



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Σχεδιασμός, Ανάπτυξη και Εφαρμογή Προηγμένου Λογισμικού Διεπαφής χρηστών διαδικτυακών συζητήσεων (webChat), με τη χρήση της γλώσσας Java, για το σκοπό ανίχνευσης λέξεων της φυσικής γλώσσας που μαρτυρούν θυματοποίηση (Bullying). Design, Development and Implementation of Advanced User Interface Software through webChat, with the use of Java language in order to detect words of natural language indicating bullying.
Όνοματεπώνυμο Φοιτητή	Μαθαίος Σταύρου
Πατρώνυμο	Ευάγγελος
Αριθμός Μητρώου	ΜΠΠΛ 16021
Επιβλέπων	Ευθύμιος Αλέπης, Επίκουρος Καθηγητής

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Ευθύμιος Αλέπης
Επίκουρος Καθηγητής

Γεώργιος Τσιχριντζής
Καθηγητής

Μαρία Βίρβου
Καθηγητής

Περίληψη

Το αντικείμενο της παρούσας εργασίας είναι η ανάπτυξη ενός λογισμικού κοινωνικού δικτύου για την ανίχνευση του cyber bullying (διαδικτυακής θυματοποίησης) ανάμεσα στους μαθητές ενός σχολείου. Το εργαλείο αυτό είναι ένα μέσω επικοινωνίας (webChat) στον παγκόσμιο ιστό. Ο μαθητής θα μπορεί να επιλέξει το γκρούπ του μαθήματος που επιθυμεί, θα κάνει login μέσω του facebook ή του gmail και τέλος θα μπορεί να συνομιλεί με τους υπόλοιπους μαθητές του ίδιου γκρουπ. Οι συνομιλίες θα αναλύονται λέξη προς λέξη με το λεξικό των <<κακών>> λέξεων που θα έχουμε δημιουργήσει.

Μέσω του αλγορίθμου μας θα ψάχνουμε την ομοιότητα της λέξης του μαθητή με το λεξικό μας, σε περίπτωση άνω του 50% ομοιότητας της λέξης θα την αποθηκεύουμε με τα στοιχεία του σε ένα αρχείο για την ενημέρωση του διαχειριστή συστήματος ή του δασκάλου, ουτωσώστε στο τέλος ο δάσκαλος να αποφασίσει, εάν ένα παιδί υφίσταται βία και να το προστατεύσει.

Αρχικά θα δωθεί ο ορισμός του bullying και τα βασικά χαρακτηριστικά του θύματος και του θύτη, οι μορφές του σχολικού εκφοβισμού.

Θα μιλήσουμε πιο συγκεκριμένα για τον όρο της θυματοποίησης στο διαδίκτυο και για τις μορφές που έχει πάρει καθώς και τρόπους αντιμετώπισης με το εργαλείο μας.

Έπειτα θα δωθούν τα τεχνικά χαρακτηριστικά της εφαρμογής μας, ποιά θα είναι τα κριτήρια στα οποία θα βάζουμε τους χρήστες στην μαύρη λίστα μας. Απο το σημείο που θα κάνει login στην εφαρμογή, μέχρι το σημείο που θα εξέλθει από αυτήν.

Έπειτα θα γίνει ανάλυση των τεχνικών μεθόδων, της γλώσσας προγραμματισμού και του αλγορίθμου που θα χρησιμοποιήσουμε, για την ανάλυση των δεδομένων μας.

Τέλος θα παρουσιαστεί μία σύνοψη του έργου, τα συμπεράσματα τα οποία προέκυψαν, καθώς και προτάσεις για μελλοντικές βελτιώσεις προσθήκες, σε ένα πλαίσιο εμπειριστατωμένης και ενδεδεχούς επιτόπιας μελλοντικής έρευνας σε σχολικό περιβάλλον.

Λέξεις-κλειδιά: διεπαφή χρήστη, webChat, θυματοποίηση, διαδικτυακές εφαρμογές, java web application, διαδικτυακός server

Abstract

The subject of this work is the development of a software interface system for the detection of bullying. This tool is a webChat on the web to user groups. The user will be able to choose the desired group, then login and finally be able to chat with the group's users. The conversations will be recorded, word-for-word with the dictionary of << bad >> words we will have created and finally stored in a file.

Firstly, the definition of bullying and its basic features will be given, then a dictionary of "bad" words will be created, based on the bibliography. If the word used by the user matches this word of the dictionary we have created, then I will store it in a file.

Secondly, we will be given the technical characteristics of our application, what will be the criteria in which we will put users on our blacklist. From the point of login to the application, to the point where it will exit. We will analyze the techniques of the methods, the programming language and the algorithm that we will use to analyze our data.

Finally, a summary of the project will be presented, the conclusions that have been drawn up, as well as proposals for future improvements, in the framework of a thorough and thorough future research.

Keywords: user interface, webChat, bullying, web applications, Java Web Applications, Web Server

Περιεχομενα

Πρόγραμμα Μεταπτυχιακών Σπουδών.....	1
Μεταπτυχιακή Διατριβή.....	1
Περίληψη.....	3
Abstract.....	3
1. Εισαγωγή.....	7
1.1. Σκοπός της εργασίας.....	7
2. Εκφοβισμός στο σχολικό περιβάλλον.....	8
2.1. Ορισμός εκφοβισμού (bullying).....	8
2.2. Μορφές του σχολικού εκφοβισμού.....	8
2.2.1 Τα χαρακτηριστικά των θυμάτων.....	8
2.2.2 Τα χαρακτηριστικά των Νταήδων.....	9
2.3. Οι μηχανισμοί της ομάδας.....	10
2.4. Τρόποι αντιμετώπισης της θυματοποίησης.....	10
3. Διαδικτυακός εκφοβισμός (Cyber bullying).....	11
3.1. Ορισμός διαδικτυακού εκφοβισμού.....	12
3.2. Αιτίες διαδικτυακού εκφοβισμού.....	13
3.3. Μορφές διαδικτυακού εκφοβισμού.....	13
3.4. Τρόποι αντιμετώπισης διαδικτυακού εκφοβισμού.....	13
3.5. Ακατάλληλες διαδικτυακές ιστοσελίδες.....	14
3.6. Διαδικτυακή αποπλάνηση ανηλίκου (grooming).....	14
3.7. Προστασία από διαδικτυακές ιστοσελίδες.....	15
4. Τεχνικά χαρακτηριστικά της διαδικτυακής εφαρμογής μας.....	15
4.1. Η ιστορία της Java.....	15
4.2. Τα χαρακτηριστικά της Java και η εικονική της μηχανή (Virtual Machine).....	17
4.3. Java Platform, Enterprise Edition (Java EE).....	18
4.4. Java EE Dependency Injection.....	19
4.5. Java EE Containers.....	20
4.5.1. Java EE Container – Υπηρεσίες.....	20
4.5.2. Java EE – Τύποι container.....	21
4.5.3. Web Υπηρεσίες (web services).....	22
4.6. XML (Extensible Markup Language).....	22
4.7. Βασική ορολογία της XML.....	23
4.8. WebSocket.....	24
4.9. MAVEN.....	26
4.10. Firebase.....	27
4.10.1. Η ιστορία της Firebase.....	27

4.10.2. Υπηρεσίες της Firebase.....	28
4.11. Git (λογισμικό).....	32
4.11.1 Η ιστορία του Git.....	32
4.11.2. Τα Χαρακτηριστικά του Git.....	33
4.11.3. Δομές δεδομένων του Git.....	37
4.12. Spring Framework.....	38
4.12.1. Spring MVC.....	39
4.12.2. Ο DispatcherServlet.....	39
4.13. HTML.....	41
4.14. CSS.....	41
4.15. Javascript.....	42
5. Ανάλυση του κώδικα και του αλγορίθμου στην διαδικτυακή μας εφαρμογή.....	42
5.1 Εισαγωγή.....	43
5.2. Ο Αλγόριθμος Levenshtein distance.....	44
5.2.1 Χρήση του αλγορίθμου Levenshtein distance στην εφαρμογή μας.....	44
5.3. Χρήση του design pattern Mediator (Διαμοιραστή) στην εφαρμογή μας.....	47
5.4. Οργάνωση των δεδομένων (Data model).....	48
5.5. Αποθήκευση των δεδομένων.....	49
5.6. Η υπηρεσία Web Socket.....	50
5.7. Η υπηρεσία chatMessageEncoder και chatMessageDecoder.....	51
5.8. Η Html σελίδα της εφαρμογής και η χρήση της Javascript.....	53
5.9. Η Firebase στην εφαρμογή.....	54
6. Η λειτουργία της εφαρμογής.....	55
6.1. Εισαγωγή.....	55
6.2. Σύνδεση χρήστη στην εφαρμογή.....	55
6.3. Εικονική συνομιλία μεταξύ χρηστών.....	58
7. Συμπεράσματα.....	60
8. Βιβλιογραφία.....	61

1. Εισαγωγή

Τα τελευταία χρόνια σε πολλές χώρες του κόσμου, υπάρχουν διάφορα φαινόμενα βίαιης συμπεριφοράς στο σχολικό χώρο, στον εργασιακό χώρο καθώς και στο χώρο των κοινωνικών δικτύων. Μια τέτοια συμπεριφορά αφορά κατάφορη παραβίαση του βασικού δικαιώματος όλων των ανθρώπων να πηγαίνουν στο σχολείο, αν είναι ενήλικοι στην εργασία τους ή να συνομιλούν μέσω των κοινωνικών δικτύων χωρίς να φοβούνται, ότι κάποιος θα τους επιβληθεί με τη βία, ανεξάρτητα από την ηλικία, το φύλο, την εθνική τους ταυτότητα ή οποιονδήποτε άλλο παράγοντα.

Τέτοιες συμπεριφορές μπορεί να έχουν σοβαρές σωματικές και ψυχολογικές επίπτώσεις στους ανθρώπους που τις υφίστανται. Το φαινόμενο αυτό της επιθετικής συμπεριφοράς και βίας ονομάζεται εκφοβισμός ή θυματοποίηση. Και αναφέρεται στις καταστάσεις όπου ένας άνθρωπος ή ένα γκρουπ ανθρώπων θυματοποιούν έναν άλλο άνθρωπο, εκμεταλλευόμενοι μια προυπάρχουσα ανισορροπία δύναμης, σωματικής, πνευματικής, κοινωνικής.

Τα τελευταία χρόνια δημιουργήθηκαν τεχνολογικά επιτεύγματα, όπως οι ηλεκτρονικοί υπολογιστές και τα κινητά τηλέφωνα. Το διαδίκτυο αποτελεί ένα χώρο όπου πολλοί μαθητές ενημερώνονται, μελετούν, επικοινωνούν και ψυχαγωγούνται (Amichai-Hamburger, 2013). Περίπου από 60% έως 80% άτομα εφηβικής ηλικίας το χρησιμοποιούν καθημερινά σε Ευρώπη και Αμερική (Lenhart, Purcell, Smith, & Zickuhr, 2010. Livingstone, Haddon, Görzig, & Ólafsson, 2011).

1.1. Σκοπός της εργασίας

Στην παρούσα εργασία επιδιώκεται να αναπτυχθεί ένα εργαλείο συνομιλίας μεταξύ μαθητών (web chat) και η δημιουργία ενός αλγορίθμου που θα καταγράφει μόνο τις <<κακές>> λέξεις που ταιριάζουν με το λεξικό μας προσεγγίζοντας ένα ποσοστό με αντιμετάθεση γραμμάτων, για την προστασία των αδύναμων μαθητών.

Για παράδειγμα, ένας μαθητής γράφει 'είσαι γάιδαρος' το λεξικό μας περιέχει την λέξη 'γαιδούρι' και όχι την λέξη που έχει εισάγει ο χρήστης, οπότε βλέπουμε ότι υπάρχει ομοιότητα 50% ανάμεσα στις δύο λέξεις. Με αυτό τον τρόπο αποφασίζουμε ότι λέξεις πάνω από το ποσοστό των 50% να αποθηκεύονται στο αρχείο μας, για την ενημέρωση του διαχειριστή συστήματος ή του δασκάλου.

Γενικότερα, μέσω ενός κοινωνικού δικτύου όπως είναι ένα web chat θα γίνει μία απόπειρα προστασίας των χρηστών από οποιαδήποτε μορφή λεκτικής βίας μέσω της αυτόματης καταγγελίας (αλγόριθμος) στον διαχειριστή συστήματος.

2. Εκφοβισμός στο σχολικό περιβάλλον

Στην ενότητα αυτή αναλύονται η θεωρία του εκφοβισμού στο σχολικό περιβάλλον όπου η θυματοποίηση ανθεί, στη συνέχεια αναλύονται οι μορφές που διέπουν την σημασία του εκφοβισμού στο σχολείο, καθώς και τα χαρακτηριστικά των θυτών και των θυμάτων. Τέλος, γίνεται αναφορά στους μηχανισμούς της ομάδας στο σχολικό περιβάλλον.

2.1. Ορισμός εκφοβισμού (bullying)

Ο εκφοβισμός ή θυματοποίηση μπορεί να οριστεί με τον ακόλουθο τρόπο: ένας μαθητής γίνεται αντικείμενο εκφοβισμού ή θυματοποιείται όταν υποβάλλεται, κατ' επανάληψη και κατ' εξακολούθηση, σε αρνητικές ενέργειες από έναν ή περισσότερους άλλους μαθητές (Olweus, 1986,1991). Ο ορισμός δίνει έμφαση στις ενέργειες που πραγματοποιούνται κατ' επανάληψη και κατ' εξακολούθηση. Στο πλαίσιο του σχολικού εκφοβισμού, στόχος γίνεται συνήθως ένας μεμονωμένος μαθητής. Όταν μιλάμε για εκφοβισμό - βία – πρέπει να υπάρχει διαφορά δύναμης, αυτό σημαίνει ότι ο μαθητής ο οποίος εκτίθεται στις αρνητικές ενέργειες δυσκολεύεται να αμυνθεί και είναι αβοήθητος έναντι εκείνου ή εκείνων που τον παρενοχλούν. Επίσης να είναι εμφανής η έκδηλη επιθυμία του θύτη να προκαλέσει στο θύμα σωματικό ή/και ψυχολογικό πόνο.

2.2. Μορφές του σχολικού εκφοβισμού

Ο σχολικός εκφοβισμός και η θυματοποίηση είναι δύο όροι που έχουν μεταφραστεί από τον αγγλικό όρο <<bullying>>, παρατηρούνται στο σχολικό περιβάλλον και δυσχεραίνουν την προσαρμογή των παιδιών. Πιο συγκεκριμένα, με τον όρο θυματοποίηση αποδίδουμε τον αγγλικό όρο <<victimization>> που είναι αποτέλεσμα εκφοβισμού(bullying), δηλαδή μιας συγκεκριμένης μορφής επιθετικής συμπεριφοράς με ιδιαίτερα χαρακτηριστικά (Κυριόπουλος και Παππά, 2008).

