

Πτυχιακή εργασία

**Διαδικτυακή πλατφόρμα καταγραφής συζητήσεων μέσω
παιχνιδοποίησης για την ανίχνευση καταχρηστικής
συμπεριφοράς στο διαδίκτυο**

Εμμανουήλ Σμυρνάκης

A.M. 3504



Επιβλέπων καθηγητής: Γιάννης Τζιτζικας

Συν επιβλέπων post-doc: Παναγιώτης Παπαδάκος

Τμήμα Επιστήμης Υπολογιστών

Πανεπιστήμιο Κρήτης

Ηράκλειο, Σεπτέμβριος 2020

Περιεχόμενα

Εισαγωγή	3
Βασικές έννοιες και ορισμοί	4
Περιγραφή Πλατφόρμας	5
Χρήστης(User)	5
Παιχνίδι(Game).....	6
Μήνυμα(Message).....	7
Ετικέτα(Tag)	7
Εργασία(Task)	7
Αποστολή(Challenge)	8
Σχολιασμός(Annotation)	9
Κριτική(Rating).....	9
Χαρακτηριστικά Πλατφόρμας	10
Βάση Δεδομένων	10
Front-end	13
Back-end.....	13
REST API.....	14
Εισαγωγή REST API	14
Πόροι(Resources) REST API	15
Οδηγίες χρήσης REST API	16
User	16
Game	17
Message	19
Tag	20
Task.....	21
Challenge.....	22
Annotation	23
Rating	24
Σημειώσεις.....	25
Πηγές.....	26

Εισαγωγή

Το διαδίκτυο στις μέρες μας είναι ένα αναπόσπαστο κομμάτι της καθημερινότητας μας, είτε στην δουλεία μας ή στον ελεύθερο μας χρόνο. Βεβαίως ,όπως παρατηρούμε στην καθημερινή ζωή παραβατικές και καταχρηστικές συμπεριφορές έτσι και αυτές δεν μπορεί να λείπουν από το διαδίκτυο. Η συγκεκριμένη πτυχιακή εργασία έχει ως σκοπό την ανίχνευση και καταγραφή παραβατικών συμπεριφορών στο διαδίκτυο, ώστε να αποφευχθούν παρόμοιες μελλοντικές συμπεριφορές. Με βάση αυτό, προχώρησα στην δημιουργία μίας διαδικτυακής πλατφόρμας ανταλλαγής μηνυμάτων, η οποία χρησιμοποιεί την μέθοδο της παιχνιδοποίησης δίνοντας στον χρήστη μέσα από την δημιουργική ενασχόληση ενός παιχνιδιού την ευκαιρία να προσφέρει χρήσιμα δεδομένα προς επεξεργασία.

Βασικές έννοιες και ορισμοί

Στην παρούσα εργασία χρησιμοποιούνται οι όροι crowdsourcing και gamification για την σχεδίαση και υλοποίηση της.

Gamification (Παιχνιδοποίηση) αποτελεί την ενσωμάτωση μηχανισμών παιχνιδιού σε ένα περιβάλλον το οποίο δεν αποτελεί μέρος παιχνιδιού. Αναγνωρίζεται ως μια τακτική που αξιοποιεί τα χαρακτηριστικά που κάνουν ένα παιχνίδι διασκεδαστικό ώστε να ενισχύσει την ενασχόληση του χρήστη με το περιβάλλον στο οποίο βρίσκεται.

Crowdsourcing (πληθοπορισμός) αναφέρεται σε μια μορφή διαδικτυακής δραστηριότητας στην οποία μία εταιρία ή ένας οργανισμός προτείνει σε μια ομάδα ατόμων με ποικίλες γνώσεις να αναλάβουν μια εργασία είτε εθελοντικά είτε επί πληρωμή. Η ανάληψη της εργασίας, η οποία ποικίλλει σε πολυπλοκότητα και στο βαθμό στον οποίο είναι χωρισμένη και στην οποία το πλήθος πρέπει να συμμετάσχει με προσωπική εργασία, χρήματα, γνώση, εμπειρία, περιλαμβάνει πάντοτε αμοιβαίο όφελος και για τις δύο πλευρές.

Συνεπώς, χρησιμοποιώντας σαν βάση αυτούς τους 2 ορισμούς υλοποιούμε μια Crowdsourcing διαδικτυακή πλατφόρμα η οποία χρησιμοποιεί την μέθοδο του Gamification ώστε να δώσει κίνητρα στους χρήστες που την χρησιμοποιούν να ολοκληρώσουν την δουλεία που τους ανατίθεται.


