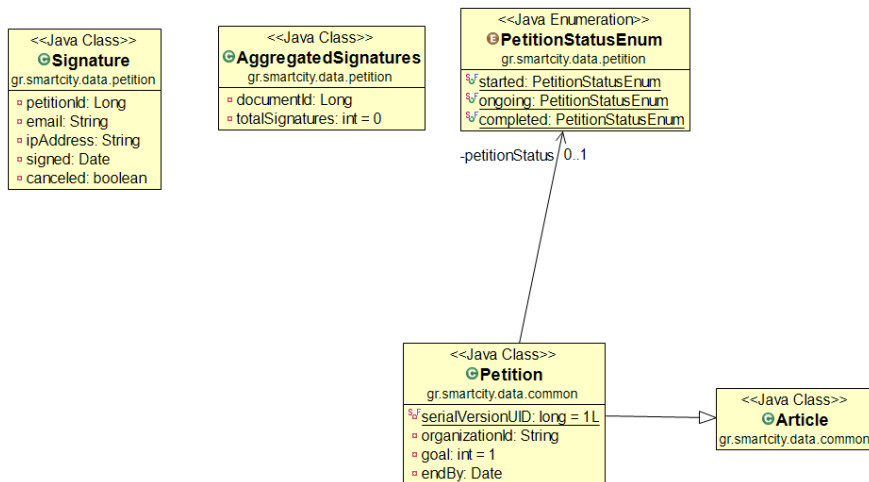
Class Diagram: Base content data model

Οι πληροφορίες που χρειαζόμαστε για την αναπαράσταση των προβλημάτων μίας περιοχής είναι το γεωγραφικό πλάτος και μήκος για την αντιστοίχιση του γεωγραφικού στίγματος με μία περιοχή στο χάρτη, που όπως έχουμε αναφέρει ο χάρτης έχει ενσωματωθεί στο σύστημα μας. Επιπλέον ένα Event περιγράφεται με μία κατάσταση (EventState: open, ongoing, closed), την συγκεκριμένη κατάσταση μπορεί να τη τροποποιήσει μόνο ο αρμόδιος οργανισμός που έχει αναλάβει το πρόβλημα.



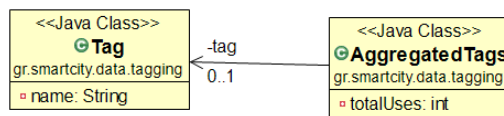
Class Diagram: Event data model

Η εφαρμογή όπως έχουμε προαναφέρει υποστηρίζει δημοψηφίσματα (petitions) όπου δημοσιεύονται μόνο από εγγεγραμμένους οργανισμούς. Έχουν την δυνατότητα να συλλέξουν ψηφιακές υπογραφές από όλους τους χρήστες ή και από άλλους οργανισμούς. Μία ψηφιακή υπογραφή χρειάζεται το petitionId, ένα email, την ip διεύθυνση για να γνωστοποιούμε ένα μη εγγεγραμμένο χρήστη. Για τις ψηφιακές υπογραφές δημιουργούμε στατιστικά (AggregatedSignatures) για να έχουμε αποθηκευμένη την πληροφορία των συνολικών υπογραφών. Ένα δημοψήφισμα περιγράφεται με μία κατάσταση (PetitionStatusEnum: started, ongoing, completed) όπου αλλάζει αυτόματα σύμφωνα με την τελική ημερομηνία επίτευξης (endBy) και τον τελικό στόχο των ψήφων που θέλει ένας οργανισμός να συλλέξει. Τέλος ο τελικός στόχος των ψήφων δεν είναι κάποια δικλίδα, ο οργανισμός μπορεί να συλλέξει παραπάνω υπογραφές αν θέλει.



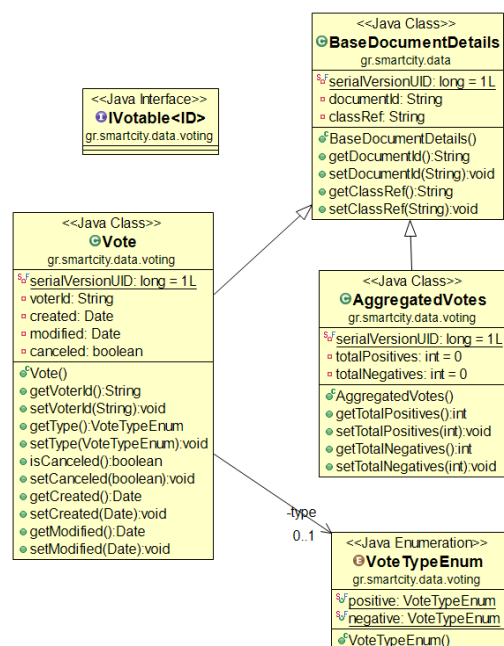
Class Diagram: Petition-signature data model

Για να προσδιορίσουμε δυναμικά την κατηγορία στα περιεχόμενα που δημοσιεύονται από τους απλούς χρήστες και τους οργανισμούς ενσωματώσαμε την λογική των Ετικετών (Tags) όπου παράγονται από τους χρήστες καθώς δημιουργούν περιεχόμενο, αυτό μας βοηθά να δημιουργούμε πίσω στην βάση μας στατιστικά όπου υπολογίζουμε την συνολική χρησιμότητα ενός tag από τους χρήστες.