Γενικότερα, συμπεριφορές που αποτελούν προσπάθεια εκφοβισμού και θυματοποίησης είναι (Νικολαΐδης και Παπαστυλιανού, 2013) :

- Οι μορφές που έχουν να κάνουν με σωματική βία, όπως χτυπήματα, φτύσιμο κ.α
- Κλοπή προσωπικών αντικειμένων.
- Λεκτικές συμπεριφορές, όπως βρίσιμο, αρνητικά σχόλια εναντίον του παιδιού-στόχου ή της οικογενειάς του.
- Αποκλεισμός του παιδιού από το παιχνίδι με άλλα παιδιά.
- Κοροϊδία για τον τρόπο που το παιδί ντύνεται.
- Ηλεκτρονικός εκφοβισμός, μέσω εργαλείων τεχνολογιών επικοινωνίας.

2.2.1 Τα χαρακτηριστικά των θυμάτων

Υπάρχουν έρευνες που έχουν διαμορφώσει μια σχετικά σαφή εικόνα των τυπικών θυμάτων (Olweus, 1973 και 1978 Bjorkqvist et al.,1982-Lagerspetz et al.,1982 Boulton και Smith, 1994-Perry et al.,1988-Farrington, 1993). Η εικόνα αυτή αντιπροσωπεύει τόσο τα αγόρια όσο και τα κορίτσια.

Τα θύματα είναι αγχώδη και ανασφαλή από το μέσο όρο των μαθητών. Εμφανίζουν επίσης χαμηλή αυτοεκτίμηση, αδυναμία στην επίλυση προβλημάτων. Όταν δέχονται επίθεση από άλλους μαθητές, αντιδρούν συχνά με κλάμα και με απόσυρση. Κυριαρχεί στα θύματα το αίσθημα μοναξιάς,

ενω έχουν χαμηλές σχολικές επιδόσεις και απουσίες κατά τη φοίτηση. Τα θύματα εμφανίζουν στο μέλλον αδυναμία να αναλάβουν τις ευθύνες τους.

Κατά κανόνα στο σχολείο τα θύματα δείχνουν απομόνωση και δεν έχουν ούτε έναν καλό φίλο στην τάξη τους. Ωστόσο, δεν είναι αντιδραστικά ούτε έχουν ενοχλητική ή επιθετική συμπεριφορά. Τα παιδιά αυτά έχουν αρνητική στάση προς τη βία και τη χρήση βίαιων μέσων.

Στην περίπτωση που είναι αγόρια, θα είναι μάλλον πιο αδύναμα από το μέσο όρο (Olweus, 1978). Στα θύματα του εκφοβισμού περιλαμβάνονται αγόρια και κορίτσια όλων των ηλικιών, σωματικού μεγέθους και νοητικού επιπέδου. Ωστόσο, ορισμένα παιδιά είναι πιο πιθανό να υποστούν βία γιατί εμφανίζονται μικρά αδύναμα, ευαίσθητα ή διαφορετικά από τους συνομιληκούς τους.

Μερικά παιδιά προσπαθούν να μειώσουν τον κίνδυνο, να πέσουν θύματα εκφοβισμού με το να τύνονται ή να συμπεριφέρονται με τέτοιο τρόπο ώστε να ταιριάζουν με το σύνολο. Όμως δεν είναι όλα τα παιδιά ικανά να αλλάξουν τα προσωπικά χαρακτηριστικά τους και ούτε θα πρέπει να καταπιέζονται για να το κάνουν.

2.2.2 Τα χαρακτηριστικά των Νταήδων

Το χαρακτηριστικό γνώρισμα των νταήδων είναι η επιθετικότητα, καθώς και η αδυναμία τους για την τήρηση κανόνων και ορίων, έχουν έντονο παρορμητισμό, μειωμένη αίσθηση αυτοελέγχου και έλειψης του μέτρου (Goleman, 1998). Έχει αποδειχθεί μετά από έρευνες, ότι τα παιδιά – θύτες που ασκούν βία εξελίσσονται σε μεγάλο ποσοστό που αγγίζει το 50% σε ενήλικες με παραβατική συμπεριφορά (Γλατής, 2006).

Τα αγόρια νταήδες έχουν συνήθως μεγαλύτερη σωματική δύναμη από τους υπόλοιπους συνομιληκούς τους και ειδικότερα από τα θύματα (Olweus, 1978). Ένα παιδί με αυταρχική συμπεριφορά, είναι ο συνηθέστερος τύπος θύτη, επιθετικό ακόμα και προς τους ενήλικες, παρορμητικό και με σωματική δύναμη. Αποζητά την ανοιχτή και δημόσια διαμάχη με στόχο να επιβεβαιώσει την υπεροχή του και να επιβληθεί στην ομάδα των συνομιληκών του.

Ψυχολόγοι και ψυχίατροι συμφωνούν ότι άτομα με επιθετική συμπεριφορά είναι <<κατά βάθος>> ανασφαλή, υπάρχουν έρευνες που τους διαψεύδουν, οι οποίες έδειξαν ότι οι θύτες έδειχναν λίγο άγχος και ανασφάλεια, ή βρίσκονταν στο μέσο όρο των μετρήσεων (1981 και 1984 βλέπε επίσης Pukkinen & Tremblay, 1992).



2.3. Οι μηχανισμοί της ομάδας

Μια διαφορετική οπτική της ενδοσχολικής βίας εστιάζει το ενδιαφέρον της στη σημασία της ομάδας και στο πώς ακόμα και οι μαθητές που δεν εμπλέκονται άμεσα με το θύτη επηρεάζονται από την επιθετικότητα (Salmivalli, 2010, Espelage & Swearer, 2008).

Πολλές μελέτες έχουν δείξει ότι τόσο τα παιδιά, όσο και οι ενήλικοι μπορούν να φερθούν πιο επιθετικά αν δουν κάποιον άλλο, ένα <<πρότυπο>>, να έχει επιθετική στάση. Πρώτη η Salmivalli (2010) αναγνώρισε τέσσερις συμπληρωματικούς ρόλους (ο βοηθός του θύτη, ο υποστηρικτής του θύτη, ο εξωτερικός ουδέτερος παρατηρητής και ο υπερασπιστής του θύματος) στο φαινόμενο εκτός των δύο βασικών εμπλεκόμενων, θύτη και θύματος και πολλές έρευνες την ακολούθησαν, για να μελετήσουν αυτό που ο Olweus (1993) αποκάλεσε «κύκλο του εκφοβισμού».

Ειδικά στην έναρξη της εφηβείας, η ομάδα των συνομηλίκων γίνεται ο σημαντικότερος «άλλος», στον οποίο τα αγόρια τείνουν να αναζητούν την αποδοχή μέσω της δύναμης και τα κορίτσια μέσω της εμφάνισης (Espelage & Swearer, 2008).

Είναι γνωστό από την ατομική ψυχολογία ότι το αίσθημα ευθύνης του θύτη για μια αρνητική πράξη, όπως ο εκφοβισμός μπορεί να μειωθεί σημαντικά όταν συμμετέχουν σε αυτήν αρκετά άτομα.

Τέλος, με το πέρασμα του χρόνου μπορεί να συντελεστούν αλλαγές στη γνώμη των συμμαθητών για το θύμα, εξαιτίας των επιθέσεων και των ταπεινωτικών σχολίων.

2.4. Τρόποι αντιμετώπισης της θυματοποίησης

Η έκταση του σχολικού εκφοβισμού σε συνδυασμό με την κρισιμότητα των συνεπειών του καθιστούν επιτακτική την ανάγκη άμεσης αντιμετώπισής του. Οι περιστάσεις καλούν για άμεση παρέμβαση.

Η πρόληψη είναι αποτελεσματικότερη της θεραπείας. Στην περίπτωση του σχολικού εκφοβισμού, παρέμβαση και πρόληψη είναι αλληλένδετες.

Η απαλοιφή του φαινομένου πρέπει να στοχεύει τόσο στην πρόληψη όσο και στη διαχείριση του φόβου. Και μάλιστα οι παρεμβάσεις που αποσκοπούν στη μείωση του σχολικού εκφοβισμού επιβάλλεται να συμπεριλάβουν όλα τα επίπεδα όπου κινούνται τα παιδιά, δηλαδή το σχολείο, την οικογένεια και την τοπική κοινωνία (Espelage & Swearer, 2003 στην εργασία της Τσικρικά, 2009).

Έρευνες αναφέρουν ότι τα παιδιά ζητούν συμβουλές, για τις δύσκολες αποφάσεις από τους γονείς και από τους δασκάλους. Οι γονείς και οι δάσκαλοι θα πρέπει να βοηθούν τα παιδιά, να τα στέλνουν σε δραστηριότητες και χόμπι. Οι δραστηριότητες αυτές δίνουν την ευκαιρία στα παιδιά, να διασκεδάσουν και να συναντήσουν συνομήλικους με τα ίδια ενδιαφέροντα, που μπορεί να τα βοηθήσουν και να τα προστατέψουν από τον εκφοβισμό.

Οι κοινωνικοί οι πολιτισμικοί γενικότερα και περιβαλλοντικοί παράγοντες παίζουν ένα σημαντικό ρόλο στην μάθηση των επιθετικών συμπεριφορών. Η κοινωνικοποίηση ως διαδικασία του παιδιού στην κοινωνία έχει κατά τον Durkheim το χαρακτήρα ενός ηθικού καταναγκασμού και είναι αναγκαίος για την σωστή ένταξη του παιδιού στην κοινωνία και την πολιτιστική ομοιογενοποίηση των μελών της ομάδας.

Οι εκπαιδευτικοί, οι οικογένειες, οι μαθητές του σχολείου είναι συνυπεύθυνοι για την σωστή αντιμετώπιση του κοινωνικού προβλήματος, θα πρέπει να περιοριστεί κάθε μορφή βίας. Κανένας

μαθητής δεν θα πρέπει να υφίσταται τις επιθετικές συμπεριφορές συνομηλίκων του. Κανένα παιδί δεν θα πρέπει να αποκλείεται από ομαδικές δραστηριότητες, για να μην αισθάνονται ότι είναι μόνο τους ή ότι τα έχουν απορρίψει οι συνομηλικοί τους.

Οι γονείς θα πρέπει να ακούνε τα παιδιά τους, σε περίπτωση που θελήσουν να μιλήσουν θα πρέπει να τα επιβραβεύσουν γι αυτό. Πολλά παιδιά δεν λένε ότι φοβούνται, θεωρώντας ότι οι γονείς του δεν θα τα πιστέψουν, ή ότι θα επιδεινωθεί η θέση τους στο σχολείο αν μαθευτεί. Άλλες φορές τα παιδιά θεωρούν ότι εκείνα είναι υπεύθυνα γι αυτή την συμπεριφορά. Οι γονείς θα πρέπει να αγκαλιάσουν τα παιδιά τους και να τους δώσουν να καταλάβουν, ότι είναι ευθύνη του θύτη και όχι του θύματος η βίαιη αυτή συμπεριφορά και ότι δεν την προκαλούν τα ίδια.

Τέλος, θα πρέπει οι γονείς να αποτανθούν στους δασκάλους και όχι στους γονείς του θύτη, για να αντιμετωπίσουν αυτή τη βίαιη συμπεριφορά.

3. Διαδικτυακός εκφοβισμός (Cyber bullying)

Στην παρούσα ενότητα θα ορίσουμε τον διαδικτυακό εκφοβισμό, τις μορφές του διαδικτυακού εκφοβισμού, καθώς και την αντιμετώπιση της απειλής και του αποκλεισμού.

Το cyber bullying φαντάζει ίσως αθώο. Μπορεί όμως να έχει πολύ σοβαρές συνέπειες για τη ζωή του θύματος, τόσο ψυχολογική όσο και σωματική μέσω αυτοτραυματισμού. Ενδεχομένως μπορεί να δημιουργηθούν περιπτώσεις όπως αποχή από τα μαθήματα, κοινωνική απομόνωση, απότομη πτώση στις σχολικές επιδόσεις, εκτέλεση πράξεων αντίθετων με τον χαρακτήρα του παιδιού ή παράνομες πράξεις λόγω εκβιασμού.

Σε αρκετές περιπτώσεις τα παιδιά έχουν έντονη επιθυμία να αλλάξουν σχολικό περιβάλλον, κάποιες φορές έχουν κατάθλιψη η οποία μπορεί να καταλήξει ακόμα και σε αυτοκτονία.

**TAKE A
STAND
AGAINST
CYBERBULLING**

***HELP OTHERS WHEN
THEY NEED IT**

3.1. Ορισμός διαδικτυακού εκφοβισμού

Ο όρος διαδικτυακός εκφοβισμός αφορά τα παιδιά ή και εφήβους και δηλώνει την παρενόχληση, τον εκφοβισμό, την απειλή την παρενόχληση, που δέχονται μέσω του διαδικτύου, κινητών ή άλλων τεχνολογικών ψηφιακών μέσων απο συνομηλικούς τους.

Είναι μία επιθετική πράξη ή συμπεριφορά ενός ατόμου ή μιας ομάδας, η οποία χρησιμοποιεί διαδικτυακά μέσα. Υπάρχουν διάφορες παραλλαγές του διαδικτυακού εκφοβισμού όπως είναι η δημοσίευση κειμένου ή εικόνων που σκοπεύουν να βλάψουν η να φέρουν σε δύσκολη θέση το θύμα.

Ο διαδικτυακός εκφοβισμός είναι πολλές φορές παρόμοιος με τον παραδοσιακό εκφοβισμό, με κάποιες αξιοσημείωτες διαφορές, μία απο αυτές είναι ότι τα θύματα πολλές φορές δεν γνωρίζουν την ταυτότητα του θύτη. Η παρενόχληση του θύματος μέσω για παράδειγμα μιας εικόνας ή ενός κειμένου μπορεί να έχει ευρεία διάδοση και να διαμοιραστεί πολύ εύκολα ανάμεσα στους χρήστες, μέσω του διαδικτύου.

Ο εκφοβισμός μέσω του διαδικτύου μπορεί να είναι άμεσος ή έμμεσος εμπλέκοντας και άλλα άτομα που ενδεχομένως να μην γνωρίζουν προσωπικά το θύμα.

Παρακάτω βλέπουμε σε διάγραμμα μία έρευνα που έγινε και τα ευρήματα που είχε, η οποία επιβεβαιώνει άλλες έρευνες. Η συγκεκριμένη έδειξε ότι η ηλικία και το φύλο δείχνουν ότι τα αγόρια και τα κορίτσια χρησιμοποιούν το διαδίκτυο online on-the-go που σημαίνει μέσω του κινητού στο δρόμο. Bullying via SNS και bullying via IM σημαίνει ότι έγιναν θύματα μέσω μιας ιστοσελίδας και μέσω άμεσου τηλεφωνικού ή ηλεκτρονικού μηνύματος.