Περιγραφή Πλατφόρμας

Η διαδικτυακή εφαρμογή-παιχνίδι περιέχει τα εξής:

- Χρήστες
- Παιχνίδια
- Εργασίες
- Μηνύματα
- Ετικέτες
- Αποστολές
- Σχολιασμούς
- Κριτικές

Χρήστης(User)

Κάθε χρήστης που θέλει να χρησιμοποιήσει την εφαρμογή πρέπει ,αν δεν έχει εγγραφθεί προηγουμένως, πρέπει να εγγραφθεί ώστε να συνδεθεί. Έχει την δυνατότητα να συνδεθεί είτε ως παίχτης, ή ως δημιουργός εργασίας (Task Administrator) ή ακόμα και ως σχολιαστής(Annotator). Αν συνδεθεί ως χρήστης-παίχτης ,έχει ένα προφίλ όπου αναγράφονται οι πόντοι του, τα παιχνίδια που έχει παίξει και τα μετάλλια του καθώς και ένα ιστορικό των παιχνιδιών που έχει πάρει μέρος .Αντίστοιχα έχει ένα προφίλ ως δημιουργός εργασιών και άλλο ένα ως σχολιαστής.



Edit

ManosManos


Games

22

Points

305

Badges

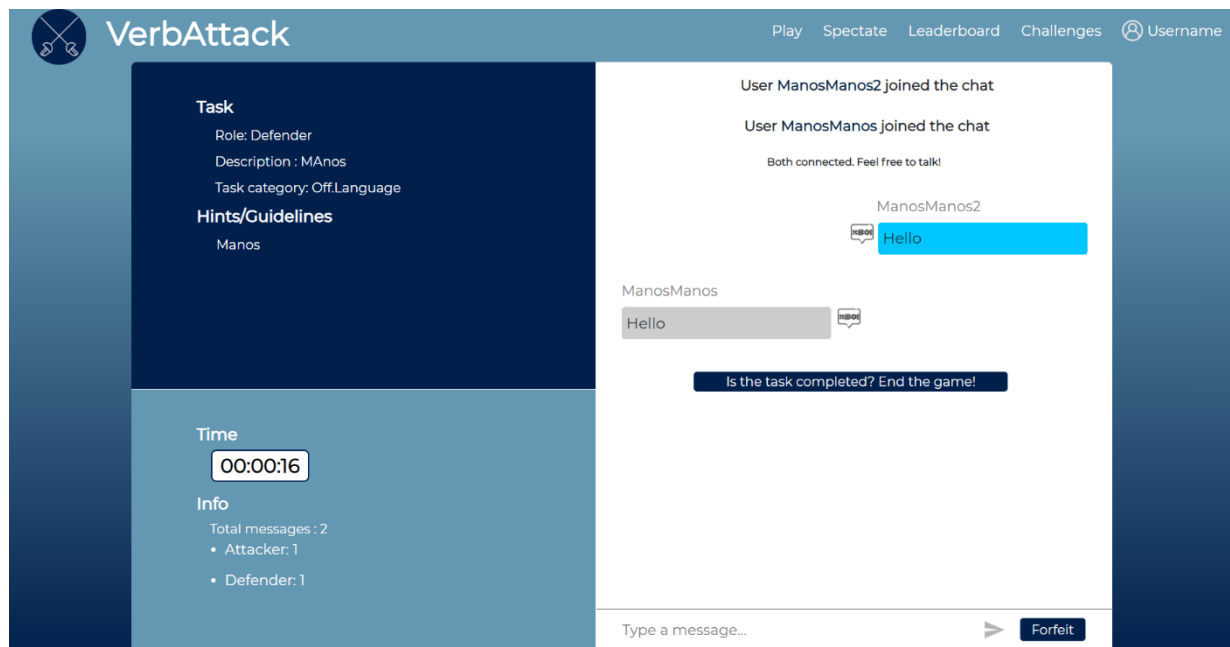


Match history

ManosManos 0	ManosManos2 10	Tags 2	Messages 4	Time 00:01:17	Category Phising	Date 9/23/2020
ManosManos2 20	ManosManos 0	Tags 0	Messages 4	Time 00:00:24	Category Phising	Date 9/23/2020
ManosManos 10	ManosManos2 30	Tags 4	Messages 2	Time 00:00:34	Category Phising	Date 9/23/2020
ManosManos 22	ManosManos2 42	Tags 2	Messages 4	Time 00:00:41	Category Phising	Date 9/23/2020
ManosManos 37	ManosManos2 57	Tags 0	Messages 2	Time 00:00:12	Category Phising	Date 9/23/2020
ManosManos 47	Maria 0	Tags 2	Messages 4	Time 00:04:46	Category Phising	Date 9/23/2020

Παιχνίδι(Game)