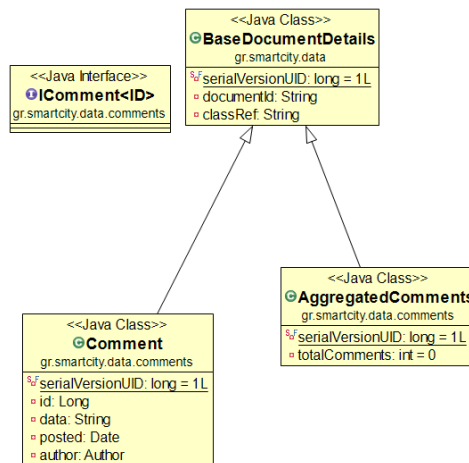
Class Diagram: Tagging data model

Το σημαντικό προτέρημα της εφαρμογής είναι η εκδήλωση της διάθεσης των χρηστών ως προς το δημοσιευμένο περιεχόμενο. Στο σύστημα μας αποθηκεύουμε τις ψήφους των χρηστών για το περιεχόμενο, τα στοιχεία που χρειαζόμαστε είναι το voterId που είναι το id του χρήστη, τις ημερομηνίες δημιουργίας και τροποποίησης και το αν έγινε ακύρωση της. Επιπλέον δημιουργούμε στατιστικά με το πόσες συνολικά θετικές και αρνητικές ψήφους έχει το περιεχόμενο. Λόγω της δυναμικότητας της ψηφοφορίας τα στοιχεία που πρέπει να γνωρίζουμε είναι το είδος του περιεχομένου (classRef) και το id (documentId).



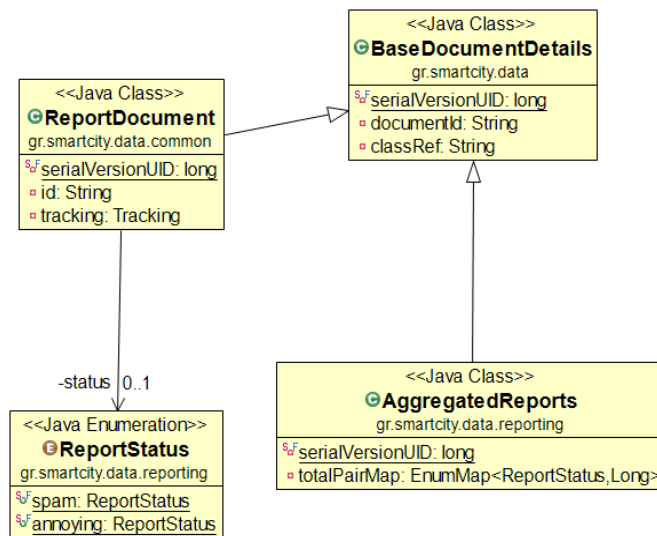
Class Diagram: Voting data model

Μία ακόμη λειτουργία της εφαρμογής είναι η υποστήριξη των σχολίων από τους χρήστες ως προς το δημοσιευμένο περιεχόμενο. Στο σύστημα μας αποθηκεύουμε τα σχόλια των χρηστών για το περιεχόμενο, τα στοιχεία που χρειαζόμαστε είναι τα δεδομένα του σχόλιου (data), την ημερομηνία που δημοσιεύτηκε (posted). Επιπλέον δημιουργούμε στατιστικά με το πόσα συνολικά σχόλια έχει το περιεχόμενο. Λόγω της δυναμικότητας των σχολίων τα στοιχεία που πρέπει να γνωρίζουμε είναι το είδος του περιεχομένου (classRef) και το id (documentId).



Class Diagram: Commenting data model

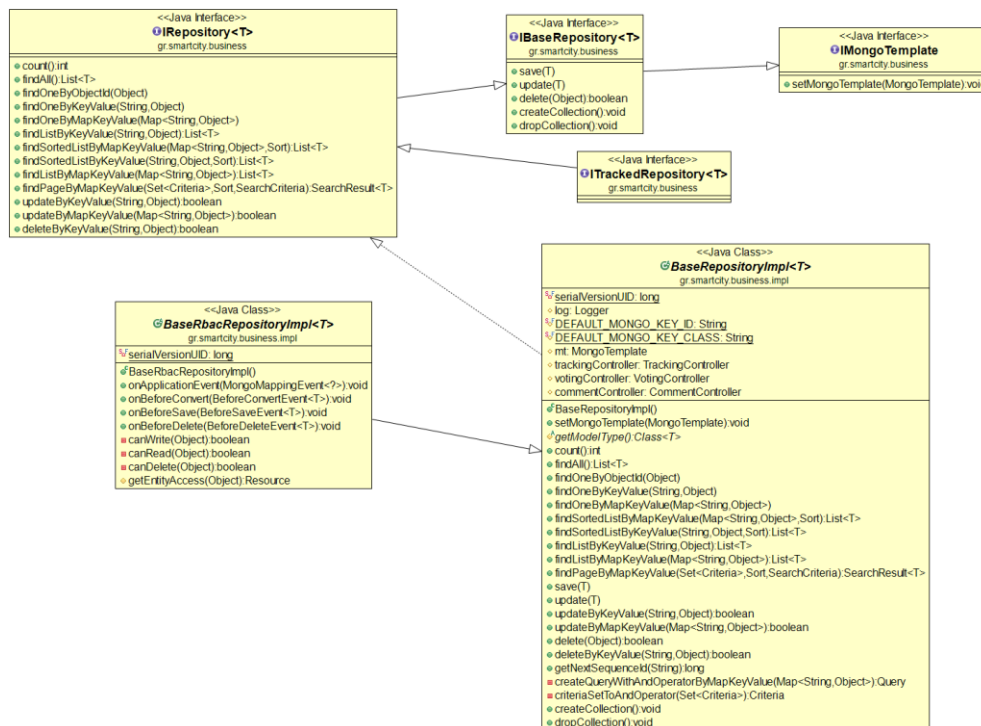
Τέλος σημαντική λειτουργία αποτελεί και η υποστήριξη της αναφοράς από τους χρήστες ως προς το δημοσιευμένο περιεχόμενο. Στο σύστημα μας αποθηκεύουμε τις αναφορές των χρηστών για το περιεχόμενο, τα στοιχεία που χρειαζόμαστε είναι το είδος της αναφοράς (ReportStatus: spam, annoying). Επιπλέον δημιουργούμε στατιστικά με το πόσα συνολικά spam ή annoying αναφορές έχει το περιεχόμενο. Λόγω της δυναμικότητας των σχολίων τα στοιχεία που πρέπει να γνωρίζουμε είναι το είδος του περιεχομένου (classRef) και το id (documentId). Ο χρήστης μπορεί να κάνει μόνο μία αναφορά για κάποιο περιεχόμενο με τον περιορισμό ότι δεν μπορεί το κάνει για περιεχόμενα που έχει δημιουργήσει ο ίδιος.