	Range (scale) or number of items	M	SD	Correlations*										
				1.	2.	3.	4.	5.	6.	7.	8.	9.		
1. Age	9-16	13.23	2.09	1.00										
2. Gender	Male = 0	.59	.49	.04	1.00									
3. Online on-the-go	No = 0	.77	.42	.15**	.15**	1.00								
4. Sensation seeking	1-3	1.56	.61	.12**	.12**	.09**	1.00							
5. Psychological difficulties	1-3	1.55	.29	-.04	-.04	-.02	.24**	1.00						
6. Intensity of harm	0-3	2.47	1.02	.00	.00	.00	-.11**	.13**	1.00					
7. Bullying via SNS	No = 0	.45	.50	.12**	.12**	.05	.07*	.10**	.07**	1.00				
8. Bullying via IM	No = 0	.46	.50	.11**	.11**	.02	.01	.01	.09**	-.19**	1.00			
9. Types of online bullying	1-5	1.35	.72	.16**	.16**	.06*	.17**	.20**	.15**	.24**	.14**	1.00		
10. Mobile victim	Online victim = 0	.32	.47	.16**	.16**	.12**	.12**	.09**	.09**	.11**	.16**	.15**	1.00	

* Correlation is significant at the 0.05 level (2-tailed), ** Correlation is significant at the 0.01 level (2-tailed).

Notes: *Coefficients for correlations between dichotomous and continuous measures are point-biserial (r_{pb}) and between two dichotomous measures are measured by χ^2 -analyses and indicated by Φ .

Analysis was performed on those children who had indicated to have been bullied by someone else on the internet and showed valid responses on all measures ($N = 1300$).

Σχεδιασμός, Ανάπτυξη και Εφαρμογή Προηγμένου Λογισμικού Διεπαφής χρηστών διαδικτυακών συζητήσεων (webChat), με τη χρήση της γλώσσας Java, για το σκοπό ανίχνευσης λέξεων της φυσικής γλώσσας που μαρτυρούν θυματοποίηση (Bullying).

3.2. Αιτίες διαδικτυακού εκφοβισμού

Συχνά άτομα νεαρά εκτίθονται στον διαδικτυακό εκφοβισμό, εξαιτίας της βίωσης έντονων συναισθημάτων όπως θυμός, απόγνωση ακόμα και εκδίκηση, που μπορεί να προέρχεται από τις σχέσεις στο οικογενειακό περιβάλλον, όσο και μιας ευρύτερης κοινωνικής απομόνωσης που παρουσιάζει το άτομο.

Ένας άλλος σημαντικός λόγος είναι ότι ο εκφοβισμός στο διαδίκτυο είναι πιο ασφαλής, από τον παραδοσιακό εκφοβισμό γιατί είναι κατά βάση ανώνυμος και ο χρήστης κρύβει την ταυτότητα του, πιστεύοντας ότι δεν θα πιαστεί.

Τέλος, σε κάποιες περιπτώσεις ο διαδικτυακός εκφοβισμός αποτελεί μορφή ψυχαγωγίας, στοχεύοντας στην ικανοποίηση της ανάγκης εκείνης που σχετίζεται με την επιβολή εξουσίας και ελέγχου πάνω στο θύμα.

3.3. Μορφές διαδικτυακού εκφοβισμού

- Εκφοβισμός με επαναλαμβανόμενα γραπτά ή τηλεφωνικά μηνύματα.
- Παρενόχληση και εκφοβισμό του θύματος της διαδικτυακής του δραστηριότητας.
- Παράνομη είσοδο σε προσωπικούς διαδικτυακούς λογαριασμούς του θύματος. Για παράδειγμα, είσοδο σε κοινωνικά δίκτυα, ή email επικοινωνίας.
- Δημοσίευση και διαμοιρασμός εικόνων σε γνωστούς και αγνώστους, χωρίς την συγκατάθεση του θύματος.
- Αποστολή προσωπικών πληροφοριών του θύματος, σε πολλαπλούς παραλήπτες.
- Υποκίνηση τρίτων ατόμων για την διαδικτυακή παρακολούθηση του θύματος.

3.4. Τρόποι αντιμετώπισης διαδικτυακού εκφοβισμού

Αν το παιδί πέσει θύμα εκφοβισμού στο διαδίκτυο, οι γονείς θα πρέπει να ενημερώσουν το παιδί τους, ότι σε περίπτωση που ένας <<διαδικτυακός φίλος>> τους στείλει προσβλητικά μηνύματα, θα μπορούν πολύ εύκολα να τον αποκλείσουν από το email του ή από τα κοινωνικά δίκτυα. Οι ίδιοι οι γονείς μπορούν να προβούν σε μια σειρά ενεργειών για να προστατεύσουν το παιδί τους.

Οι γονείς θα πρέπει να ενημερώσουν τα παιδιά τους να μην σβήσουν τα προσβλητικά μηνύματα αλλά να τα αποθηκεύσουν σαν αποδεικτικά στοιχεία. Έπειτα, θα πρέπει να τους δείξουν τα απειλητικά μηνύματα.

Κάποιες φορές ενδέχεται ο θύτης να είναι γνωστός του θύματος, σε αυτή την περίπτωση θα πρέπει οι γονείς να ενημερώσουν τους γονείς του παιδιού ή τον δάσκαλο σε περίπτωση που πάνε στο ίδιο σχολείο. Υπάρχουν περιπτώσεις που το θύμα εμφανίζεται αγχωμένο ή φοβισμένο, σε αυτή την περίπτωση οι γονείς θα πρέπει να ζητήσουν βοήθεια από ψυχολόγο.

Αντίθετα, αν οι γονείς καταλάβουν ότι το παιδί τους χρησιμοποιεί την τεχνολογία για να εκφοβίσει ή να απειλήσει κάποιον, τότε θα πρέπει να μιλήσουν στο παιδί τους για τον διαδικτυακό εκφοβισμό και να του δώσει να καταλάβει ότι είναι κάτι ανεπίτρεπτο. Ακόμα, θα πρέπει να συζητήσουν με το παιδί τους, τι το ώθησε ώστε να προχωρήσει σε αυτή την πράξη, ποιές είναι οι αιτίες που το οδήγησαν.

Σε περίπτωση που το παιδί δεν συνηθιστοποίησε ότι αυτό που έκανε ήταν εκφοβισμός, θα πρέπει οι γονείς να του εξηγήσουν ότι ο εκφοβισμός δεν είναι μόνο σωματικός. Δεν θα πρέπει να χρησιμοποιείται η τεχνολογία ως άλλοθι για τον εξευτελισμό της ατομικής προσωπικότητας του θύματος. Επίσης, θα πρέπει να μιλήσουν στο δάσκαλο και να δείξουν ότι είναι προθυμοί να συνεργαστούν με το σχολείο για την αντιμετώπιση αυτής της συμπεριφοράς.

Τέλος, υπάρχουν φορές που τα παιδιά είναι οι τρίτοι στον διαδικτυακό εκφοβισμό, σε αυτή την περίπτωση που είναι μάρτυρες τέτοιας συμπεριφοράς θα πρέπει να μιλήσουν στους ίδιους τους γονείς ή στους δασκάλους τους.



3.5. Ακατάλληλες διαδικτυακές ιστοσελίδες

Ο όρος ακατάλληλες διαδικτυακές ιστοσελίδες είναι υποκειμενικός σε σχέση με το φύλο την ηλικία και την ψυχική κατάσταση του ατόμου. Ένα περιεχόμενο ιστοσελίδας μπορεί να θεωρηθεί ακατάλληλο υλικό, σε περίπτωση που μπορεί να προκαλέσει ψυχικές διαταραχές ή να σοκάρει το άτομο. Συνήθως, ο όρος ακατάλληλο περιεχόμενο μια διαδικτυακής ιστοσελίδας, μπορεί να θεωρηθεί οτιδήποτε περιλαμβάνει ρατσιστικό ή ξενοφοβικό περιεχόμενο, προώθηση βίας, παρουσίαση πορνογραφικού υλικού ή προώθηση παράνομων τυχερών παιχνιδιών.

3.6. Διαδικτυακή αποπλάνηση ανηλίκου (grooming)

Η αποπλάνηση ανηλίκου είναι η διαδικασία κατά την οποία ο θύτης, προσποιούμενος ότι είναι έφηβος, χρησιμοποιεί δωμάτια ανοιχτής επικοινωνίας προσελκύει παιδιά με σκοπό να τα κακοποιήσει. Τέτοιου είδους ιστοσελίδες θεωρούνται από τα παιδιά ασφαλείς τόποι συνομιλίας στο διαδίκτυο, εξαιτίας της λανθασμένης εκτίμησης των παιδιών ότι διατηρείται η ανωνυμία τους.

Τα παιδιά δεν θα πρέπει να υποκύπτουν σε κανένα διαδικτυακό εκβιασμό και θα πρέπει άμεσα να καταγγείλουν τέτοιες συμπεριφορές στους γονείς ή τον δάσκαλο. Πρέπει όλοι να θυμούνται, ότι είναι πολύ εύκολο κάποιος να κρύψει την ταυτότητά του στο διαδίκτυο ή να πει

ψέμματα για το ποιός πραγματικά είναι. Δεν θα πρέπει να ανταλλάσσονται πληροφορίες βίντεο ή φωτογραφίες, διότι μπορούν να μείνουν στο διαδίκτυο και να διανεμηθούν από τον οποιοδήποτε. Τα παιδιά δεν θα πρέπει σε καμιά περίπτωση να κανονίζουν συναντήσεις με ανθρώπους που δεν γνωρίζουν έστω και αν είναι <<φίλοι>> στο διαδίκτυο. Οι γονείς θα πρέπει να επιβλέπουν τα παιδιά τους και να καταγγείλουν αυτά τα άτομα, στην υπηρεσία διαδικτυακού εγκλήματος ή στην ελληνική γραμμή για το παράνομο περιεχόμενο στο διαδίκτυο.

Η διαδικτυακή μας εφαρμογή είναι η ίδια διαδικτυακός χώρος συνομιλίας ελεγχόμενος από τον δάσκαλο, όπου τα παιδιά δεν έχουν ανωνυμία μεταξύ τους αλλά ονοματεπώνυμο, όπου η εφαρμογή ελέγχει τα λεγόμενά τους με αυτόματη καταγγελία στον δάσκαλο ή στον διαχειριστή συστήματος σε περίπτωση παράβατικής συμπεριφοράς.

3.7. Προστασία από διαδικτυακές ιστοσελίδες

Οι γονείς ή οι δάσκαλοι σε περίπτωση που πέσει στην αντίληψη τους ότι το παιδί έχει πρόσβαση σε ακατάλληλο περιεχόμενο θα πρέπει να του αποκλείσουν την πρόσβαση. Επίσης θα πρέπει οι γονείς να ενημερώνονται για το περιεχόμενο που υπάρχει στο διαδίκτυο και να συζητούν με τα παιδιά για το θέμα αυτό. Να διασταυρώνουν τις πληροφορίες που διαβάζουν στο διαδίκτυο από πολλαπλές πηγές. Εάν τα παιδιά είναι μικρότερα των 10 ετών θα πρέπει να καθόνται οι γονείς μαζί τους.

Οι γονείς μπορούν να ελέγχουν και να καταγράφουν την πλοήγηση των παιδιών με ειδικό λογισμικό που καταγράφουν την δραστηριότητα του υπολογιστή. Για παράδειγμα μέσω του ιστορικού του εκάστοτε λογισμικού πλοήγησης, φυλλομετρητή (google, mozilla, internet explorer).

4. Τεχνικά χαρακτηριστικά της διαδικτυακής εφαρμογής μας.

Στην ενότητα αυτή θα μιλήσουμε για την γλώσσα προγραμματισμού Java, καθώς και για τα εργαλεία που χρησιμοποιήσαμε εντός της εφαρμογής, προκειμένου η εφαρμογή να είναι λειτουργική. Πρέπει κάποιος να κατανοήσει τις έννοιες όπως Java EE, Java Container, XML, WebSocket προκειμένου να περάσουμε στο επόμενο κεφάλαιο στην ανάλυση της εφαρμογής μας.

4.1. Η ιστορία της Java

Η γλώσσα προγραμματισμού που χρησιμοποιήσαμε για την υλοποίηση της διαδικτυακής μας εφαρμογής, είναι η Java. Η Java είναι μία αντικειμενοστρεφής γλώσσα προγραμματισμού που σχεδιάστηκε απ την εταιρεία Sun Microsystems. Το 1991 η Sun έψαχνε το κατάλληλο εργαλείο, για να αποτελέσει μία πλατφόρμα ανάπτυξης λογισμικού που θα κάλυπτε ανάγκες, από μικροσυσκευές έως πολύπλοκα συστήματα παραγωγής γραφικών. Οι γλώσσες προγραμματισμού που ήταν πιο διαδεδομένες εκείνη την εποχή ήταν ή C++ και η C.

Ο δημιουργός της Java, James Gosling, εκείνη την εποχή εργαζόταν στην Sun και έκανε πειραματισμούς με τις ήδη υπάρχουσες γλώσσες όπως η C++ , κατα καιρούς παρουσίαζε κάποιες πειραματικές γλώσσες (C++ ++, που μετέπειτα ονομάστηκε C#) ως πρότυπα για το νέο εργαλείο που αναζητούσαν στη Sun. Τελικά κατέληξαν στην γλώσσα Oak, το όνομα της το πήρε από το ομώνυμο δένδρο βελανιδιάς το οποίο είχε ο Gosling έξω απ το γραφείο του και έβλεπε κάθε μέρα.