Κάθε παιχνίδι είναι μια συνομιλία μεταξύ 2 χρηστών. Σε ένα παιχνίδι υπάρχουν 2 ρόλοι: Ένας επιτιθέμενος και ένας αμυνόμενος. Επίσης κάθε παιχνίδι έχει μια κατηγορία και μία εργασία στην οποία υπάγεται. Σε ένα παιχνίδι οι 2 χρήστες συνομιλούν προσπαθώντας από την μεριά του ο καθένας να πετύχουν το σκοπό τους και το παιχνίδι τελειώνει είτε με την σύμφωνη γνώμη και των 2 ή αν ένας από τους 2 παραδοθεί. Κάθε παιχνίδι και ανάλογα με την απόδοση του χρήστη, του προσφέρει πόντους οι οποίοι προσμετρούνται στην βαθμολογία του. Για κάθε ετικέτα σε μήνυμα που καταχωρεί ο χρήστης λαμβάνει 2 βαθμούς. Αν και ο αντίπαλος έχει καταχωρήσει του ίδιου τύπου ετικέτα και οι 2 λαμβάνουν από 3 έξτρα βαθμούς. Αν και οι 2 συμφωνήσουν στην λήξη του παιχνιδιού από κοινού λαμβάνουν ο καθένας από 10 βαθμούς. Ενώ σε ενδεχόμενη παραίτηση ο χρήστης δεν λαμβάνει τους 10 βαθμούς.



Μήνυμα(Message)

Μήνυμα μπορεί να στείλει οποιοσδήποτε χρήστης έχει συνδεθεί και παίρνει μέρος σε ένα παιχνίδι. Ένα μήνυμα αποτελείται από τον αποστολέα (sender) όπου εμφανίζεται από πάνω, το περιεχόμενο (context) μέσα στο πλαίσιο, καθώς και ένα κουμπί για την επιλογή και προσθήκη ετικέτας (tag) στο μήνυμα.

ManosManos

Hello



Ετικέτα(Tag)

Σε κάθε μήνυμα είναι δυνατό ένας χρήστης να εισάγει μία ετικέτα η οποία δηλώνει κάποια καταχρηστική συμπεριφορά και αποτελεί μέρος του παιχνιδιού. Το είδος της ετικέτας εμφανίζεται από κάτω από το περιεχόμενο του μηνύματος καθώς και ένα icon 'μολύβι' ώστε να έχει την δυνατότητα ο χρήστης να κάνει επεξεργασία της ετικέτας.

ManosManos


Hello



#Phising,

Εργασία(Task)

Η εργασία αποτελεί βασικό κομμάτι της εφαρμογής καθώς όλα τα παιχνίδια που διεξάγονται έχουν ως βάση μια συγκεκριμένη εργασία(Task). Κάθε εργασία έχει χαρακτηριστικά που τα ορίζει ο δημιουργός της και υπάγεται πάντα σε μία κατηγορία. Κάθε εργασία (Task) πρέπει να χαρακτηρίζεται από ένα όνομα(Task Name), μία περιγραφή-σενάριο (Task Description) και μια κατηγορία(Category). Επιπροσθέτως, κάθε task πρέπει να δίνει πληροφορίες στους χρήστες για τον σκοπό του παιχνιδιού και κάποιες βοήθειες (hints).

 VerbAttack

Create a TaskTasksUsername

Task b4eae24

Task Name

PIN acquisition

Task Description

Attacker acquires the defenders card PIN

Attacker's Description/Scenario

As attacker you must obtain defenders card PIN using various verbal methods

Defender's Description/Scenario

As defender you must be very careful with attackers demands

Max Messages

5

Category

Phising

Attacker's Hints/Guidelines

Attacker should try to deceive defender so you should be really crafty

Defender's Hints/Guidelines

PIN acquisition

Number of players


2-players game

Created

9/23/2020, 8:17:21 PM

Αποστολή(Challenge)

Κάθε χρήστης από την στιγμή της δημιουργίας του προφίλ του έχει μία λίστα με αποστολές που πρέπει να φέρει εις πέρας έτσι ώστε να κερδίσει πόντους ή μετάλλια ανεβαίνοντας στην κατάταξη παιχτών. Αν ολοκληρωθεί μία αποστολή εμφανίζεται μήνυμα και καταχωρούνται οι πόντοι σαν ανταμοιβή στον χρήστη.