Class Diagram: Reporting data model

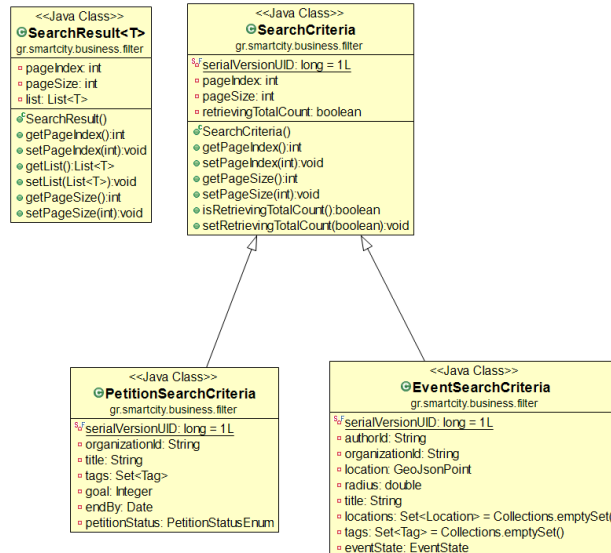
5.2 Επίπεδο Λειτουργιών

Για την διαχείριση των Repositories όπως τα ορίζει η MongoDB δημιουργήθηκε μία δομή που υποστηρίζει τις βασικές λειτουργίες όπου παρουσιάζονται και στο σχήμα. Πρέπει να τονίσουμε ότι στις περιπτώσεις που χρειαζόμαστε να παρακολουθήσουμε την δραστηριότητα των χρηστών γύρω από τα δεδομένα που δημιουργούν ή τροποποιούν τότε χρησιμοποιούμε την κλάση BaseRbacRepositoryImpl αλλιώς την BaseRepositoryImpl.



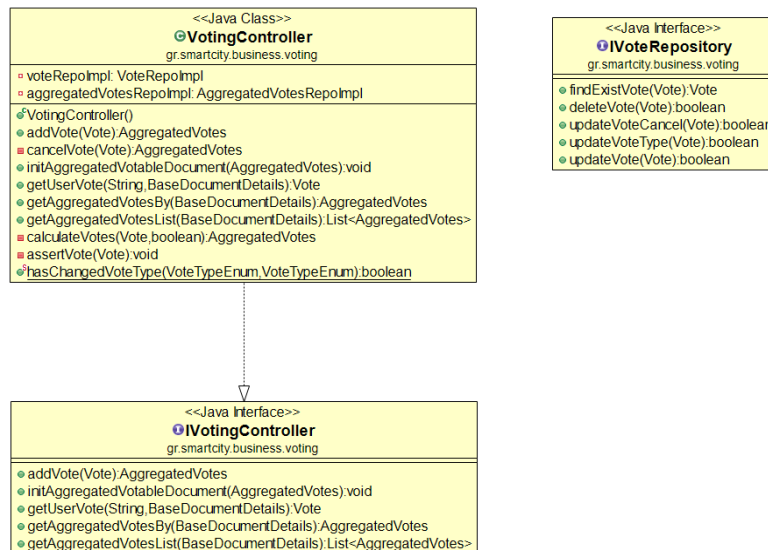
Class Diagram: MongoDB Repository business model

Λόγω διαχείρισης αρκετών δεδομένων η πλατφόρμα υποστηρίζει σελιδοποίηση δεδομένων κυρίως σε αιτήματα δυναμικής αναζήτησης (SearchCriteria). Τα αποτελέσματα που επιστρέφονται κατάγονται από την οντότητα SearchResult όπου περιέχει τον αριθμό της σελίδας, το μέγεθος των δεδομένων που έχουμε θέσει ανά σελίδα και τα δεδομένα που βρέθηκαν.



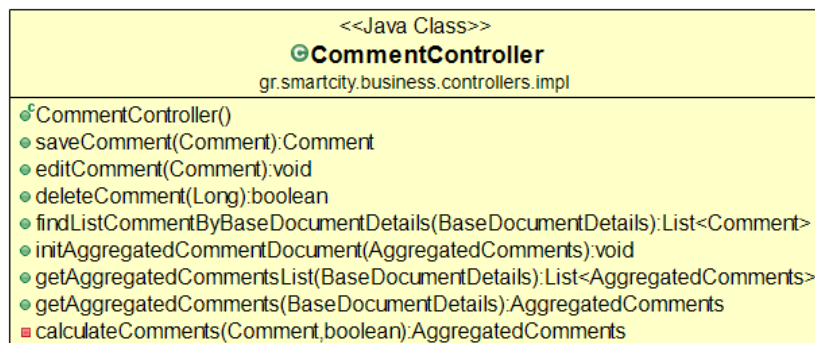
Class Diagram: Dynamic search criteria business model

Για να διαχειριστούμε 2 repositories της εφαρμογής μας, συγκεκριμένα τα vote, aggregatedVotes δημιουργήσαμε ένας controller όπου δώσαμε όλες τις πιθανές περιπτώσεις στην μέθοδο addVote προκειμένου να ξέρουμε αν ένας χρήστης ψηφίζει για πρώτη φορά, αν αλλάζει την ψήφο του, αν ακύρωσε την ψήφο του.