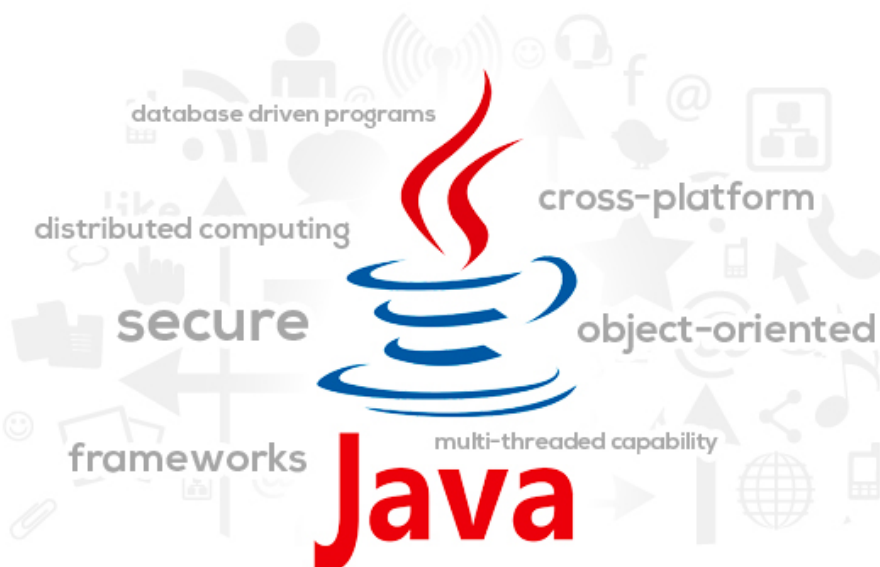
Η Oak ήταν μία γλώσσα που είχε αρκετή συνάφεια με την γλώσσα C++. Όμως είχε πιο έντονο τον αντικειμενοστρεφή χαρακτήρα και ήταν αρκετά πιο απλή απ την C++. Οι υπεύθυνοι ανάπτυξης της νέας γλώσσας αποφάσισαν να την μετονομάσουν απο Oak σε Java, που ήταν το όνομα της αγαπημένης ποικιλίας καφέ για τους δημιουργούς της. Η πρώτη εμφάνιση της Java αλλά και του HotJava (πλοηγός με υποστήριξη Java) στη βιομηχανία της πληροφορικής έγινε το Μάρτιο του 1995.

Ο πρώτος μεταγλωττιστής (compiler) ήταν γραμμένος στη γλώσσα C από τον James Gosling. Το 1994, ο A.Van Hoff γράφει ξανά τον μεταγλωττιστή της Java. Το Δεκέμβριο του 1995 πρώτες οι IBM, Borland, Mitsubishi Electronics, Sybase και Symantec ανακοινώνουν σχέδια να χρησιμοποιήσουν τη Java για την δημιουργία λογισμικού. Από εκεί και πέρα η Java ακολουθεί μία ανοδική πορεία και είναι πλέον μία από τις πιο δημοφιλείς γλώσσες στον χώρο της πληροφορικής. Στις 13 Νοεμβρίου του 2006 η Java έγινε πλέον μια γλώσσα ανοιχτού κώδικα (GPL) όσον αφορά το μεταγλωττιστή (javac) και το πακέτο ανάπτυξης (JDK, Java Development Kit).

Στις 27 Απριλίου 2010 η εταιρεία λογισμικού Oracle Corporation εξαγόρασε την Sun Microsystems και των τεχνολογιών της (πνευματικά δικαιώματα) που η εταιρεία Sun είχε στην κατοχή της, η συμφωνία αυτή ήταν ορόσημο για την ανάπτυξη και το μέλλον του οικοσυστήματος της Java [12].

Τέλος, η Java είναι μία Cross-platform γλώσσα προγραμματισμού, το οποίο σημαίνει ότι μπορεί να τρέξει σε διαφορετικά λειτουργικά συστήματα και πλατφόρμες, "Write Once, Run Anywhere" (WORA). Υπάρχουν διαφορετικές εκδόσεις που μπορεί να χρησιμοποιήσει κάποιος προγραμματιστής όπως οι Java Standard Edition, Java Enterprise Edition, Java Micro Edition.

Στην δική μας περίπτωση για την υλοποίηση της εφαρμογής χρησιμοποιήσαμε την Java Enterprise Edition.

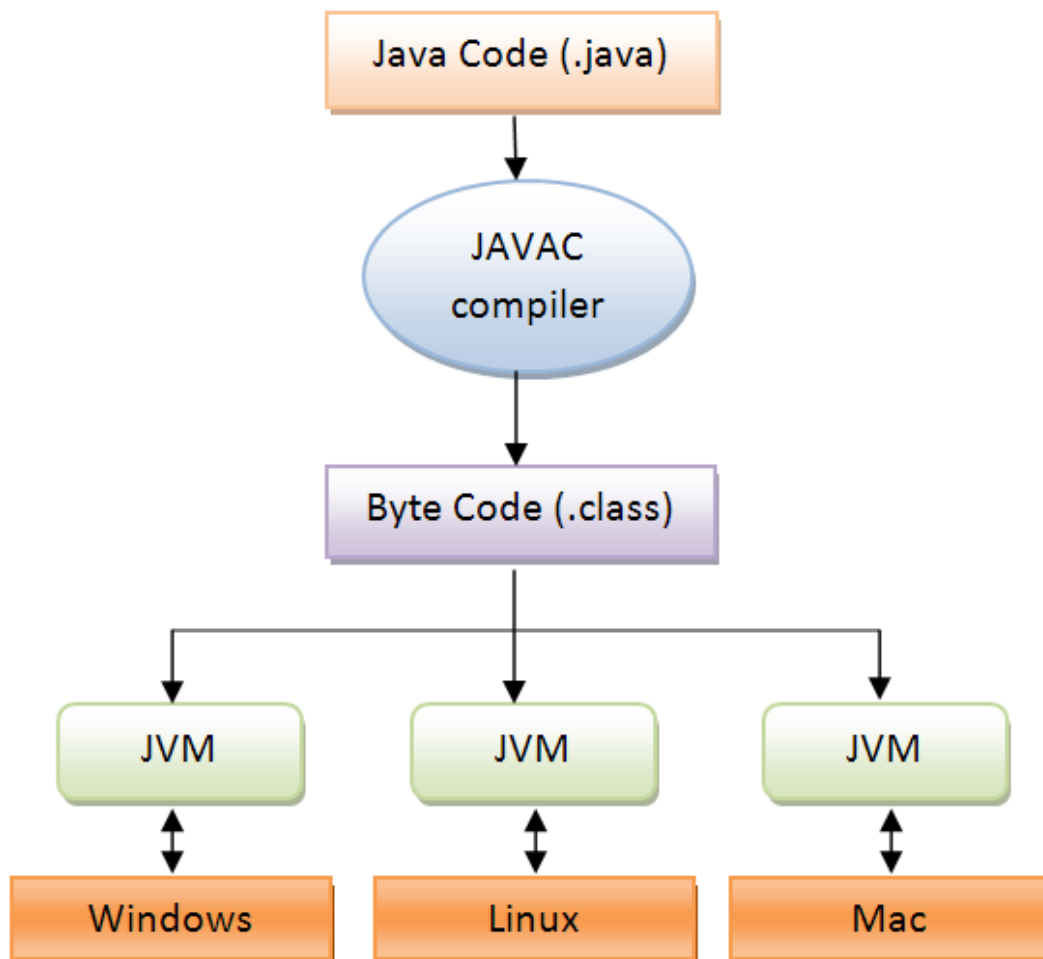


4.2. Τα χαρακτηριστικά της Java και η εικονική της μηχανή (Virtual Machine)

Το μεγαλύτερο πλεονέκτημα της Java έναντι των άλλων γλωσσών προγραμματισμού είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Ότι πρόγραμμα γραφτεί σε Java θα τρέχουν ακριβώς το ίδιο σε όλα τα λειτουργικά συστήματα Windows, Linux, Unix και Macintosh. Για να γίνει όμως αυτό πραγματικότητα θα πρέπει, το πρόγραμμα της Java να είναι <<κατανοητά>> από κάθε υπολογιστή ανεξάρτητα από την κεντρική μονάδα επεξεργασίας. Ο συμβολικός κώδικας (assembly) που μεταφράζεται και εκτελείται σε Windows είναι διαφορετικός από αυτόν που μεταφράζεται και εκτελείται σε έναν υπολογιστή Macintosh. Η λύση δόθηκε με την ανάπτυξη της Εικονικής Μηχανής (Virtual Machine ή VM ή EM στα ελληνικά).

Όταν γραφτεί ένα πρόγραμμα σε Java, στη συνέχεια μεταγλωττίζεται μέσω του μεταγλωττιστή `javac`, ο οποίος παράγει αρχεία `.class` (κώδικας `byte` ή `bytecode`). Ο κώδικας `byte` είναι η μορφή που παίρνει ο πηγαίος κώδικας της Java όταν μεταγλωττιστεί. Όταν θα εκτελεστεί η εφαρμογή σε ένα μηχάνημα, το Java Virtual Machine που πρέπει να είναι εγκατεστημένο σε αυτό θα αναλάβει να διαβάσει τα αρχεία `.class`. Στην συνέχεια τα μεταφράζει σε γλώσσα μηχανής που καταλαβαίνει ο υπολογιστής, που να υποστηρίζεται από το λειτουργικό σύστημα και τον επεξεργαστή. Αυτό συμβαίνει με την παραδοσιακή Εικονική Μηχανή (Virtual Machine).[3] Πιο σύγχρονες εφαρμογές της εικονικής Μηχανής μπορούν και μεταγλωττίζουν εκ των προτέρων τμήματα `bytecode` απευθείας σε κώδικα μηχανής (εγγενή κώδικα ή `native code`) με αποτέλεσμα να βελτιώνεται η ταχύτητα. Χωρίς αυτό δε θα ήταν δυνατή η εκτέλεση λογισμικού γραμμένου σε Java. Η JVM είναι λογισμικό που εξαρτάται από την πλατφόρμα, δηλαδή για κάθε είδος λειτουργικού συστήματος και αρχιτεκτονικής επεξεργαστή υπάρχει διαφορετική έκδοση του. Έτσι υπάρχουν διαφορετικές JVM για Windows, Linux, Unix, Macintosh, κινητά τηλέφωνα, παιχνιδιομηχανές κλπ.

Οτιδήποτε θέλει να κάνει ο προγραμματιστής (ή ο χρήστης) γίνεται μέσω της εικονικής μηχανής. Αυτό βοηθάει στο να υπάρχει μεγαλύτερη ασφάλεια στο σύστημα γιατί η εικονική μηχανή είναι υπεύθυνη για την επικοινωνία χρήστη - υπολογιστή. Ο προγραμματιστής δεν μπορεί να γράψει κώδικα ο οποίος θα έχει καταστροφικά αποτελέσματα για τον υπολογιστή γιατί η εικονική μηχανή θα τον ανιχνεύσει και δε θα επιτρέψει να εκτελεστεί. Από την άλλη μεριά ούτε ο χρήστης μπορεί να κατεβάσει «κακό» κώδικα από το δίκτυο και να τον εκτελέσει. Αυτό είναι ιδιαίτερα χρήσιμο για μεγάλα καταναμεμημένα συστήματα όπου πολλοί χρήστες χρησιμοποιούν το ίδιο πρόγραμμα συγχρόνως [12].



4.3. Java Platform, Enterprise Edition (Java EE)

Η Java Platform, Enterprise Edition ή αλλιώς Java EE είναι μια ευρέως διαδεδομένη πλατφόρμα για προγραμματισμό Server που στηρίζεται πάνω στην γλώσσα προγραμματισμού Java. Η Java EE διαφέρει από την βασική έκδοση της Java από το γεγονός ότι προσθέτει βιβλιοθήκες οι οποίες παρέχουν λειτουργικότητα ώστε να αναπτυχθεί διαδικτυακό, πολυμερές λογισμικό σε Java, βασισμένο σε αυτοτελή μέρη που τρέχουν σε application servers. Η Java EE ήταν γνωστή παλιότερα και ως J2EE, ονομασία που είχε μέχρι και την έκδοση 1.4. Η επόμενη έκδοση ονομάστηκε Java EE 5 [13].

Οι προγραμματιστές σήμερα έχουν όλο και περισσότερο την ανάγκη για καταναμημένες, αξιόπιστες εφαρμογές που μπορούν να παίξουν παντού, σε οποιοδήποτε λειτουργικό σύστημα και προσφέρουν ασφάλεια και αξιοπιστία των τεχνολογιών διακομιστή (server-side application).

Επίσης, οι εφαρμογές αυτές είναι κεντρικά διαχειριζόμενες και αλληλεπιδρούν με άλλο επιχειρησιακό λογισμικό (Enterprise solutions). Ο σκοπός είναι ότι οι επιχειρησιακές εφαρμογές θα πρέπει να σχεδιάζονται, να κατασκευάζονται και να παράγονται με λιγότερους πόρους και με λιγότερα χρήματα.

Με την πλατφόρμα Java, Enterprise Edition (Java EE), η ανάπτυξη επιχειρηματικών εφαρμογών Java δεν ήταν ποτέ πιο εύκολη ή ταχύτερη σε σχέση με το παρελθόν. Ο στόχος της πλατφόρμας Java EE είναι να παρέχει στους προγραμματιστές ένα ισχυρό σύνολο διεπαφής προγραμματισμού εφαρμογών (Application Programming Interface, API), ενώ μειώνει τον χρόνο ανάπτυξης, μειώνει την πολυπλοκότητα των εφαρμογών και βελτιώνει την απόδοση των εφαρμογών.

Η πλατφόρμα Java EE αναπτύσσεται μέσω της Κοινοτικής Διαδικασίας Java (JCP), η οποία είναι υπεύθυνη για όλες τις τεχνολογίες Java. Ομάδες εμπειρογνομόνων αποτελούμενες από εκπαιδευμένους επθιστήμονες μαζεύουν τα αιτήματα προδιαγραφών (Java Specification Requests, JSRs) για να καθορίσουν τις διάφορες τεχνολογίες Java EE. Το έργο της Κοινότητας Java στο πλαίσιο του προγράμματος JCP συμβάλλει στην εξασφάλιση των προτύπων σταθερότητας της τεχνολογίας Java και της συμβατότητας ανάμεσα στις πλατφόρμες.

4.4. Java EE Dependency Injection

Η Java EE χρησιμοποιεί ένα πρότυπο σχεδίασης το Dependency Injection, το οποίο επιτρέπει στον προγραμματιστή να μην έχει εξαρτήσεις από τον κώδικα. Με αυτό τον τρόπο, μπορεί να αλλάξει τις εξαρτήσεις, είτε κατά τον χρόνο εκτέλεσης, είτε κατά τον χρόνο μεταγλώττισης.

Η Java EE μέσω του Dependency Injection δίνει την δυνατότητα να μπορούν να μετατρεπούν οι Java κλάσεις σε διαχειριζόμενα αντικείμενα και να μπορεί ο προγραμματιστής να τα εισάγει σε οποιοδήποτε άλλο αντικείμενο.

```
@javax.enterprise.context.RequestScoped
public class CurrencyConverter { ... }
```

Use the `javax.inject.Inject` annotation to inject managed beans; for example:

```
public class MyServlet extends HttpServlet {
    @Inject CurrencyConverter cc;
    ...
}
```

Για παράδειγμα, βλέπουμε ότι μπορούμε να κάνουμε Inject την κλάση CurrencyConverter μέσω του Annotation που θα δούμε στην συνέχεια, στην κλάση MyServlet και αυτό το αντικείμενο να είναι εύκολα διαχειρίσιμο.