 VerbAttack

PlaySpectateLeaderboardChallengesUsername

Challenges

Daily challenges

Play 5 games

40%

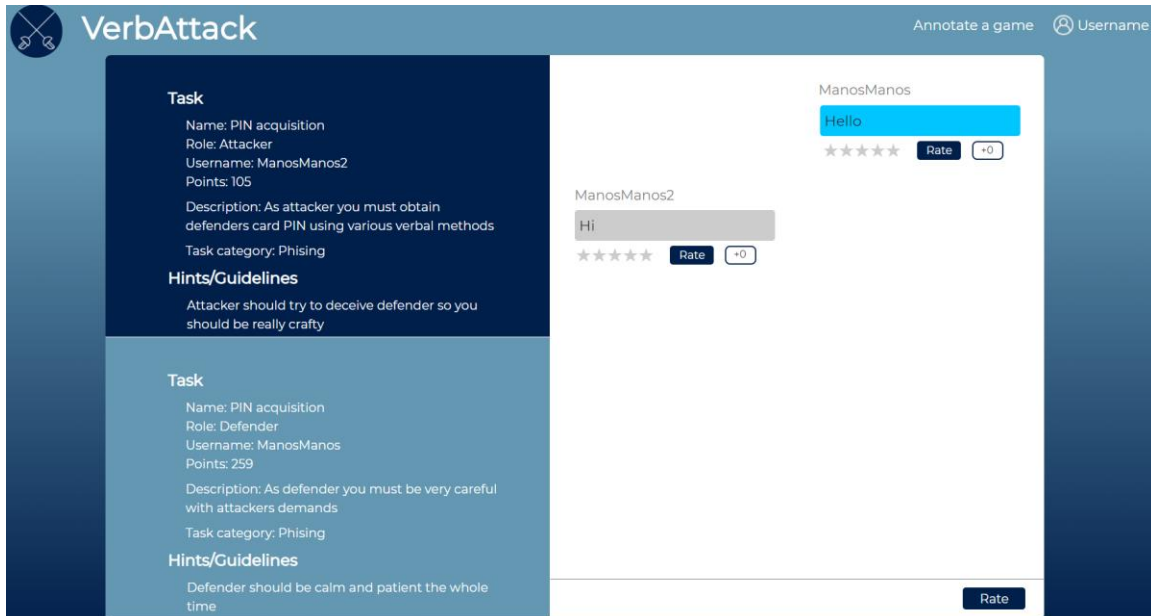
Reward

100

General challenges

Σχολιασμός(Annotation)

Κάθε χρήστης εκτός από την συμμετοχή του στα παιχνίδια και στην δημιουργία εργασιών έχει την δυνατότητα συνδεδεμένος ως σχολιαστής να κρίνει παιχνίδια που παρακολουθεί.



Κριτική(Rating)

Η κριτική αποτελεί μέρος του σχολιασμού ενός παιχνιδιού από έναν χρήστη-σχολιαστή και αναλόγως το πόσο θετική είναι προσφέρει πόντους στους χρήστες που παίρνουν μέρος στο παιχνίδι. Για κάθε αστέρι που προσθέτει ο σχολιαστής λαμβάνει ο αποστολέας του μηνύματος ένα πόντο. Επιπρόσθετα, ο σχολιαστής βαθμολογεί τον κάθε παίκτη ξεχωριστά και το συνολικό παιχνίδι με τον ίδιο τρόπο. Τέλος, ο σχολιαστής για κάθε κριτική, λαμβάνει 2 πόντους και 10 έξτρα για το παιχνίδι.

ManosManos2



Χαρακτηριστικά Πλατφόρμας

Βάση Δεδομένων

Η συγκεκριμένη πλατφόρμα διαθέτει μια βάση δεδομένων που υλοποιήθηκε στο δωρεάν σύστημα βάσεων δεδομένων του PostgreSQL.

Αποτελείται από τους εξής πίνακες:

Users: Αποτελεί τον πίνακα που αποθηκεύει τους χρήστες της πλατφόρμας και τα στοιχεία τους.

Όνομα πεδίου	Τύπος	Περιγραφή
User_id	Varchar	Primary key
Username	Varchar	Όνομα χρήστη
Password	Varchar	Κωδικός χρήστη
Points	Integer	Πόντοι χρήστη
Registered_since	Varchar	Ημερομηνία εγγραφής
Games_played	Integer	Παιχνίδια χρήστη

Games: Αποτελεί τον πίνακα που αποθηκεύει τα παιχνίδια της πλατφόρμας και τα στοιχεία του κάθε παιχνιδιού.

Όνομα πεδίου	Τύπος	Περιγραφή
Game_id	Varchar	Primary key
Starting_time	Timestamp	Χρόνος έναρξης παιχνιδιού
Category	Varchar	Κατηγορία παιχνιδιού
Task_id	Varchar	Αναγνωριστικό του task
Rounds	Integer	Μηνύματα παιχνιδιού
Attacker_username	Varchar	Όνομα χρήστη επιτιθέμενου
Defender_username	Varchar	Όνομα χρήστη αμυνόμενου
Attacker_points	Integer	Πόντοι επιτιθέμενου
Defender_points	Integer	Πόντοι αμυνόμενου
Finished	Boolean	Ένδειξη τερματισμού παιχνιδιού
Completion_time	Integer	Χρόνος παιχνιδιού

Tasks: Αποτελεί τον πίνακα που αποθηκεύει τα tasks της πλατφόρμας και τα χαρακτηριστικά του κάθε task.

Όνομα πεδίου	Τύπος	Περιγραφή
Task_id	Varchar	Primary key
Taskname	Varchar	Όνομα task
Description	Varchar	Περιγραφή task
Category	Varchar	Κατηγορία task
Attacker_hints	Varchar	Συμβουλές επιτιθέμενου
Defender_hints	Varchar	Συμβουλές αμυνόμενου
Attackers_description	Varchar	Περιγραφή εργασίας επιτιθέμενου
Defenders_description	Varchar	Περιγραφή εργασίας αμυνόμενου
Max_messages	Integer	Μέγιστος αριθμός μηνυμάτων
Timestamp	Timestamp	Ημερομηνία και ώρα έναρξης task
Task_creator	Varchar	Δημιουργός task
Active	Boolean	Ένδειξη ενεργού task

Challenges: Αποτελεί τον πίνακα με τις ημερήσιες και μακροχρόνιες αποστολές του χρήστη.