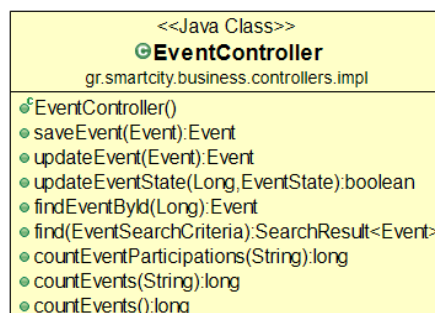
Class Diagram: Voting controller business model

Ακολουθώντας την ίδια λογική, δημιουργήσαμε ένα controller για να διαχειριστούμε τα σχόλια των προβλημάτων και των δημοψηφισμάτων που είναι δημοσιευμένα. Παρακάτω βλέπουμε την κλάση με τις απαραίτητες μεθόδους για την διαχείριση των σχολίων.



Class Diagram: Comment controller business model

Ακόμη για την διαχείριση προβλημάτων δημιουργήθηκε ο EventController απ' όπου μπορούμε να αποθηκεύσουμε ένα καινούριο πρόβλημα, να ανανεώσουμε ένα ήδη υπάρχων, να κάνουμε δυναμική αναζήτηση.



Class Diagram: Event controller business model

Για την διαχείριση των δημοψηφισμάτων, ο PetitionController περιέχει όλες τις κατάλληλες μεθόδους διαχείρισης ενσωματώνοντας και τα σενάρια χρήσης για την διαχείριση των ηλεκτρονικών υπογραφών, συγκεκριμένα η καταχώριση της υπογραφής, η ακύρωση της.

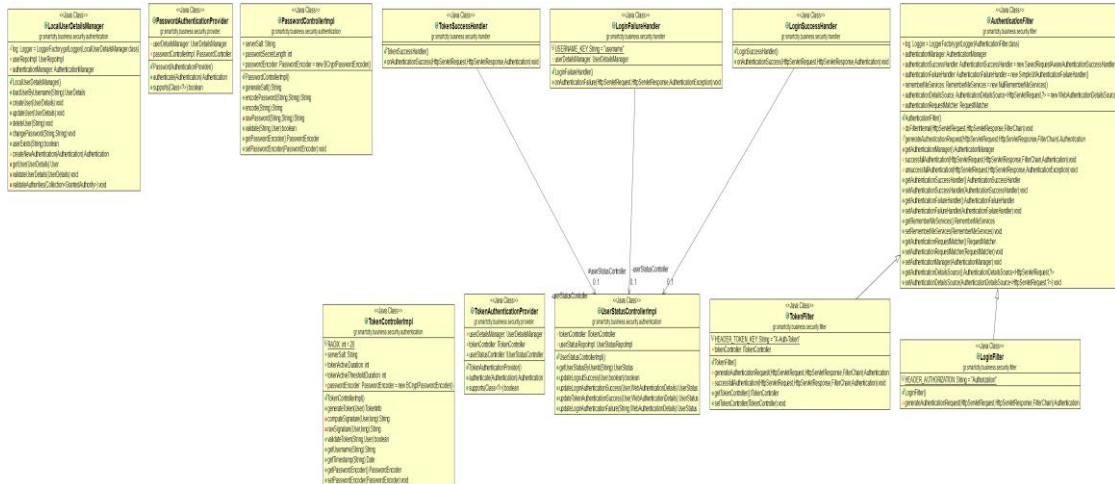
```

<<Java Class>>
PetitionController
gr.smartcity.business.controllers.impl

PetitionController()
PetitionController(PetitionRepoImpl,SignatureRepoImpl,AggregatedSignaturesRepoImpl,TagController)
signToPetition(Signature):AggregatedSignatures
cancelSignature(Signature):AggregatedSignatures
calculateSignatures(Signature):AggregatedSignatures
savePetition(Petition):Petition
updatePetition(Petition):Petition
deletePetition(Long):boolean
getAggregatedSignaturesByPetitionId(Long):AggregatedSignatures
findPetitionById(Long):Petition
findListByEndDateAndNotComplete(Date):List<Petition>
modifyPetitionStatusByEndDateAndNotComplete(Date,PetitionStatusEnum):void
find(PetitionSearchCriteria):SearchResult<Petition>
countPetitions():long
countPetitions(String):long
  
```

Class Diagram: Petition controller business model

Σε επίπεδο ασφάλειας και όπως προαναφέραμε λόγω της stateless αρχιτεκτονικής πρέπει να διαχειριστούμε tokens που αποτελούν διαπιστευτήρια των χρηστών που επικοινωνούν με την εφαρμογή. Γι' αυτό το λόγο πρέπει να κάνουμε κρυπτογράφηση των δεδομένων που δίνουμε στο token, αυτό γίνεται μέσω της κλάσης PasswordControllerImpl απ' όπου βλέπουμε αν το token έχει χρονικά λήξει. Όλα τα αιτήματα προωθούνται από το AuthenticationFilter στα Login και Token Filter, όπου βλέπουμε τις ετικέτες του αιτήματος προκειμένου να προχωρήσουμε στην κατάλληλη ενέργεια.