4.5. Java EE Containers

Οι εφαρμογές πολλών επιπέδων που χρησιμοποιούν thin-Client δεν μπορούν να περιέχουν πολλές γραμμές περίπλοκου κώδικα για τον χειρισμό λεπτομερειών χαμηλού επιπέδου. Η Java EE αρχιτεκτονική είναι ανεξάρτητη κάποιας πλατφόρμας επιτρέποντας την εύκολη ανάπτυξη Java EE εφαρμογών ώστε η λογική της επιχείρησης να οργανώνεται σε επαναχρησιμοποιήσιμα στοιχεία.

Επιπλέον, ο Java EE διακομιστής παρέχει υπηρεσίες με τη μορφή ενός container για κάθε τύπο στοιχείου. Έτσι, ο προγραμματιστής δε χρειάζεται να αναπτύξει ο ίδιος κώδικα για τον χειρισμό λεπτομερειών χαμηλού επιπέδου και μπορεί να συγκεντρωθεί στην επίλυση των προβλημάτων της επιχείρησης.

4.5.1. Java EE Container - Υπηρεσίες

Το Java EE container είναι η σύνδεση μεταξύ ενός στοιχείου και της λειτουργίας χαμηλού επιπέδου μιας συγκεκριμένης πλατφόρμας που υποστηρίζει το στοιχείο. Πριν εκτελεστούν τα web ή enterprise bean στοιχεία ή τα στοιχεία εφαρμογής Client, πρέπει να συγκεντρωθούν σε ένα Java EE module και να τοποθετηθούν (deploy) στον αντίστοιχο container τους.

Η διαδικασία συγκέντρωσης (assembly) σε ένα Java EE module σχετίζεται με τον καθορισμό των ρυθμίσεων του container για κάθε στοιχείο της Java EE εφαρμογής αλλά και της ίδιας Java EE εφαρμογής. Οι ρυθμίσεις του container προσαρμόζουν τις υπηρεσίες που παρέχονται από τον Java EE διακομιστή συμπεριλαμβανομένων των υπηρεσιών ασφάλειας, διαχείρισης συναλλαγών, Java Naming και Directory Interface (JNDI) αναζήτησης (lookups) και απομακρυσμένων συνδέσεων. Οι κυριότερες υπηρεσίες είναι:

- Το Java EE μοντέλο ασφαλείας που επιτρέπει την ρύθμιση ενός web στοιχείου ή enterprise bean ώστε μόνο εξουσιοδοτημένοι χρήστες να έχουν πρόσβαση στις πηγές του συστήματος.
- Το Java EE μοντέλο συναλλαγών που επιτρέπει τον ορισμό σχέσεων μεταξύ μεθόδων που χρειάζονται για την ολοκλήρωση μιας μοναδικής συναλλαγής, ώστε όλες οι μέθοδοι μιας συναλλαγής να αντιμετωπίζονται ως μονάδα.
- Οι υπηρεσίες JNDI αναζήτησης που παρέχουν μια ενοποιημένη διεπιφάνεια ώστε τα στοιχεία της εφαρμογής να μπορούν να έχουν πρόσβαση σε υπηρεσίες ονοματοδοσίας και καταλόγου.

Το Java EE μοντέλο απομακρυσμένης σύνδεσης που διαχειρίζεται χαμηλού επιπέδου επικοινωνία μεταξύ πελατών και enterprise beans. Όταν ένα enterprise bean δημιουργείται ο πελάτης καλεί μεθόδους σε αυτό σαν να βρισκόταν στην ίδια εικονική μηχανή. Επειδή η Java EE αρχιτεκτονική παρέχει ρυθμιζόμενες υπηρεσίες, τα στοιχεία μιας Java EE εφαρμογής μπορούν να συμπεριφέρονται διαφορετικά ανάλογα με το που εγκαθίστανται. Για παράδειγμα, ένα enterprise bean μπορεί να έχει ρυθμίσεις ασφαλείας που του επιτρέπουν ως ένα σημείο πρόσβαση στα δεδομένα της Βάσης Δεδομένων σε ένα περιβάλλον εγκατάστασης και ως ένα διαφορετικό σημείο σε ένα διαφορετικό περιβάλλον εγκατάστασης.

Το container διαχειρίζεται, επίσης, μη ρυθμιζόμενες υπηρεσίες όπως τον κύκλο ζωής ενός enterprise bean και ενός servlet, την διατήρηση των δεδομένων και την πρόσβαση σε APIs της Java EE πλατφόρμας. Παρόλο που η διατήρηση δεδομένων δεν είναι μια ελεγχόμενη υπηρεσία, η Java EE αρχιτεκτονική επιτρέπει την συμπερίληψη κατάλληλου κώδικα στην υλοποίηση του enterprise bean όταν χρειάζεται περισσότερος έλεγχος από αυτόν που εξορισμού παρέχει το container (container managed persistence). Για παράδειγμα, μπορούν να υλοποιηθούν από τον προγραμματιστή (bean-managed persistence) μέθοδοι αναζήτησης (finder) ή να δημιουργηθεί μια προσαρμοσμένη προσωρινή αποθήκευση δεδομένων.

4.5.2. Java EE - Τύποι container

Υπάρχει μία διαδικασία εγκατάστασης (deployment) της εφαρμογής μας σε Container. Τα containers διακρίνονται σε :

- Enterprise JavaBeans (EJB) container, το container αυτό διαχειρίζεται την εκτέλεση των enterprise beans τα οποία εκτελούνται απ τον application server.
- Web container, το οποίο διαχειρίζεται την εκτέλεση των JSP σελίδων. Και αυτά εκτελούνται στον application server.
- Container Client, τα στοιχεία αυτά εκτελούνται στον client.
- Applet container, διαχειρίζεται την εκτέλεση των applets. Αποτελείται από ένα πρόγραμμα πλοήγησης και ένα Java Plug-in που εκτελούνται στον client και όχι στον server.

4.5.3. Web Υπηρεσίες (web services)

Οι web υπηρεσίες είναι εφαρμογές επιχειρησιακές που χρησιμοποιούν open source πρωτόκολλα μεταφοράς βασισμένα στο XML (eXtensible Markup Language – γλώσσα σήμανσης) για την ανταλλαγή δεδομένων καλώντας clients. Η πλατφόρμα της Java EE παρέχει τα api xml και τα εργαλεία που χρειάζεται ένας προγραμματιστής, για να σχεδιάσει και να αναπτύξει γρήγορα web services και clients συνδυάζοντας τα προηγούμενα με άλλες web services (υπηρεσίες ιστού) που εκτελούνται στον client η βασίζονται σε πλατφόρμες Java.

Για να γράψει ένας προγραμματιστής web services και client μέσω του API JAVA EE XML, το μόνο που χρειάζεται είναι να περάσει τα δεδομένα στις παραμέτρους, όταν καλεί τις μεθόδους και έπειτα να επεξεργαστεί τα δεδομένα που παίρνει ως απάντηση. Όσον αφορά την μεταφορά εγγράφων δεν απαιτείται προγραμματισμός χαμηλού επιπέδου, γιατί μέσω του API XML δημιουργείται μία ροή μηνυμάτων, που στέλνεται μέσω τυποποιημένων πρωτοκόλλων μεταφοράς XML.



Εικόνα: Σχήμα σχέσης XML - CLIENT - WEB SERVICES

Η μετάφραση των δεδομένων σε μια τυποποιημένη μορφή μηνυμάτων που βασίζεται στην υπηρεσία XML, είναι αυτό που καθιστά τις εφαρμογές αυτές διαλειτουργικές. Αυτό δεν σημαίνει απαραίτητα ότι τα δεδομένα που μεταφέρονται περιλαμβάνουν XML tags, τα δεδομένα αυτά μπορεί να είναι κείμενα, εικόνες, βίντεο, χάρτες, αρχεία προγραμμάτων κ.α

4.6. XML (Extensible Markup Language)

Η XML (αγγλ. αρκτ. από το eXtensible Markup Language) είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Ορίζεται, κυρίως, στην

Σχεδιασμός, Ανάπτυξη και Εφαρμογή Προηγμένου Λογισμικού
Διεπαφής χρηστών διαδικτυακών συζητήσεων (webChat), με τη
χρήση της γλώσσας Java, για το σκοπό ανίχνευσης λέξεων της
φυσικής γλώσσας που μαρτυρούν θυματοποίηση (Bullying).

προδιαγραφή XML 1.0 (XML 1.0 Specification), που δημιούργησε ο διεθνής οργανισμός προτύπων W3C (World Wide Web Consortium), αλλά και σε διάφορες άλλες σχετικές προδιαγραφές ανοιχτών προτύπων.

Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο Διαδίκτυο. Είναι μία μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη Unicode για όλες τις γλώσσες του κόσμου. Αν και η σχεδίαση της XML εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν για παράδειγμα στις υπηρεσίες ιστού.

Υπάρχει μία ποικιλία διεπαφών προγραμματισμού εφαρμογών, που μπορούν να χρησιμοποιούν οι προγραμματιστές, για να προσπελαίνουν δεδομένα XML, αλλά και διάφορα συστήματα σχημάτων XML, τα οποία είναι σχεδιασμένα για να βοηθούν στον ορισμό γλωσσών, που προκύπτουν από την XML.

Έως το 2009, έχουν αναπτυχθεί εκατοντάδες γλώσσες που βασίζονται στην XML, συμπεριλαμβανομένων του RSS, του SOAP και της XHTML. Προεπιλεγμένες κωδικοποιήσεις βασισμένες στην XML, υπάρχουν για τις περισσότερες σουίτες εφαρμογών γραφείου, συμπεριλαμβανομένων του Microsoft Office (Office Open XML), του OpenOffice.org (OpenDocument) και του iWork της εταιρίας Apple. [14]

4.7. Βασική ορολογία της XML

Το περιεχόμενο της ενότητας αυτής δεν αποτελεί μία πλήρη και απόλυτη λίστα όλων των όρων που υπάρχουν στην γλώσσα XML. Είναι μία βασική εισαγωγή στους όρους που βρίσκουμε στην καθημερινή χρήση της.

- **Χαρακτήρας Unicode** : Σε ένα κείμενο είναι μία ακολουθία χαρακτήρων και σχεδόν κάθε χαρακτήρας Unicode εμφανίζεται σε ένα κείμενο XML.
- **Επεξεργαστής και εφαρμογή** : Είναι το λογισμικό που επεξεργάζεται ένα κείμενο XML. Ο επεξεργαστής δουλεύει για μία εφαρμογή και υπάρχουν συγκεκριμένες απαιτήσεις στο τι μπορεί να κάνει ένας επεξεργαστής XML. Ο επεξεργαστής αναφέρεται συχνά με τον αγγλικό όρο XML parser.
- **Σήμανση και Περιεχόμενο** : Σημαίνει ότι οι χαρακτήρες που απαρτίζουν ένα κείμενο xml, αποτελούν είτε την σήμανση είτε το περιεχομένο του. Η σήμανση και το περιεχομένο του μπορούν να διακριθούν, ύστερα από την εφαρμογή κάποιων απλών συντακτικών κανόνων. Όλα τα αλφαριθμητικά ξεκινούν με το χαρακτήρα "<" και τελειώνουν με το χαρακτήρα ">", ακόμα μπορεί να ξεκινούν με το χαρακτήρα "&" και να τελειώνουν με το χαρακτήρα ";". Ότι δεν ακολουθεί την προηγούμενη σήμανση, αποτελεί το περιεχόμενο ενός κειμένου XML.
- **Ετικέτα** : Ένα στοιχείο σήμανσης που ξεκινά με το χαρακτήρα "<" και καταλήγει στο χαρακτήρα ">". Υπάρχουν τρία είδη ετικέτας: ετικέτες-αρχής, για παράδειγμα <section>, ετικέτες-τέλους, για παράδειγμα </section>, και ετικέτες-χωρίς-περιεχόμενο, για παράδειγμα <line-break/>.
- **Στοιχείο** : Ένα λογικό απόσπασμα ενός κειμένου, που είτε ξεκινά με μία ετικέτα-αρχής και καταλήγει σε μία ετικέτα-τέλους, είτε αποτελείται μόνο από μία ετικέτα-χωρίς-περιεχόμενο. Οι χαρακτήρες που υπάρχουν, αν υπάρχουν, μεταξύ μιας ετικέτας-αρχής και μιας ετικέτας-τέλους, συνιστούν το περιεχόμενο του στοιχείου, το οποίο μπορεί να περιέχει σήμανση, συμπεριλαμβανομένων και άλλων στοιχείων, που ονομάζονται στοιχεία-παιδιά. Ένα παράδειγμα ενός στοιχείου είναι το <Greeting>Hello, world.</Greeting>. Ένα άλλο είναι το <line-break/>.

- **Χαρακτηριστικό** : Ένα στοιχείο σήμανσης που αποτελείται από ένα ζευγάρι όνομα/τιμή, το οποίο υπάρχει μέσα σε μία ετικέτα-αρχής ή σε μία ετικέτα-χωρίς-περιεχόμενο. Στο παράδειγμα παρακάτω, το στοιχείο `img` έχει δύο χαρακτηριστικά, τα `src` και `alt`: ``. Ένα άλλο παράδειγμα θα ήταν το `<step number="3">Connect A to B.</step>`, όπου το όνομα του χαρακτηριστικού είναι "number" και η τιμή του είναι "3".
- **Δήλωση XML** : Τα κείμενα XML μπορούν να αρχίζουν, με τη δήλωση κάποιων πληροφοριών σχετικών με αυτά, όπως στο ακόλουθο παράδειγμα:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Εικόνα : Παράδειγμα αρχικής δήλωσης XML

Το παρακάτω παράδειγμα είναι ένα πλήρες κείμενο XML που χρησιμοποιεί τις παραπάνω έννοιες και στοιχεία [14] :

```
<?xml version="1.0" encoding='UTF-8'?>
<painting>
  
  <caption>This is Raphael's "Foligno" Madonna, painted in
  <date>1511</date>-<date>1512</date>.</caption>
</painting>
```

Υπάρχουν πέντε στοιχεία σε αυτό το κείμενο του παραδείγματος: τα `painting`, `img`, `caption`, και δύο `date`. Τα στοιχεία `date`, είναι παιδιά του στοιχείου `caption`, το οποίο είναι παιδί του στοιχείου-ρίζας `painting`. Το στοιχείο `img` έχει δύο χαρακτηριστικά, τα `src` και `alt`.