Όνομα πεδίου	Τύπος	Περιγραφή
Challenge_id	Varchar	Primary key
Progress	Numeric	Πρόοδος αποστολής χρήστη
Description	Varchar	Περιγραφή αποστολής χρήστη
Category	Varchar	Κατηγορία αποστολής χρήστη
Max_progress	Numeric	Μέγιστο πρόοδου αποστολής
Reward_points	Numeric	Ανταμοιβή αποστολής
User_id	Varchar	Αναγνωριστικό χρήστη
Daily	Boolean	Ένδειξη ημερήσιας αποστολής

Messages: Αποτελεί τον πίνακα με τα μηνύματα ενός παιχνιδιού που έχουν ανταλλάξει οι 2 χρήστες.

Όνομα πεδίου	Τύπος	Περιγραφή
Message_id	Varchar	Primary key
Game_id	Varchar	Αναγνωριστικό παιχνιδιού
User_id	Varchar	Αναγνωριστικό αποστολέα
Context	Varchar	Περιεχόμενο μηνύματος
Timestamp	Timestamp	Ημερομηνία και χρόνος αποστολής μηνύματος

Tags: Αποτελεί τον πίνακα που αποθηκεύει τις ετικέτες που βάζουν οι χρήστες στα μηνύματα της πλατφόρμας.

Όνομα πεδίου	Τύπος	Περιγραφή
Tag_id	Varchar	Primary key
User_id	Varchar	Αναγνωριστικό χρήστη
Message_id	Varchar	Αναγνωριστικό μηνύματος
Category	Varchar	Κατηγορία ετικέτας
Timestamp	Timestamp	Ημερομηνία και ώρα καταχώρησης ετικέτας
Game_id	Varchar	Αναγνωριστικό παιχνιδιού

Annotations: Αποτελεί τον πίνακα που αποθηκεύει τους σχολιασμούς που κάνουν οι χρήστες στα μηνύματα και στις συνομιλίες γενικά στην πλατφόρμα.

Όνομα πεδίου	Τύπος	Περιγραφή
Annotation_id	Varchar	Primary key
Game_id	Varchar	Αναγνωριστικό παιχνιδιού
User_id	Varchar	Αναγνωριστικό σχολιαστή
Created	Timestamp	Ημερομηνία και ώρα σχολιασμού

Ratings: Αποτελεί τον πίνακα που αποθηκεύει τις κριτικές που κάνουν οι χρήστες στα μηνύματα και στις συνομιλίες στην πλατφόρμα.

Όνομα πεδίου	Τύπος	Περιγραφή
Rating_id	Varchar	Primary key
Annotation_id	Varchar	Αναγνωριστικό σχολιασμού
Stars	Integer	Βαθμός κριτικής σχολιασμού
Message_id	Varchar	Αναγνωριστικό μηνύματος
Timestamp	Timestamp	Ημερομηνία και χρόνος καταχώρησης κριτικής

Front-end

Για την σχεδίαση και την υλοποίηση του Front-end έγινε εκτενής χρήση της React που αποτελεί μια βιβλιοθήκη της Javascript για την σχεδίαση διεπαφών χρηστών(user interfaces).

Back-end

Αντίστοιχα, για την σχεδίαση και υλοποίηση του Back-end χρησιμοποιήθηκε μια ελαφριά και απλή διαδικτυακή δομή(web framework) της Java, η Javalin η οποία λαμβάνει και παρέχει δεδομένα JSON.

REST API

Εισαγωγή REST API

Προκειμένου να καλύψουμε τις ανάγκες επικοινωνίας του client με τον server και αντιστοίχως με την βάση δεδομένων χρησιμοποιήθηκε ένα API σύμφωνα με την αρχιτεκτονική REST κατά την οποία χρησιμοποιούνται μέθοδοι του πρωτοκόλλου HTTP. Χαρακτηριστικά, η μέθοδος GET χρησιμοποιείται για ανάκτηση αντικειμένων, η μέθοδος POST για την δημιουργία αντικειμένων, η μέθοδος PUT για τροποποίηση αντικειμένων και η μέθοδος DELETE για διαγραφή αντικειμένων.

Επιπρόσθετα, σημαντική έννοια στην αρχιτεκτονική REST είναι οι πόροι (Resources) οι οποίοι περιγράφουν τα αντικείμενα τα οποία η εφαρμογή καλείται να διαχειριστεί. Οι πόροι είναι είτε διαθέσιμοι μέσω URL καθώς ένα URL υλοποιείται με σκοπό να δίνει πρόσβαση σε κάποιο συγκεκριμένο αντικείμενο ή σε μια συλλογή από αυτά, ή αντίστοιχα όταν υπάρχει μία βάση δεδομένων από τους εκάστοτε πίνακες που αντιστοιχούν.