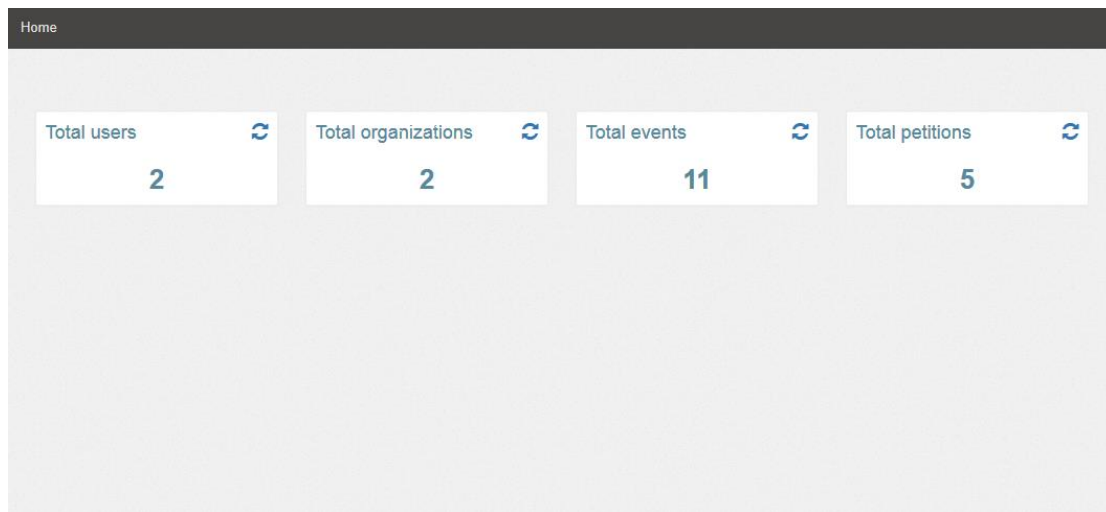


Class Diagram: Security business model

5.3 Επίπεδο Παρουσίασης

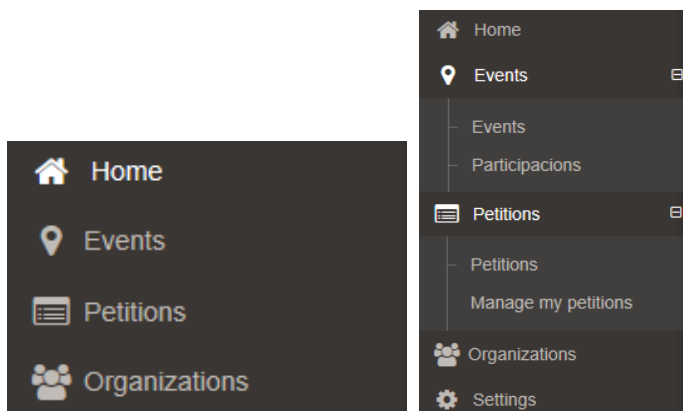
Στην ενότητα αυτή θα δούμε την web εφαρμογή που δημιουργήθηκε για να αναδείξει τις δυνατότητες της πλατφόρμας. Στο σύστημα έχουν οριστεί 2 ρόλοι αυτός του Πολίτη (ROLE_USER) και αυτός του Οργανισμού (ROLE_USER_ADMIN), έτσι και στις σελίδες που δημιουργήθηκαν υπάρχουν διαφορετικές δυνατότητες για τον εκάστοτε ρόλο.

Αρχικά βλέπουμε την πρώτη σελίδα που παρουσιάζονται συγκεντρωτικά οι συνολικοί εγγεγραμμένοι πολίτες-οργανισμοί, τα συνολικά προβλήματα-δημοψηφίσματα που έχουν δημοσιευτεί.



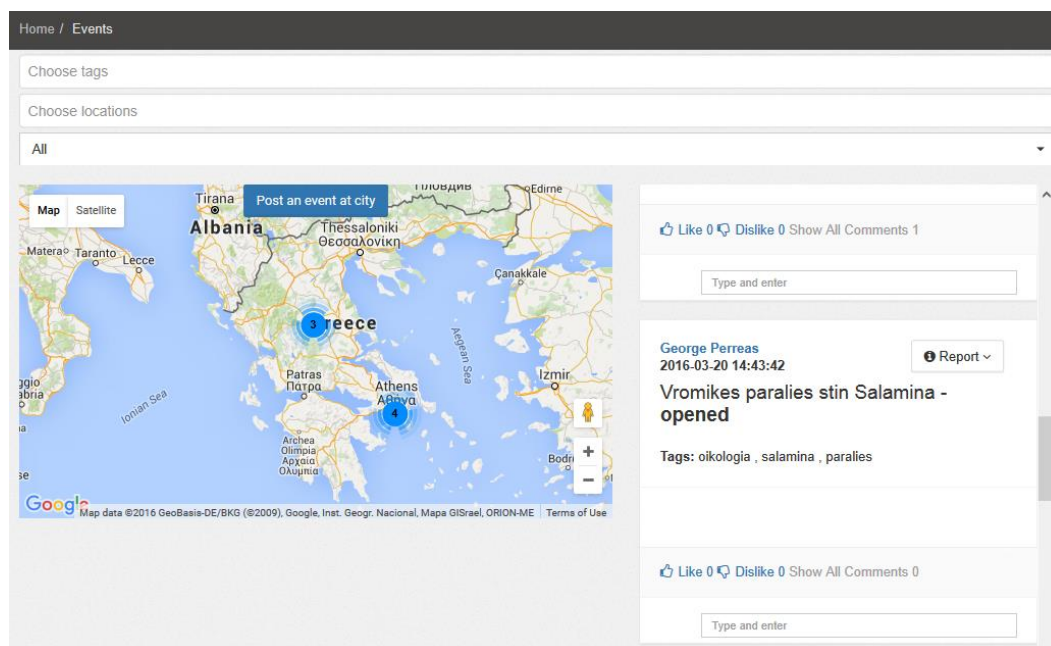
Web Interface: Home page

Το menu του χρήστη διαφέρει σύμφωνα με τον ρόλο του. Στα δεξιά βλέπουμε ότι ένας οργανισμός έχει περισσότερες δυνατότητες από ένα πολίτη. Συγκεκριμένα ένας οργανισμός είναι σε θέση να διαχειριστεί τα προβλήματα στα οποία έχει επιλέξει να συνεισφέρει καθώς να διαχειριστεί και τα δημοψηφίσματα που έχει δημιουργήσει. Τέλος μπορεί μέσω της επιλογής Settings να επεξεργαστεί το προφίλ του.