4.8. WebSocket

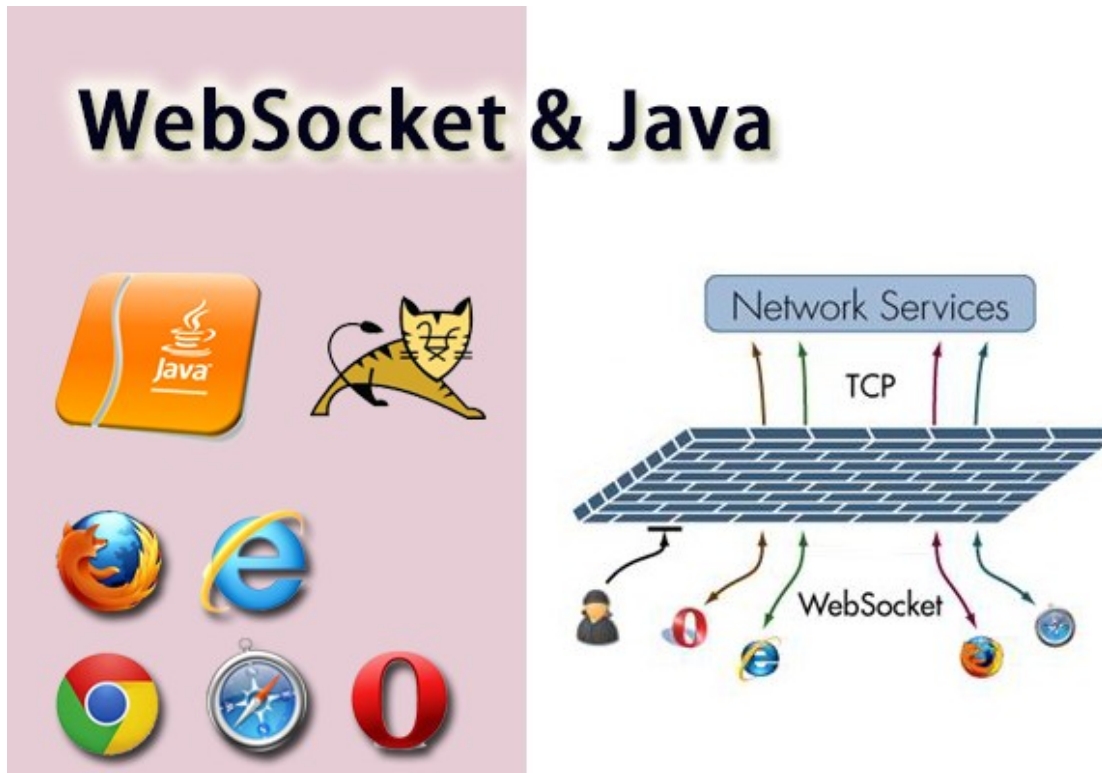
Το WebSocket είναι ένα πρωτόκολλο επικοινωνίας υπολογιστών το οποίο παρέχει κανάλια πλήρους αμφίδρομης επικοινωνίας μέσω μίας TCP σύνδεσης.

Το πρωτόκολλο WebSocket έγινε πρότυπο από τον οργανισμό IETF με τον κωδικό RFC 6455 το 2011.

Το πρωτόκολλο WebSocket υποστηρίζεται σχεδόν σε όλους τους κύριους φυλλομετρητές, όπως οι Google Chrome, Microsoft Edge, Internet Explorer, Firefox, Safari και Opera. Το WebSocket απαιτεί επίσης τις εφαρμογές κινητών στον διακομιστή να το υποστηρίζουν.

Το WebSocket είναι διαφορετικό πρωτόκολλο από το HTTP. Και τα 2 πρωτόκολλα βρίσκονται στο 7ο επίπεδο του Μοντέλου αναφοράς OSI και έτσι εξαρτώνται από το TCP στο επίπεδο 4. Παρ' ότι είναι διαφορετικά, το RFC 6455 δηλώνει ότι το WebSocket "είναι σχεδιασμένο να λειτουργεί πάνω από τις πόρτες 80 και 443 του HTTP αλλά και να υποστηρίζει HTTP proxies και

μεσάζοντες" κάνοντάς το έτσι συμβατό με το πρωτόκολλο HTTP. Για να επιτευχθεί η συμβατότητα με το HTTP, η χειραψία του WebSocket χρησιμοποιεί το HTTP Upgrade header για να αλλάξει το HTTP πρωτόκολλο σε WebSocket [15].



Εικόνα : Διαδικασία επικοινωνίας μέσω WebSocket

Σε μια εφαρμογή Java Websocket, ο διακομιστής δημοσιεύει ένα τελικό σημείο που ο client χρησιμοποιεί το URI του τελικού σημείου, για να συνδεθεί με τον διακομιστή (application server).

Το πρωτόκολλο WebSocket είναι συμμετρικό και μετά την συνδεσή του ανάμεσα στον client και στον server, μπορούν να στείλουν μηνύματα ο ένας στον άλλο ανα πάσα στιγμή. Συνήθως ο client συνδέεται μόνο με ένα διακομιστή (server) ενώ οι διακομιστές δέχονται συνδέσεις από πολλούς client.

Το πρωτοκολλο μεταφοράς WebSocket έχει δύο σημαντικά μέρη : handshake και data transfer. Ο client ξεκινά μέσω του handshake στέλνοντας ένα αίτημα σε ένα endpoint WebSocket χρησιμοποιώντας το URI του. Το handshake είναι συμβατό με την υπάρχουσα υποδομή σε HTTP.

Ένα παράδειγμα handshake είναι κάπως έτσι :

```
GET /path/to/websocket/endpoint HTTP/1.1
Host: localhost
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: xqBt3ImNzJbYqRINxEFlkg==
Origin: http://localhost
Sec-WebSocket-Version: 13
```

Σχε

Διεπίσης χρηστών διαδικτυακών συζητήσεων (webChat), με τη χρήση της γλώσσας Java, για το σκοπό ανίχνευσης λέξεων της φυσικής γλώσσας που μαρτυρούν θυματοποίηση (Bullying).

Παράδειγμα handshake από τον server σε απάντηση του client, είναι κάπως έτσι :

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: K7DJLdLooIwIG/M0pnWFB3y3FE8=
```

Παράδειγμα handshake από την μεριά του διακομιστή

Ο διακομιστής στο παραπάνω παράδειγμα, εφαρμόζει μια γνωστή λειτουργία στην κεφαλίδα Sec-WebSocket-Key για να δημιουργήσει την τιμή της κεφαλίδας Sec-WebSocket-Accept. Ο υπολογιστής client εφαρμόζει την ίδια ενέργεια στην τιμή της κεφαλίδας Sec-WebSocket-Key και η σύνδεση δημιουργείται με επιτυχία αν το αποτέλεσμα αντιστοιχεί στην τιμή που έλαβε από τον διακομιστή. Ο client και ο server μπορούν να στέλνουν μεταξύ τους μηνύματα μετά από ένα επιτυχημένο handshake. Επίσης, το WebSocket υποστηρίζει μηνύματα κειμένου (κωδικοποιημένα ως UTF-8) και δυαδικά μηνύματα.

Τέλος ένα παράδειγμα των τελικών σημείων του WebSocket αντιπροσωπεύεται από το URI και έχει την ακόλουθη μορφή.