Πόροι(Resources) REST API

- **/user**
Αντιστοιχεί στον πίνακα **users** της βάσης δεδομένων που περιγράφει την έννοια ενός χρήστη.
- **/games**
Αντιστοιχεί στον πίνακα **games** της βάσης δεδομένων που περιγράφει την έννοια ενός παιχνιδιού.
- **/tasks**
Αντιστοιχεί στον πίνακα **tasks** της βάσης δεδομένων που περιγράφει την έννοια μιας εργασίας.
- **/messages**
Αντιστοιχεί στον πίνακα **messages** της βάσης δεδομένων που περιγράφει την έννοια ενός μηνύματος.
- **/tags**
Αντιστοιχεί στον πίνακα **tags** της βάσης δεδομένων που περιγράφει την έννοια μίας ετικέτας.
- **/challenges**
Αντιστοιχεί στον πίνακα **challenges** της βάσης δεδομένων που περιγράφει την έννοια μίας αποστολής.
- **/annotations**
Αντιστοιχεί στον πίνακα **annotations** της βάσης δεδομένων που περιγράφει την έννοια ενός σχολιασμού.
- **/ratings**
Αντιστοιχεί στον πίνακα **ratings** της βάσης δεδομένων που περιγράφει την έννοια μίας κριτικής.

Οδηγίες χρήσης REST API

Παρακάτω ακολουθεί αναλυτικός οδηγός χρήσης του API και των resources του.

User

User object format:

```
{  
  "userID" : "abcd",  
  "username" : "ManosManos",  
  "password" : "manos123",  
  "points" : "10",  
  "date" : "2020-08-04 10:45:57.327",  
  "gamesPlayed" : "1",  
}
```

Methods	HTTP Request	HTTP Response	Description
addUser()	POST /users +<params>	201 created URI: /users/{id}	Δημιουργία καινούργιου χρήστη με μοναδικό {id} Τα <params> είναι ένα JSON με τα στοιχεία του χρήστη που δημιουργούμε.
updateUser()	PUT /users/{id} + <params>	200 OK	Τροποποίηση στοιχείων χρήστη με μοναδικό {id}. Τα <params> είναι τα καινούργια στοιχεία του χρήστη.
getUser()	GET /users/{id}	200 OK	Λήψη ενός JSON με τα στοιχεία χρήστη με μοναδικό {id}
deleteUser()	DELETE /users/{id}	200 OK	Διαγραφή χρήστη με μοναδικό {id}
getUsers()	GET /users	200 OK	Λήψη ενός JSON με τα στοιχεία όλων των χρηστών.
getTopUsers()	GET /leaderboard	200 OK	Λήψη ενός JSON με τα στοιχεία όλων των χρηστών ταξινομημένα σε φθίνουσα σειρά ως προς την βαθμολογία τους.

Game

Game object format:

```
{  
  "gameID" : "abcde",  
  "starting_time" : " 2020-08-30 16:42:01.221",  
  "category" : "Phising",  
  "task_id" : "abcdef",  
  "attackerUsername" : "ManosManos",  
  "defenderUsername" : "Manos123",  
  "attackerPoints" : "100",  
  "defendePoints " : "100",  
  "ongoing" : "true",  
  "finished" : "false",  
  "completion_time" : " 100",  
}
```

Methods	HTTP Request	HTTP Response	Description
addGame()	POST /games +<params>	201 created URI: /games/{id}	Δημιουργία καινούργιου παιχνιδιού με μοναδικό {id} Τα <params> είναι ένα JSON με τα στοιχεία του παιχνιδιού που δημιουργήθηκε.
editGame()	PATCH /games/{id} + <params>	200 OK	Τροποποίηση στοιχείων παιχνιδιού με μοναδικό {id}. Τα <params> είναι τα καινούργια στοιχεία του παιχνιδιού.
getGame()	GET /games/{id}	200 OK	Λήψη ενός JSON με τα στοιχεία του παιχνιδιού με μοναδικό {id}
getAllGames ()	GET /games	200 OK	Λήψη ενός πίνακα με όλα τα παιχνίδια
getLiveGames()	GET / games/ tv	200 OK	Λήψη ενός πίνακα με όλα τα live παιχνίδια
getLiveGame()	GET /games/tv/{id}	200 OK	Λήψη ενός JSON με τα στοιχεία ενός live παιχνιδιού με μοναδικό {id}
joinGame()	PUT /games/{id} + <params>	200 OK	Είσοδος χρήστη σε παιχνίδι που δημιουργήθηκε από τον αντίπαλο με μοναδικό {id}
getGameTags()	GET /games/{id} /tags	200 OK	Λήψη ενός JSON πίνακα με τα tags στο παιχνίδι με μοναδικό {id}

Message

Message object format:

```
{  
  "messageID" : " 9c9b5bc",  
  "gameID" : " 0c9db02",  
  "context" : " Hello",  
  "timestamp" : " 2020-08-09 12:49:31.908",  
  "userID" : " 0dd7d4d",  
}
```