Web Interface: Menu view per role

Τα προβλήματα της πόλης (events) είναι μία από τις κύριες σελίδες της εφαρμογής, οι πολίτες έχουν την δυνατότητα να δημοσιεύσουν κάποιο πρόβλημα, να κάνουν αναζήτηση επιλέγοντας ετικέτες (tags) περιοχές ή/και την κατάσταση που μπορεί να βρίσκεται κάποιο πρόβλημα. Επιπλέον όλα τα προβλήματα που είναι στα δεξιά παρουσιάζονται και στο χάρτη αριστερά κάνοντας τα διαδραστικά με τις γεωγραφικές περιοχές παρουσιάζοντας τα συγκεντρωτικά ανάλογα με τη μεγέθυνση που έχει γίνει από τον χρήστη.



Web Interface: Events page

Ένα πρόβλημα παρουσιάζεται όπως την παρακάτω εικόνα όπου ο κάθε χρήστης μπορεί να δείξει την προτίμηση του κάνοντας “like” ή “dislike” ή/και γράφοντας κάποια σχόλια. Επιπλέον αν νιώθει ότι προσβάλλει τον ίδιο ή τους συμπολίτες του το συγκεκριμένο πρόβλημα μπορεί να το αναφέρει κάνοντας “Report”. Τέλος υπάρχει σύνδεσμος σε προφίλ και για τον δημιουργό του προβλήματος αλλά και στα δημοσιευμένα σχόλια.

T. Papas
2016-03-16 03:12:00

Diskoli i odiki kikloforia - opened

Tags: botiliarisma , odiki-kikloforia

logo kakis simansis

Report ▾

Like 1
Dislike 0
Show All Comments 1

Chalandri council
2016-03-17 00:43:09

prospathoume me tin dimotiki arxi na lisoume to provlima

Type and enter

Web Interface: Event view

Για την διαχείριση της κατάστασης των προβλημάτων που συμβάλει ο οργανισμός υπάρχει μία επιπλέον σελίδα, η “Participations”. Ο οργανισμός βλέπει συγκεντρωτικά σε ένα πίνακα τα βασικά στοιχεία ενός προβλήματος, πιο συγκεκριμένα την κατάσταση του (Event state) και τις ημερομηνίες δημιουργίας (Created) και τροποποίησης (Modified). Ο οργανισμός έχει δικαίωμα να τροποποιήσει μόνο την κατάσταση του προβλήματος.

Home / Events / Participations

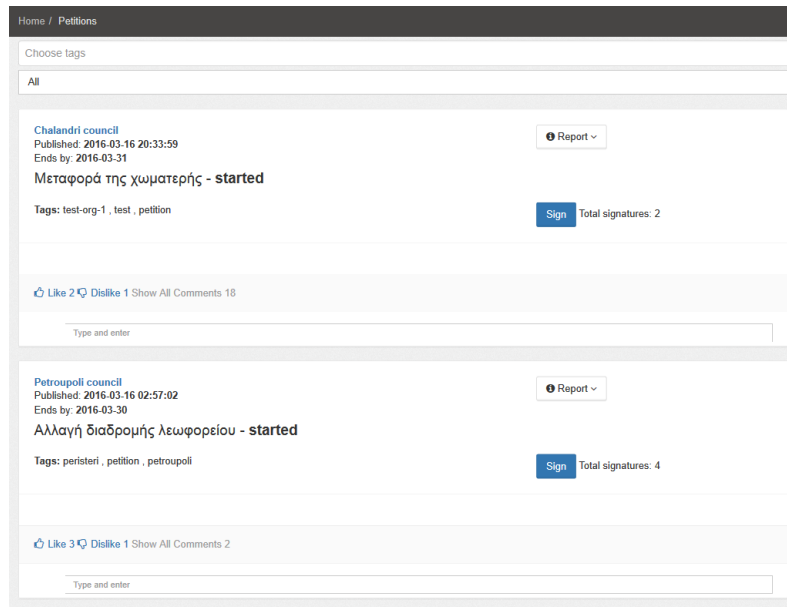
Participated Events

Title	EventState	Created	Modified	Actions
Λακούβα σε δρόμο	On going ▾	2016-03-21 23:48:31	2016-04-13 22:30:13	Unparticipate
Test Chalandri	Opened ▾	2016-03-20 17:41:32	2016-04-07 18:30:42	Unparticipate

[Previous](#) [Next](#)

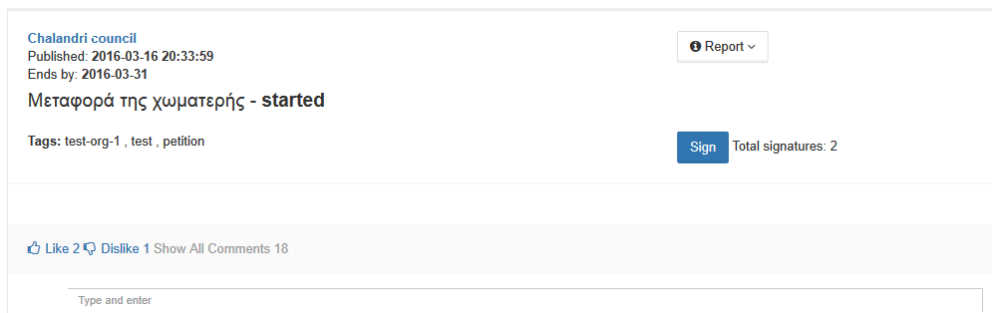
Web Interface: Manage participations page

Τα δημοψηφίσματα (petitions) είναι εξίσου σημαντική σελίδα της εφαρμογής, οι χρήστες, να κάνουν αναζήτηση επιλέγοντας ετικέτες (tags) περιοχές ή/και την κατάσταση που μπορεί να βρίσκεται κάποιο πρόβλημα. Επιπλέον όλα τα προβλήματα που είναι στα δεξιά παρουσιάζονται και στο χάρτη αριστερά κάνοντας τα διαδραστικά με τις γεωγραφικές περιοχές παρουσιάζοντας τα συγκεντρωτικά ανάλογα με τη μεγέθυνση που έχει γίνει από τον χρήστη.