```
ws://host:port/path?query
wss://host:port/path?query
```

Εικόνα WebSocket URI

4.9. MAVEN

Το Maven είναι ένα εργαλείο αυτοματοποίησης κατασκευής έργων που είναι φτιαγμένα σε Java. Το Maven είναι ένας συσσωρευτής γνώσης, στην αρχή ξεκίνησε ως μια προσπάθεια απλοποίησης των διαδικασιών κατασκευής για το Jakarta Project. Υπήρχε η ανάγκη για ένα τυποποιημένο τρόπο για την κατασκευή ενός έργου σε Java και η δημοσίευσή των πληροφοριών και ο διαμοιρασμός των JAR (Java archive – είναι ένα πακέτο αρχείου με συγκεκριμένο format) σε διάφορα έργα φτιαγμένα σε Java.

Σχεδιασμός, Ανάπτυξη και Εφαρμογή Προηγμένου Λογισμικού
Διεπαφής χρηστών διαδικτυακών συζητήσεων (webChat), με τη
χρήση της γλώσσας Java, για το σκοπό ανίχνευσης λέξεων της
φυσικής γλώσσας που μαρτυρούν θυματοποίηση (Bullying).

Το Maven ασχολείται με δύο πτυχές του λογισμικού κατασκευής ενός έργου γραμμένο σε Java, περιγράφει τον τρόπο που το λογισμικό κατασκευάστηκε και τις εξαρτήσεις του. Σε αντίθεση με προηγούμενα εργαλεία όπως το Apache Ant, χρησιμοποιεί συμβάσεις για τη δημιουργία του έργου και των εξαιρέσεων που πρέπει να καταγραφούν. Ένα αρχείο XML περιγράφει το έργο του λογισμικού που κατασκευάζεται, τις εξαρτήσεις, τη σειρά κατασκευής και τους καταλόγους. Έρχεται με προκαθορισμένους στόχους για την εκτέλεση ορισμένων σαφώς καθορισμένων εργασιών, όπως η σύνταξη κώδικα και το packaging του κωδικά μας.

Το Maven δυναμικά κατεβάζει βιβλιοθήκες Java και Plug-ins από ένα ή περισσότερα αποθετήρια, όπως το Maven 2 Central Repository και τα αποθηκεύει σε μια τοπική μνήμη cache. Αυτή η τοπική μνήμη cache μπορεί να ενημερωθεί από artifacts που δημιούργησαν άλλοι, επίσης μπορούν να ενημερωθούν.

Το Maven είναι κατασκευασμένο χρησιμοποιώντας μια αρχιτεκτονική που βασίζεται σε προσθήκες, η οποία του επιτρέπει να κάνει χρήση οποιασδήποτε εφαρμογής που ελέγχεται μέσω πρότυπης εισαγωγής. Θεωρητικά, αυτό θα επέτρεπε σε οποιονδήποτε να γράψει plugins για διασύνδεση με εργαλεία δημιουργίας (μεταγλωττιστές, εργαλεία δοκιμής μονάδων κ.λπ.) για οποιαδήποτε άλλη γλώσσα. Στην πραγματικότητα, η υποστήριξη και η χρήση για άλλες γλώσσες εκτός από την Java ήταν ελάχιστη. Ένα plugin για το .NET Framework υπάρχει και διατηρείται και ένα εγγενές plugin C / C ++ διατηρείται για το Maven 2.

Εναλλακτικές τεχνολογίες όπως το Gradle και το sbt ως εργαλεία δημιουργίας δεν βασίζονται σε XML, αλλά διατηρούν τις βασικές έννοιες που εισήγαγε η Maven. Με τον Apache Ivy, αναπτύχθηκε εξειδικευμένος διαχειριστής εξάρτησης, ο οποίος υποστηρίζει επίσης αποθετήρια Maven.

Στόχος του Maven είναι να επιτρέψει σε ένα προγραμματιστή να κατανοήσει την πλήρη κατάσταση του να αναπτύσει ένα λογισμικό σε μικρό χρονικό διάστημα. Για να επιτευχθεί αυτό το Maven χρησιμοποιεί κάποιους τομείς :

- Κάνει την διαδικασία ευκολότερη κατασκευής ενός έργου Java.
- Παρέχει ένα ενιαίο σύστημα κατασκευής ενός έργου Java.
- Παρέχει σημαντικές πληροφορίες για το έργο που φτιάχνουμε σε Java.
- Παρέχει οδηγίες για την ανάπτυξη βέλτιστων πρακτικών.

4.10. Firebase

Η Firebase είναι μία πλατφόρμα ανάπτυξης εφαρμογών και ιστοσελίδων που παρέχει στους προγραμματιστές εργαλεία και υπηρεσίες, που τους βοηθούν να αναπτύξουν εφαρμογές υψηλής ποιότητας, να αυξήσουν την βάση των χρηστών τους, και να κερδίσουν περισσότερα χρήματα.



4.10.1. Η ιστορία της Firebase

Η Firebase αναπτύχθηκε από την Envolv, μια προηγούμενη StartUp εταιρεία που ίδρυσε ο James Tamplin και ο Andrew Lee το 2011. Παρείχαν ένα API στους προγραμματιστές, που επέτρεπε την online δυνατότητα συνομιλίας σε ιστοσελίδα (web chat). Μετά την απελευθέρωση της

υπηρεσίας συνομιλίας, οι Tamplin και Lee διαπίστωσαν ότι χρησιμοποιείται για να διαβιβάζει δεδομένα εφαρμογών που δεν ήταν μηνύματα συνομιλίας. Οι προγραμματιστές χρησιμοποίησαν το Envolv για να συγχρονίσουν δεδομένα εφαρμογών, όπως η κατάσταση του παιχνιδιού, σε πραγματικό χρόνο των χρηστών τους. Οι Tamplin και Lee αποφάσισαν να χωρίσουν το σύστημα συνομιλίας και την αρχιτεκτονική πραγματικού χρόνου που την τροφοδοτούσαν. Ίδρυσαν τη Firebase ως ξεχωριστή εταιρεία τον Απρίλιο του 2012.

Η Firebase Inc. αύξησε τη χρηματοδότηση το Μάιο του 2012. Η εταιρεία αύξησε περαιτέρω τη χρηματοδότηση της Σειράς A τον Ιούνιο του 2013. Τον Οκτώβριο του 2014, η Firebase εξαγοράστηκε από την Google . Τον Οκτώβριο του 2015, η Google αγόρασε το Divshot για να το συγχωνεύσει με την ομάδα Firebase. Από την εξαγορά, το Firebase έχει αναπτυχθεί μέσα στο Google και έχει επεκτείνει τις υπηρεσίες του για να γίνει μια ενοποιημένη πλατφόρμα για προγραμματιστές για κινητά. Η Firebase ενσωματώνεται τώρα με διάφορες άλλες υπηρεσίες της Google για να προσφέρει ευρύτερα προϊόντα και κλίμακα για προγραμματιστές. Τον Ιανουάριο του 2017, η Google εξαγόρασε το Fabric και Crashlytics από το Twitter για να συμμετάσχει σε αυτές τις υπηρεσίες στην ομάδα της Firebase. Το Firebase ξεκίνησε το Cloud Firestore, μια βάση δεδομένων εγγράφων, τον Οκτώβριο του 2017 [16].

4.10.2. Υπηρεσίες της Firebase

Η Firebase έχει μία πληθώρα υπηρεσιών, οι οποίες διευκολύνουν τους προγραμματιστές και είναι οι εξής:



Analytics

Σχεδιασμός, Ανάπτυξη και Εφαρμογή Προηγμένου Λογισμικού Διεπαφής χρηστών διαδικτυακών συζητήσεων (webChat), με τη χρήση της γλώσσας Java, για το σκοπό ανίχνευσης λέξεων της φυσικής γλώσσας που μαρτυρούν θυματοποίηση (Bullying).

Firestore Analytics

- Το Firestore Analytics είναι μια δωρεάν λύση μέτρησης πληροφοριών, που μετράει την χρήση της εφαρμογής από τον χρήστη.

Develop (Ανάπτυξη)

Firestore Cloud Messaging

- Ανάπτυξη του Firestore Cloud Messaging, παλαιότερα ήταν γνωστό με την ονομασία ως Google Cloud Messaging (FCM). Είναι μία λύση για πολλαπλές πλατφόρμες μέσω μηνυμάτων και ειδοποιήσεις για εφαρμογές Android, iOS και web, οι οποίες από το 2016 μπορούν να χρησιμοποιηθούν χωρίς οικονομικό κόστος.

Firestore Auth

- Το Firestore Auth είναι μια υπηρεσία που μπορεί να πιστοποιήσει τους χρήστες, χρησιμοποιώντας μόνο των κωδικό του client-side. Υποστηρίζει παρόχους κοινωνικών δικτύων όπως Facebook, GitHub, Twitter και Google (και Παιχνίδια Google Play). Επιπλέον, περιλαμβάνει ένα σύστημα διαχείρισης χρηστών, στο οποίο οι προγραμματιστές μπορούν να ενεργοποιήσουν τον έλεγχο ταυτότητας χρήστη με σύνδεση μέσω ηλεκτρονικού ταχυδρομείου και κωδικού πρόσβασης που έχει αποθηκευτεί μέσω της Firestore.

Firestore Database (Βάση δεδομένων σε πραγματικό χρόνο)

- Η Firestore παρέχει μια βάση δεδομένων σε πραγματικό χρόνο και μια backend ως υπηρεσία. Η υπηρεσία παρέχει στους προγραμματιστές εφαρμογών ένα API που επιτρέπει στα δεδομένα εφαρμογών να συγχρονίζονται μεταξύ των πελατών και να αποθηκεύονται στο νέφος του Firestore. Η εταιρεία παρέχει βιβλιοθήκες πελατών που επιτρέπουν την ενσωμάτωση με εφαρμογές Android, iOS, JavaScript, Java, Objective-C, Swift και Node.js. Η βάση δεδομένων είναι επίσης προσβάσιμη μέσω ενός API REST και συνδέσεων για διάφορα πλαίσια JavaScript όπως AngularJS, React, Ember.js και Backbone.js. Το API REST χρησιμοποιεί το πρωτόκολλο Server-Sent Events, το οποίο είναι ένα API για τη δημιουργία συνδέσεων HTTP για τη λήψη ειδοποιήσεων push από ένα διακομιστή. Οι προγραμματιστές που χρησιμοποιούν τη βάση δεδομένων σε πραγματικό χρόνο μπορούν να εξασφαλίσουν τα δεδομένα τους, χρησιμοποιώντας τους κανόνες ασφαλείας που εφαρμόζει η εταιρεία από πλευράς διακομιστή. Το Cloud Firestore, το οποίο είναι η επόμενη γενιά της Firestore της Realtime Database, κυκλοφόρησε για beta χρήση.

Firestore Storage (Αποθήκευση Firestore)

- Η αποθήκευση Firestore παρέχει ασφαλείς λήψεις και λήψεις αρχείων για εφαρμογές Firestore, ανεξάρτητα από την ποιότητα του δικτύου. Ο προγραμματιστής μπορεί να το χρησιμοποιήσει για την αποθήκευση εικόνων, ήχου, βίντεο ή άλλου περιεχομένου που δημιουργεί ο χρήστης. Η αποθήκευση Firestore υποστηρίζεται από το Google Cloud Storage.

Firestore Hosting

- Το Firestore Hosting είναι μια στατική και δυναμική υπηρεσία φιλοξενίας ιστοσελίδων που ξεκίνησε στις 13 Μαΐου 2014. Υποστηρίζει φιλοξενία στατικών αρχείων όπως CSS, HTML, JavaScript και άλλα αρχεία, καθώς και υποστήριξη μέσω των λειτουργιών Cloud. Η υπηρεσία παρέχει αρχεία μέσω δικτύου παροχής περιεχομένου (CDN) μέσω της κρυπτογράφησης HTTP Secure (HTTPS) και της κρυπτογράφησης Secure Sockets Layer (SSL). Η Firestore συνεργάζεται με το Fast, ένα CDN, για να παρέχει το υποστηρικτικό CDN Firestore Hosting. Η εταιρεία δηλώνει ότι η Firestore Hosting αναπτύχθηκε από αιτήματα πελατών. οι προγραμματιστές χρησιμοποιούσαν τη Firestore για τη βάση δεδομένων σε πραγματικό χρόνο, αλλά χρειαζόνταν ένα μέρος για να φιλοξενήσουν το περιεχόμενό τους.

ML Kit

- Το ML Kit είναι ένα κινητό σύστημα εκμάθησης μηχανών για προγραμματιστές που ξεκίνησε στις 8 Μαΐου 2018 σε beta κατά τη διάρκεια του Google I / O 2018. Τα εργαλεία AP API του ML Kit διαθέτουν μια ποικιλία χαρακτηριστικών όπως αναγνώριση κειμένου, ανίχνευση προσώπων, σάρωση γραμμικών κωδικών, ετικετών εικόνων και αναγνώριση ορόσημων. Αυτήν τη στιγμή διατίθεται για προγραμματιστές iOS ή Android. Επίσης, μπορείτε να εισαγάγετε τα δικά σας μοντέλα TensorFlow Lite, αν τα δεδομένα API δεν επαρκούν. Τα API μπορούν να χρησιμοποιηθούν σε συσκευή ή σε cloud.

Stability (Σταθερότητα)

Crashlytics

- Η Αναφορά σφαλμάτων δημιουργεί λεπτομερείς αναφορές για τα σφάλματα στην εφαρμογή. Τα σφάλματα ομαδοποιούνται σε ομάδες παρόμοιων ιχνών στοιβάς και ταξινομούνται βάσει της σοβαρότητας των επιπτώσεων στους χρήστες εφαρμογών. Εκτός από τις αυτόματες αναφορές, ο προγραμματιστής μπορεί να καταγράψει προσαρμοσμένα συμβάντα για να βοηθήσει να καταγράψει τα βήματα που οδηγούν σε σύγκρουση. Πριν από την απόκτηση του Crashlytics, η Firestore χρησιμοποίησε τη δική της Αναφορά Crash Firestore.

Performance (Απόδοση)

- Η απόδοση της Firestore παρέχει πληροφορίες σχετικά με την απόδοση μιας εφαρμογής και τις λανθάνουσες περιόδους λειτουργίας των χρηστών της εφαρμογής (crash reports).

Firestore Test Lab για Android και iOS

- Το Firestore Test Lab για Android και iOS παρέχει υποδομή βασισμένη στο σύννεφο (cloud) για δοκιμή εφαρμογών Android και iOS. Με μία ενέργεια, οι προγραμματιστές μπορούν να ξεκινήσουν τη δοκιμή των εφαρμογών τους σε μια μεγάλη ποικιλία συσκευών και διαμορφώσεων συσκευών. Τα αποτελέσματα των δοκιμών - συμπεριλαμβανομένων των αρχείων καταγραφής, των βίντεο και των στιγμιότυπων οθόνης - διατίθενται στο έργο στην κονσόλα Firestore. Ακόμη και αν ένας προγραμματιστής δεν έχει γράψει κανένα κώδικα δοκιμής για την εφαρμογή τους, το Test Lab μπορεί να ασκήσει αυτόματα την

εφαρμογή, αναζητώντας crashes. Το Test Lab για iOS βρίσκεται αυτή τη στιγμή στην έκδοση beta.

Grow

Firestore Notifications (Ειδοποιήσεις της Firebase)

- Η ειδοποίηση Firebase είναι μια υπηρεσία που επιτρέπει τη δωρεάν ειδοποίηση χρήστη για προγραμματιστές εφαρμογών για κινητά τηλέφωνα.

Firestore App Indexing

- Η ευρετηρίαση του Firestore App, παλαιότερα η ευρετηρίαση του Google App, παίρνει μια εφαρμογή στην Αναζήτηση Google. Η προσθήκη της ευρετηρίασης εφαρμογών προάγει και τους δύο τύπους αποτελεσμάτων εφαρμογής στην Αναζήτηση Google και παρέχει επίσης αυτόματες συμπληρώσεις ερωτήσεων.

Firestore Dynamic Links (Δυναμικές συνδέσεις Firebase)

- Οι Firestore Dynamic Links είναι “έξυπνες” διευθύνσεις URL που αλλάζουν δυναμικά τη συμπεριφορά για να παρέχουν την καλύτερη εμπειρία ανάμεσα σε διάφορες πλατφόρμες (ιστοσελίδα / iOS / Android) καθώς και σύνδεση με την εφαρμογή.

Firestore Invites

- Το Firestore Invites είναι μια λύση για πολλαπλές πλατφόρμες, για την αποστολή εξατομικευμένων προσκλήσεων μέσω ηλεκτρονικού ταχυδρομείου και SMS, των χρηστών που την χρησιμοποιούν και μετράει τις προσκλήσεις προς τους χρήστες αυτούς.

Firestore Remote Config

- Το Firestore Remote Config είναι μια υπηρεσία cloud που επιτρέπει στους προγραμματιστές να αλλάζουν τη συμπεριφορά και την εμφάνιση των εφαρμογών τους, χωρίς να απαιτούν από τους χρήστες να κάνουν λήψη ενημέρωσης της εφαρμογής.

Adwords

- Το Adwords είναι η υπηρεσία διαφήμισης μέσω διαδικτύου της Google, η οποία ενσωματώνεται ώστε να επιτρέπει στους προγραμματιστές να βρίσκουν <<στοχεύουν>> χρήστες χρησιμοποιώντας το Firestore Analytics.

Earn

- Κερδίστε μέσα από την υπηρεσία διαφήμισης admob και στο πεδίο εφαρμογής μάρκετινγκ.

Admob

- Το Admob είναι ένα προϊόν της Google που ενσωματώνεται στο κοινό της Firestore.

4.11. Git (λογισμικό)

Το Git (/git/) είναι ένα σύστημα ελέγχου εκδόσεων (λέγεται και σύστημα ελέγχου αναθεωρήσεων ή σύστημα ελέγχου πηγαιού κώδικα) με έμφαση στην ταχύτητα, στην ακεραιότητα των δεδομένων και στην υποστήριξη για κατανεμημένες μη γραμμικές ροές εργασίας. Το Git σχεδιάστηκε και αναπτύχθηκε αρχικά από τον Λίνους Τόρβαλντς για τη ανάπτυξη του πυρήνα Linux το 2005 και έχει γίνει από τότε το πιο διαδεδομένο σύστημα ελέγχου εκδόσεων για ανάπτυξη λογισμικού.