Methods	HTTP Request	HTTP Response	Description
addMessage()	POST /games/{game_id}/messages +<params>	201 created URI: /games/{game_id} /messages/{id}	Δημιουργία καινούργιου μηνύματος με μοναδικό {id} στο παιχνίδι με {game_id}. Τα <params> είναι ένα JSON με τα στοιχεία του μηνύματος που δημιουργούμε.
getMessage()	GET/games/{game_id}/messages/{id}	200 OK	Λήψη ενός JSON με τα στοιχεία του μηνύματος με μοναδικό {id} στο παιχνίδι με {game_id}
getAllMessages()	GET/games/{game_id}/messages	200 OK	Λήψη ενός πίνακα JSON με τα στοιχεία όλων των μηνυμάτων στο παιχνίδι με {game_id}

Tag

Tag object format:

```
{  
  "tagID" : " 22e44ba",  
  "gameID" : " 0c9db02",  
  "messageID" : " ae8d3c6",  
  "timestamp" : " 2020-08-09 12:49:31.908",  
  "userID" : " Odd7d4d",  
  "category" : " Racism",  
}
```

Methods	HTTP Request	HTTP Response	Description
addTag()	POST /games/{game_id}/messages/ {messages_id}/tags/ +<params>	201 created URI: /games/{game_id}/ messages/{messages_id}/ tags/{id}	Δημιουργία καινούργιου tag σε ένα μήνυμα με {message_id} σε παιχνίδι με {game_id} Τα <params> είναι ένα JSON με τα στοιχεία του tag που προσθέτουμε.
updateTag()	PUT /games/{game_id}/messages/ {message_id}/tags/{id} + <params>	200 OK	Τροποποίηση του tag με μοναδικό {id} που είναι πάνω σε ένα μήνυμα με {message_id} σε παιχνίδι με {game_id}. Τα <params> είναι το καινούργιο tag.
getTag()	GET /games/{game_id}/messages/ {message_id}/tags/{id}	200 OK	Λήψη ενός JSON με τα στοιχεία του tag με μοναδικό {id}
deleteTag()	DELETE /games/{game_id}/messages/ {message_id}/tags/{id}	200 OK	Διαγραφή tag με μοναδικό {id}
getAllTags()	GET /games/{game_id}/messages/ {message_id}/tags	200 OK	Λήψη ενός πίνακα JSON με τα tags στο message με μοναδικό {message_id}

Task

Task object format:

```
{  
  "taskID" : " 6d51826",  
  "taskName" : " NumberOne",  
  "category" : "Phising",  
  "description" : "Get the PIN",  
  "attackersDescription" : "Act Fast",  
  "defendersDescription" : "Act Slow",  
  "attackerHints" : "Be smart",  
  "defenderHints" : "Be serious",  
  "maxMessages" : "10",  
  "timestamp" : " 2020-09-05 12:58:07.432",  
  "taskCreator" : " b52ea08",  
  "active" : " true",  
}
```

Methods	HTTP Request	HTTP Response	Description
addTask()	POST /tasks + <params>	201 created URI: /tasks/{id}	Δημιουργία καινούργιου task με μοναδικό{id} Τα <params> είναι ένα JSON με τα στοιχεία του task που προσθέτουμε.
getTask()	GET /tasks/{id}	200 OK	Λήψη ενός JSON με τα στοιχεία του task με μοναδικό {id}
deleteTask()	DELETE /tasks/{id}	200 OK	Διαγραφή task με μοναδικό {id}
getTasks()	GET /tasks	200 OK	Λήψη ενός πίνακα JSON με όλα τα tasks
updateTask()	PUT /tasks/{id} + <params>	200 OK	Ανανέωση των στοιχείων του task με μοναδικό {id}

Challenge

Challenge object format:

```
{  
  "challengeID" : " 9cb3718",  
  "description" : "Play 5 games",  
  "progress" : "2",  
  "maxProgress" : " 5",  
  "rewardPoints" : " 100",  
  "userID" : " ae8c194",  
  "daily" : " true",  
  "category" : " games",  
}
```

Methods	HTTP Request	HTTP Response	Description
addChallenge()	POST /{uid}/challenges +<params>	201 created URI: /challenges/{id}	Δημιουργία καινούργιου challenge με μοναδικό {id} για τον χρήστη με μοναδικό {uid} Τα <params> είναι ένα JSON με τα στοιχεία του challenge που δημιουργούμε.
updateChallenge()	PUT /{uid}/challenges/{id} + <params>	200 OK	Τροποποίηση στοιχείων challenge με μοναδικό {id} για τον χρήστη με μοναδικό {uid} . Τα <params> είναι τα καινούργια στοιχεία του challenge.
getChallenge()	GET /{uid}/challenges/{id}	200 OK	Λήψη ενός JSON με τα στοιχεία του challenge με μοναδικό {id} για τον χρήστη με μοναδικό {uid}
deleteChallenge()	DELETE /{uid}/challenges/{id}	200 OK	Διαγραφή challenge με μοναδικό {id} για τον χρήστη με μοναδικό {uid}
getChallenges()	GET /{uid}/challenges	200 OK	Λήψη ενός πίνακα JSON με τα στοιχεία όλων των challenge για τον χρήστη με μοναδικό {uid}