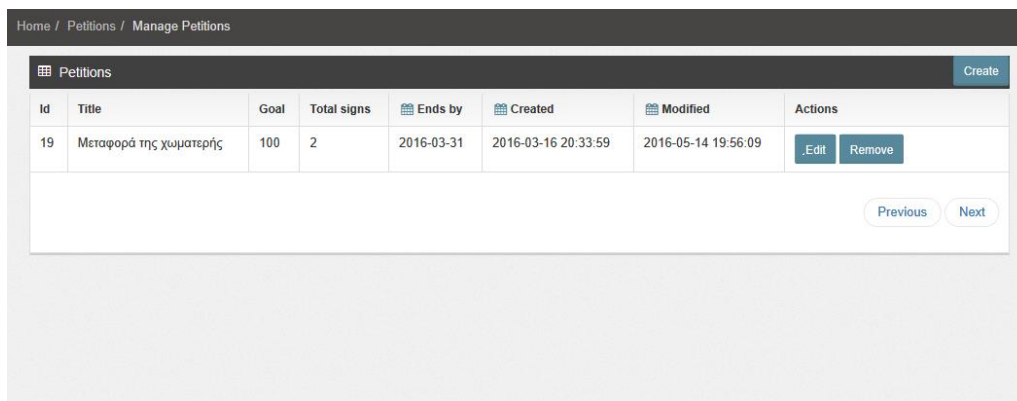
Web Interface: Petition page

Ένα δημοψήφισμα παρουσιάζεται όπως την παρακάτω εικόνα όπου ο κάθε χρήστης μπορεί να δείξει την προτίμηση του κάνοντας “like” ή “dislike” ή γράφοντας κάποια σχόλια. Επιπλέον αν νιώθει ότι προσβάλλει τον ίδιο ή τους συμπολίτες του το συγκεκριμένο δημοψήφισμα μπορεί να το αναφέρει κάνοντας “Report”. Τέλος υπάρχει σύνδεσμος σε προφίλ και για τον δημιουργό του δημοψηφίσματος αλλά και στα δημοσιευμένα σχόλια.



Web Interface: Petition view

Για την διαχείριση και την δημιουργία νέων δημοψηφισμάτων από τους οργανισμούς υπάρχει μία επιπλέον σελίδα, η “Manage Petitions”. Ο οργανισμός βλέπει συγκεντρωτικά σε ένα πίνακα τα βασικά στοιχεία ενός δημοψηφίσματος, πιο συγκεκριμένα τον στόχο (Goal) που έχει θέσει για τις συνολικές ηλεκτρονικές υπογραφές, τις συνολικές υπογραφές που έχει συγκεντρώσει (Total signs), την καταλυτική ημερομηνία (Ends by) που έχει θέσει καθώς και τις ημερομηνίες δημιουργίας (Created) και τροποποίησης (Modified).




Id	Title	Goal	Total signs	Ends by	Created	Modified	Actions
19	Μεταφορά της χωματερής	100	2	2016-03-31	2016-03-16 20:33:59	2016-05-14 19:56:09	Edit Remove

Web Interface: Manage petitions page

Ακόμη οι χρήστες της εφαρμογής έχουν την προσωπική τους σελίδα προφίλ αντίστοιχη με τον ρόλο που διαθέτουν. Στο προφίλ ενός πολίτη μπορούμε να δούμε το ονομ/πώνυμο του και τα συνολικά προβλήματα που έχει δημοσιεύσει.

Home / Profile



George Perreas

7
Total published Events

Events

George Perreas
2016-03-25 12:29:04
Report

Parousiastike fotia, voitheia! - opened

Tags: fire , trikala , trikkaiwn

Like 1
Dislike 0
Show All Comments 0

Type and enter

George Perreas
2016-03-21 23:48:31
Report

Λακούβα σε δρόμο - ongoing

Tags: dromos , chalandri , odiki-kikloforia , lakouva

Δημιουργήθηκε λόγω βροχής

Like 1
Dislike 0
Show All Comments 1

Type and enter

George Perreas
2016-03-20 14:43:42
Report

Vromikes paralties stin Salamina - opened

Tags: oikologia , salamina , paralties

Web Interface: User profile page

Αντίστοιχα στο προφίλ ενός οργανισμού μπορούμε να δούμε το όνομα του οργανισμού, την διεύθυνση του στο χάρτη και κάτω από το όνομα του, τις περιοχές όπου στηρίζει, μία μικρή περιγραφή και συγκεντρωτικά στοιχεία με την συνεισφορά του και τα συνολικά δημοσιοποιημένα δημοψηφίσματα. Τα δημοψηφίσματα παρουσιάζονται και σε μία λίστα κάτω από τις πληροφορίες του οργανισμού όπου ο κάθε χρήστης μπορεί να δράσει με τον ίδιο τρόπο όπως στην σελίδα των δημοψηφισμάτων.

Home / Profile

Chalandri council
Address: Greece, Chalandri
152 34

2
Total Participated Events

1
Total Published Petitions

Supported Locations: Χαλανδρίου

About us..
This is an organization ...