Annotation

Annotation object format:

```
{  
  "annotationID" : " 16a1897",  
  "gameID" : " 70af9db",  
  "userID" : " b52ea08",  
  "created" : " 2020-09-10 12:06:46.355",  
}
```

Methods	HTTP Request	HTTP Response	Description
addAnnotation()	POST /annotations/{uid}/{gid} + <params>	201 created URI: /annotations/{uid}/{gid}/{aid}	Δημιουργία καινούργιου annotation με μοναδικό{aid} του user με {uid} για το game με {gid} Τα <params> είναι ένα JSON με τα στοιχεία του annotation που προσθέτουμε.
getAnnotation()	GET /annotations/{uid}/{gid}/{aid}	200 OK	Λήψη ενός JSON με τα στοιχεία του annotation με μοναδικό {aid}
getAnnotations()	GET /annotations/{uid}	200 OK	Λήψη ενός πίνακα JSON με όλα τα annotations του χρήστη με {uid}

Rating

Rating object format:

```
{  
  "annotationID" : " 16a1897",  
  "ratingID" : " f6aee8e",  
  "messageID" : " fae0385",  
  "timestamp" : " 2020-09-10 12:06:46.355",  
}
```

Methods	HTTP Request	HTTP Response	Description
addRating()	POST /annotations/{uid}/{gid}/{aid}/ratings + <params>	201 created URI: /annotations/{uid}/{gid}/{aid}/ratings /{rid}	Δημιουργία καινούργιου rating με μοναδικό {rid} στο annotation με {aid} του user με {uid} για το game με {gid} Τα <params> είναι ένα JSON με τα στοιχεία του annotation που προσθέτουμε.
getRating()	GET /annotations/{uid}/{gid}/{aid}/{rid}	200 OK	Λήψη ενός JSON με τα στοιχεία του rating με μοναδικό {rid}
getRatings()	GET /annotations/{uid}/{gid}/{aid}	200 OK	Λήψη ενός πίνακα JSON με όλα τα ratings στο annotation με {aid} } του user με {uid} για το game με {gid}

Σημειώσεις

Ένα σημαντικό θέμα που πρέπει να αντιμετωπιστεί είναι η διασφάλιση της μοναδικότητας ενός χρήστη ώστε να αποφευχθεί η προσπάθεια απόκτησης πλεονεκτήματος σε βάρος άλλων χρηστών. Είναι πολύ σημαντικό κάθε χρήστης να δρα αντικειμενικά και να μην ακολουθεί κινήσεις που θα τον ανεβάσουν στην βαθμολογία με παράνομο τρόπο.

Μια λύση σε αυτό το θέμα θα ήταν η δημιουργία λογαριασμού με την χρήση ενός μοναδικού email. Παρόλο που αποτελεί μια κλασσική αντιμετώπισης του προβλήματος δεν είναι και η πιο αποδοτική.

Μια άλλη λύση που χρησιμοποιείται ευρέως από ένα πλήθος διαδικτυακών εφαρμογών είναι η επιβεβαίωση σύνδεσης χρήστη μέσω SMS. Η λύση αυτή είναι ενδεδειγμένη αλλά μπορεί και αυτή, πιο σπάνια βέβαια, να δημιουργήσει θέματα σε χρήστες που λειτουργούν πολλαπλές συσκευές κινητής τηλεφωνίας.

Η σύνδεση μέσω της προσωπικής σελίδας του χρήστη από μια πλατφόρμα κοινωνικής δικτύωσης θα μπορούσε να λύσει το ζήτημα αυτό καθώς η πλειονότητα των χρηστών χρησιμοποιούν ένα μοναδικό προφίλ.

Τέλος, μια ακόμα λύση στο ζήτημα της μοναδικότητας ενός χρήστη στην πλατφόρμα είναι η ανίχνευση και αντιστοίχιση της IP από τον χώρο που συνδέεται ο χρήστης έτσι ώστε να αποφευχθεί η σύνδεσή του με πολλαπλά προφίλ. Βέβαια και αυτή η λύση έχει τα αρνητικά της, καθώς άνθρωποι που δουλεύουν στον ίδιο χώρο θα πρέπει να μοιράζονται ένα IP.

Πηγές

- <https://javalin.io/>
- https://blogs.oracle.com/javamagazine/javalin-a-simple-modern-web-server-framework#anchor_1
- <https://www.baeldung.com/javalin-rest-microservices>
- <https://leanpub.com/javalin/read>
- <https://reactjs.org/>
- <https://react-bootstrap.github.io/>
- <https://www.w3schools.com/>
- <https://www.postgresqltutorial.com/>