Petitions

Chalandri council
Published: 2016-03-16 20:33:59
Ends by: 2016-03-31
Μεταφορά της χωματερής - started
Tags: test-org-1 , test , petition

Report

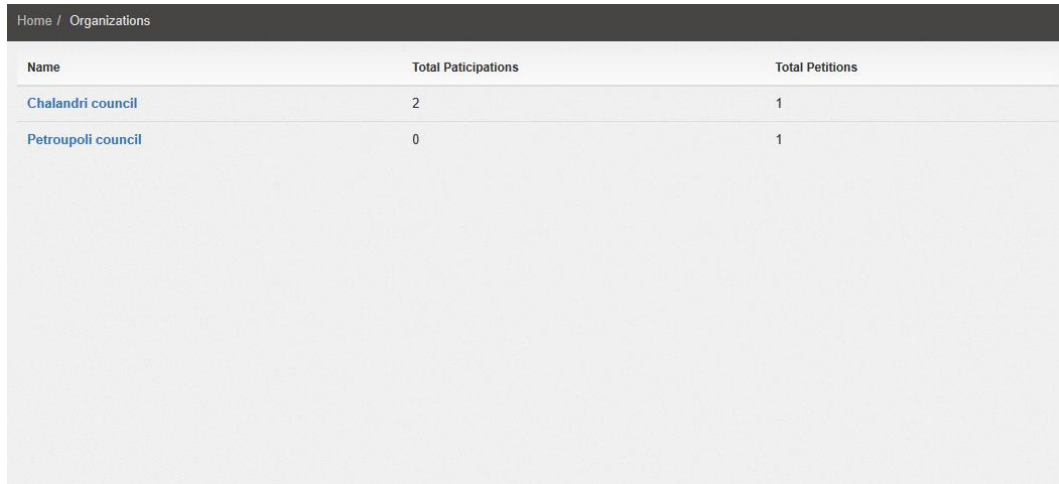
Sign Total signatures: 2

Like 2 Dislike 1 Show All Comments 18

Type and enter

Web Interface: Organization profile page

Τέλος στην διεπαφή της παρακάτω εικόνας παρουσιάζεται μία συνοπτική λίστα όλων των εγγεγραμμένων οργανισμών δείχνοντας την συνολική συνεισφορά που έχουν κάνει στα προβλήματα της περιοχής τους καθώς και τα συνολικά δημοψηφίσματα που έχουν δημοσιοποιήσει.



Name	Total Participations	Total Petitions
Chalandri council	2	1
Petrupoli council	0	1

Web Interface: Organization list page

6 Συμπεράσματα

Η εφαρμογή Smart City Project αποτελείται από κάποια βασικά χαρακτηριστικά όπως υποστήριξη προβλημάτων, δημοψηφισμάτων, χρηστών και οργανισμών. Αυτά τα χαρακτηριστικά αναδείχθηκαν μέσω της ιστοσελίδας που σχεδιάστηκε και αναπτύχθηκε. Στόχος είναι να επεκταθεί προσθέτοντας κάποιες επιπλέον δυνατότητες όπως ανάπτυξη περιβάλλοντος συνεργατικότητας μεταξύ των οργανισμών με στόχο την καλύτερη επίλυση των προβλημάτων, εργαλείων για εξαγωγή συμπερασμάτων σχετικά με τις αναφορές των περιεχομένων που δίνονται από τους χρήστες, δυναμική προσαρμογή της προσωπικής σελίδας του οργανισμού σε συνδυασμό με άλλες πηγές πληροφορίας σχετικά με αυτόν καθώς και περαιτέρω ανάπτυξη της ιστοσελίδας και της mobile (android) εφαρμογής

7 Βιβλιογραφία

1. *E-Government and Civic Engagement: How is Citizens' Use of City Web Sites Related with Civic Involvement and Political Behaviors?* **Kang, Seok και Gearhart, Sherice**. s.l. : Journal of Broadcasting & Electronic Media, 2010.
2. *Communication, Context, and Community An Exploration of Print, Broadcast and Internet Influences*. **SHAH, DHAVAN V., MCLEOD, JACK M. και YOON, SO-HYANG**. s.l. : Sage Publications, 2001.
3. *Continuous performance improvement: a case study of ICT in UK local government* . **Worrall, Les, Greenwood, Martin και Madgwick, Terry**. UK : Occasional Paper Series, 2004.
4. **Hall, Bronwyn και Khan, Beethika**. *Adoption of New Technology*. Department of Economics, Institute for Business and Economic Research, UC Berkeley. 2003. Department of Economics, Working Paper Series.
5. *Smart Cities: Definitions, Dimensions, Performance, and Initiatives*. **Albino, Vito και Dangelico, Rosa Maria**. 2015, Journal of Urban Technology, σ. 20.
6. *Investigating social factors of sustainability in a smart city*. **Monfaredzadeh, Tannaz και Krueger, Robert**. s.l. : Elsevier, 2015.
7. *G2C and C2G: Emerging principles and architectures in e-government and e-participation*. **Pappa, D. και Stergioulas, L.K.** London : s.n., 2006.
8. FixMyStreet. [Ηλεκτρονικό] <http://fixmystreet.org/>.
9. Novoville. [Ηλεκτρονικό] <http://www.novoville.com>.
10. Improve My City. [Ηλεκτρονικό] <http://www.improve-my-city.com/>.
11. Geodata Gov. [Ηλεκτρονικό] <http://geodata.gov.gr/>.
12. GeoJSON Specification. [Ηλεκτρονικό] <http://geojson.org/geojson-spec.html#introduction>.
13. Role-based access control. [Ηλεκτρονικό] https://en.wikipedia.org/wiki/Role-based_access_control.
14. KML Tutorial. [Ηλεκτρονικό] https://developers.google.com/kml/documentation/kml_tut.
15. E-Petition Signature. [Ηλεκτρονικό] <https://ec.europa.eu/digital-single-market/node/6806>.
16. About MongoDB. [Ηλεκτρονικό] <https://www.mongodb.com/index>.