Όπως τα περισσότερα άλλα κατανεμημένα συστήματα ελέγχου εκδόσεων/αναθεωρήσεων και αντίθετα με τα περισσότερα συστήματα πελάτη-διακομιστή, κάθε κατάλογος εργασίας του Git είναι ένα ολοκληρωμένο αποθετήριο λογισμικού με πλήρες ιστορικό και δυνατότητες πλήρους παρακολούθησης της έκδοσης, ανεξάρτητα από την πρόσβαση δικτύου ή ενός κεντρικού διακομιστή. Όπως ο πυρήνας Λίνουξ, το Git είναι Ελεύθερο λογισμικό που διανέμεται κάτω από τους όρους της έκδοσης 2 της Γενικής Άδειας Δημόσιας Χρήσης GNU.

4.11.1 Η ιστορία του Git

Η ανάπτυξη του Git ξεκίνησε αφού πολλοί προγραμματιστές του πυρήνα Linux σταμάτησαν την πρόσβαση στο BitKeeper, ένα ιδιοταγές (δηλαδή μη ανοικτού κώδικα) σύστημα ελέγχου εκδόσεων που χρησιμοποιούταν προηγουμένως για τη διατήρηση του έργου. Ο κάτοχος των πνευματικών δικαιωμάτων του BitKeeper, Larry McVoy, είχε αποσύρει την ελεύθερη χρήση του προϊόντος αφού ισχυρίστηκε ότι ο Andrew Tridgell είχε εφαρμόσει αντίστροφη μηχανική στα πρωτόκολλα του BitKeeper.

Ο Τόρβαλντς ήθελε ένα κατανεμημένο σύστημα που θα μπορούσε να χρησιμοποιηθεί σαν το BitKeeper, αλλά κανένα από τα διαθέσιμα ελεύθερα συστήματα δεν κάλυπτε τις ανάγκες του, ιδιαίτερα στην απόδοση. Ο Τόρβαλντς πήρε ένα παράδειγμα ενός συστήματος SCM που χρειαζότανε 30 δευτερόλεπτα για να εφαρμόσει μια διόρθωση και να ενημερώσει όλα τα συνδεδεμένα μεταδεδομένα και σημείωσε ότι αυτό δεν μπορούσε να ανταποκριθεί στις ανάγκες της ανάπτυξης του πυρήνα Linux, όπου ο συγχρονισμός με τους συνεργάτες του μπορούσε να απαιτήσει 250 τέτοιες ενέργειες τη φορά. Ήθελε η διόρθωση να κρατήσει 3 δευτερόλεπτα, και είχε αρκετά άλλα κριτήρια σχεδίασης κατά νου:

- να πάρει ταυτόχρονο σύστημα εκδόσεων (CVS) ως ένα παράδειγμα του τι να «μην» κάνει· αν υπήρχε αμφιβολία, να γίνει η ακριβώς αντίθετη απόφαση
- να υποστηρίξει μια διανεμόμενη, παρόμοια με το BitKeeper ροή εργασίας
- να έχει πολύ ισχυρές εγγυήσεις ενάντια σε αλλοίωση, είτε τυχαία είτε κακόβουλη.

Αυτά τα τρία κριτήρια απάλειψαν κάθε σύστημα ελέγχου εκδόσεων που υπήρχε τότε, εκτός από το Monotone. Εξετάζοντας την απόδοσή του, το απέρριψε και αυτό. Έτσι αμέσως μετά την έκδοση του πυρήνα Linux 2.6.12-rc2, ο Τόρβαλντς άρχισε να γράφει ένα δικό του.

Ο Τόρβαλντς δικαιολόγησε το όνομα git, που στην βρετανική αργκό σημαίνει κάτι σαν «ανεπιθύμητος», λέγοντας: «Είμαι ένα φαντασμένο άτομο και ονομάζω όλα τα έργα μου σύμφωνα με τον εαυτό μου. Πρώτα ήταν το Λίνουξ, τώρα το git». Άλλωστε και η σελίδα του εγχειριδίου (man page) περιγράφει το git ως «τον ηλίθιο εντοπιστή περιεχομένου».

Η ανάπτυξη του Git ξεκίνησε στις 3 Απριλίου 2005, ανακοινώθηκε στις 6 και έγινε αυτοφιλοξενοούμενο (self-hosting) από τις 7 Απριλίου. Η πρώτη συγχώνευση των πολλαπλών κλάδων έγινε στις 18 Απριλίου. Ο Τόρβαλντς πέτυχε τους στόχους απόδοσης. Στις 29 Απριλίου, το νεότευκτο Git επιδόθηκε στην εφαρμογή διορθώσεων στο δέντρο του πυρήνα του Linux kernel με ένα ρυθμό 6.7 ανά δευτερόλεπτο. Στις 16 Ιουνίου το Git διαχειρίστηκε την έκδοση 2.6.12 του πυρήνα. Ο Τόρβαλντς παρέδωσε τη συντήρηση στις 26 Ιουλίου 2005 στον Junio Hamano, έναν βασικό συνεργάτη του έργου. Ο Hamano ήταν υπεύθυνος για την έκδοση 1.0 στις 21 Δεκεμβρίου 2005 και από τότε παραμένει ο κύριος υπεύθυνος για την συντήρησή του [17].

4.11.2. Τα Χαρακτηριστικά του Git

Η σχεδίαση του Git εμπνεύστηκε από τα BitKeeper και Monotone. Το Git σχεδιάστηκε αρχικά ως μια μηχανή συστήματος ελέγχου σε έκδοση χαμηλού επιπέδου πάνω στην οποία άλλοι μπορούν να γράψουν προγράμματα προσκηνίου, όπως Cogito ή StGIT. Το έργο του πυρήνα του Git έχει γίνει από τότε μια πλήρης έκδοση συστήματος ελέγχου άμεσα χρησιμοποιήσιμη. Αν και επηρεάστηκε έντονα από το BitKeeper, ο Τόρβαλντς προσπάθησε εκούσια να αποφύγει συμβατικές προσεγγίσεις, που οδηγούν σε έναν μοναδικό σχεδιασμό.

Χαρακτηριστικά

Η σχεδίαση του Git είναι μια σύνθεση της εμπειρίας του Τόρβαλντς με το Λίνουξ στη συντήρηση ενός μεγάλου καταμεμημένου έργου ανάπτυξης, μαζί με την βαθιά του γνώση της απόδοσης του συστήματος των αρχείων που αποκτήθηκε από το ίδιο έργο και την επείγουσα ανάγκη παραγωγής ενός λειτουργικού συστήματος. Αυτές οι επιρροές οδήγησαν στις παρακάτω επιλογές υλοποίησης:

Ισχυρή υποστήριξη για μη γραμμική ανάπτυξη

Το Git υποστηρίζει γρήγορη κλωνοποίηση και συγχώνευση και περιλαμβάνει ειδικά εργαλεία οπτικοποίησης και περιήγησης του ιστορικού της μη γραμμικής ανάπτυξης. Μια βασική υπόθεση στο Git είναι ότι μια αλλαγή θα συγχωνεύεται πιο συχνά από ότι γράφεται, καθώς θα περνάει από διάφορες αναθεωρήσεις. Οι κλάδοι στο git είναι πολύ ελαφρείς: Ένας κλάδος στο git είναι απλά μια αναφορά σε μια απλή υποβολή. Με τις γονικές υποβολές, θα πρέπει να οικοδομείται μια πλήρης δομή του κλάδου.

Διανεμόμενη ανάπτυξη

Όπως τα Darcs, BitKeeper, Mercurial, SVK, Bazaar και Monotone, το Git δίνει σε κάθε προγραμματιστή ένα τοπικό αντίγραφο του συνολικού ιστορικού της ανάπτυξης και οι αλλαγές αντιγράφονται από ένα τέτοιο αποθετήριο σε ένα άλλο αποθετήριο. Αυτές οι αλλαγές εισάγονται ως πρόσθετοι κλάδοι ανάπτυξης και μπορούν να συγχωνευθούν με τον ίδιο τρόπο όπως ένας τοπικά αναπτυσσόμενος κλάδος.

Συμβατότητα με τα υπάρχοντα συστήματα/πρωτόκολλα

Τα αποθετήρια μπορούν να δημοσιευτούν μέσα από τα πρωτόκολλα HTTP, FTP, rsync, ή Git μέσα από είτε μια απλή υποδοχή (plain socket), ή SSH. Το Git έχει επίσης μια προσομοίωση διακομιστή CVS, που ενεργοποιεί τη χρήση των υφιστάμενων πελατών CVS και των προσθέτων IDE για να προσπελάσουν τα αποθετήρια Git. Τα αποθετήρια Subversion και svk μπορούν να χρησιμοποιηθούν άμεσα με git-svn.

Αποτελεσματικό χειρισμό μεγάλων έργων

Ο Τόρβαλντς έχει περιγράψει το Git ότι είναι πολύ γρήγορο και κλιμακώσιμο. Οι δοκιμές απόδοσης που έγιναν από το Mozilla έδειξαν ότι είναι μια τάξη μεγέθους πιο γρήγορα από κάποια άλλα συστήματα ελέγχου έκδοσης και ότι η μεταφορά του ιστορικού της έκδοσης από ένα τοπικά αποθηκευμένο αποθετήριο μπορεί να είναι εκατό φορές πιο γρήγορο από την μεταφορά του από απομακρυσμένο διακομιστή.

Κρυπτογραφική πιστοποίηση του ιστορικού

Το ιστορικό του Git αποθηκεύεται κατά τέτοιο τρόπο που το αναγνωριστικό μιας συγκεκριμένης έκδοσης (μιας υποβολής με όρους Git) εξαρτάται από το πλήρες ιστορικό ανάπτυξης που οδηγεί σε αυτήν την υποβολή. Μόλις δημοσιευτεί, δεν μπορεί να αλλαχθεί σε παλιότερες εκδόσεις χωρία αυτό να σημειωθεί. Η δομή είναι παρόμοια με ένα δένδρο Merkle, αλλά με πρόσθετα δεδομένα στους κόμβους καθώς και στα φύλλα.[23] (Τα Mercurial και Monotone έχουν επίσης αυτήν την ιδιότητα.)

Σχεδιασμός με βάση την εργαλειοθήκη

Το Git σχεδιάστηκε ως ένα σύνολο προγραμμάτων γραμμένων σε C και σε ένα αριθμό σεναρίων κελύφους που παρέχουν συσκευασίες για αυτά τα προγράμματα.[24] Αν και τα περισσότερα από αυτά τα σεναρία έχουν από τότε ξαναγραφτεί σε C για καλύτερη ταχύτητα και φορητότητα, ο σχεδιασμός παραμένει και είναι εύκολο να δημιουργηθεί μια αλυσίδα των συστατικών μαζί.[25]

Προσθέσιμες στρατηγικές συγχώνευσης

Ως μέρος του σχεδιασμού της εργαλειοθήκης, το Git έχει ένα καλοσχεδιασμένο πρότυπο ατελούς συγχώνευσης και έχει πολλούς αλγόριθμους για συμπλήρωσή του, καταλήγοντας στην πληροφόρηση του χρήστη ότι δεν μπορεί να ολοκληρώσει τη συγχώνευση αυτόματα και απαιτείται χειροκίνητη επεξεργασία.

Συσσωρεύει σκουπίδια (Garbage) εκτός και συλλεγούν αλλού

Η μεταίωση λειτουργιών ή η ανάκληση αλλαγών θα αφήσει άχρηστα αιωρούμενα αντικείμενα στη βάση δεδομένων. Αυτά είναι γενικά ένα μικρό κλάσμα του συνεχώς αυξανόμενου ιστορικού των επιθυμητών αντικειμένων. Το Git θα εκτελέσει αυτόματα τη συλλογή σκουπιδιών, όταν έχουν δημιουργηθεί αρκετά χαλαρά αντικείμενα στο αποθετήριο. Η συλλογή σκουπιδιών μπορεί να κληθεί ρητά χρησιμοποιώντας το `git gc --prune`.

Περιοδική ρητή συσκευασία αντικειμένων

Το Git αποθηκεύει κάθε νεοδημιουργούμενο αντικείμενο ως ξεχωριστό αρχείο. Αν και ατομικά συμπιεσμένο, αυτό παίρνει πολύ χώρο και είναι αναποτελεσματικό. Αυτό επιλύεται με τη χρήση συσκευασιών που αποθηκεύουν έναν μεγάλο αριθμό αντικειμένων σε ένα μοναδικό αρχείο (ή σε μια δικτυακή ροή ψηφιολέξεων) που λέγεται αρχείο συσκευασίας (packfile), συμπιεσμένο δέλτα μεταξύ τους. Οι συσκευασίες συμπιέζονται χρησιμοποιώντας την ευρετική (Heuristic) όπου αρχεία με το ίδιο όνομα είναι προφανώς παρόμοια, αλλά δεν εξαρτώνται από αυτό για την ορθότητα τους. Ένα αντίστοιχο αρχείο δεικτών δημιουργείται για κάθε αρχείο συσκευασίας, που λέει τη μετατόπιση κάθε αντικειμένου στο αρχείο συσκευασίας. Νεοδημιουργημένα αντικείμενα (νεοπροστιθέμενο

ιστορικό) εξακολουθούν να αποθηκεύονται μοναδικά και η περιοδική επανασυσκευασία απαιτείται για να διατηρηθεί η αποτελεσματικότητα του χώρου. Η διαδικασία της συσκευασίας του αποθετηρίου μπορεί να είναι υπολογιστικά πολύ δαπανηρή. Επιτρέποντας στα αντικείμενα να υπάρχουν στο αποθετήριο σε χαλαρή, αλλά γρήγορα δημιουργούμενη μορφή, το git επιτρέπει τη δαπανηρή λειτουργία συσκευασίας να αναβληθεί μέχρι αργότερα όταν ο χρόνος δεν πειράζει (π.χ. στο τέλος της ημέρας εργασίας). Το Git επανασυσκευάζει περιοδικά και αυτόματα, αλλά η χειροκίνητη επανασυσκευασία είναι επίσης δυνατή με την εντολή `git gc`. Για την ακεραιότητα των στοιχείων, και το αρχείο της συσκευασίας και ο δείκτης του έχουν εσωτερικά ένα άθροισμα ελέγχου SHA-1 και το όνομα του αρχείου του αρχείου συσκευασίας περιέχει επίσης ένα άθροισμα ελέγχου SHA-1. Για τον έλεγχο της ακεραιότητας, εκτελέστε την εντολή `git fsck`.

Μια άλλη ιδιότητα του Git είναι ότι παίρνει στιγμιότυπα των δένδρων καταλόγου των αρχείων. Τα πρώιμα συστήματα εντοπισμού εκδόσεων του πηγαίου κώδικα, σύστημα ελέγχου πηγαίου κώδικα-SCCS (Source Code Control System) και σύστημα ελέγχου αναθεώρησης-RCS (Revision Control System), δούλευαν σε ατομικά αρχεία και τόνιζαν την απόκτηση εξοικονόμησης χώρου από τα παρεμβαλλόμενα δέλτα (interleaved deltas) (SCCS) ή την κωδικοποίηση δέλτα (delta encoding) (RCS) των (κυρίως παρόμοιων) εκδόσεων. Τα μεταγενέστερα συστήματα ελέγχου έκδοσης διατήρησαν αυτήν την ιδέα ενός αρχείου που έχει μια ταυτότητα μέσα από πολλές αναθεωρήσεις ενός έργου. Όμως, ο Τόρβαλντς απέρριψε αυτήν την ιδέα.[27] Συνεπώς, το Git δεν καταγράφει ρητά τις σχέσεις αναθεώρησης του αρχείου σε οποιοδήποτε επίπεδο κάτω από δένδρο του πηγαίου κώδικα [17].

Οι έμμεσες σχέσεις αναθεώρησης έχουν κάποιες σημαντικές συνέπειες:

Είναι ελαφρά πιο δαπανηρό στην εξέταση της αλλαγής του ιστορικού ενός μοναδικού αρχείου από όλο το έργο. Για να ληφθεί ένα ιστορικό αλλαγών που επηρεάζει ένα δοσμένο αρχείο, το Git πρέπει να διατρέξει το συνολικό ιστορικό και έπειτα να καθορίσει αν κάθε αλλαγή τροποποιεί αυτό το αρχείο. Αυτή η μέθοδος εξέτασης ιστορικού, όμως, επιτρέπει στο Git να παράξει εξίσου καλά ένα μοναδικό ιστορικό που εμφανίζει τις αλλαγές με ένα ελεύθερο σύνολο αρχείων. Για παράδειγμα, ένας υποκατάλογος του πηγαίου δένδρου συν ένα συνδεδεμένο γενικό αρχείο κεφαλίδας είναι μια πολύ συνηθισμένη περίπτωση.

Οι μετονομασίες διαχειρίζονται έμμεσα αντί για άμεσα. Ένα συνηθισμένο παράπονο με το CVS(σύγχρονο σύστημα εκδόσεων) (Concurrent Versions System) είναι ότι χρησιμοποιεί το όνομα ενός αρχείου για να αναγνωρίσει το ιστορικό της αναθεώρησής του, έτσι η μετακίνηση ή η μετονομασία ενός αρχείου δεν είναι δυνατή χωρίς είτε διακοπή του ιστορικού του, είτε μετονομασία του ιστορικού και συνεπώς κάνει το ιστορικό ανακριβές. Τα περισσότερα μετα-CVS συστήματα ελέγχου αναθεωρήσεων επιλύουν αυτό το θέμα δίνοντας ένα αρχείο με ένα μοναδικό μακροχρόνιο όνομα (ένα είδος αριθμού κόμβου πληροφοριών) που επιβιώνει της μετονομασίας. Το Git δεν καταγράφει έναν τέτοιο αναγνωριστικό και αυτό αντιμετωπίζεται ως πλεονέκτημα. Τα αρχεία πηγαίου κώδικα διαιρούνται ή συγχωνεύονται μερικές φορές ως απλές μετονομασίες, και η καταγραφή αυτού ως μιας απλής μετονομασίας θα πάγωνε μια ανακριβή περιγραφή του τι συνέβη στο (αμετάβλητο) ιστορικό. Το Git αντιμετωπίζει αυτό το θέμα εντοπίζοντας τις μετονομασίες ενώ περιηγείται το ιστορικό των στιγμιότυπων αντί να το καταγράφει, όταν παίρνει το στιγμιότυπο. (Εν συντομία, με δεδομένο ένα αρχείο στην έκδοση N, ένα αρχείο με το ίδιο όνομα στην έκδοση N-1 είναι ο προεπιλεγμένος του πρόγονος. Όμως, όταν δεν υπάρχει αρχείο με παρόμοιο όνομα στην έκδοση N-1, το Git αναζητά ένα αρχείο που υπήρχε μόνο στην αναθεώρηση N-1 και είναι πολύ παρόμοιο με το νέο αρχείο.) Όμως, δεν απαιτεί περισσότερη εντατική επεξεργασία από τη Κεντρική Μονάδα Επεξεργασίας) κάθε φορά που το ιστορικό αναθεωρείται και έναν αριθμό επιλογών για να

προσαρμόσει την ευρετική. Αυτός ο μηχανισμός δεν δουλεύει πάντα· κάποιες φορές ένα αρχείο που μετονομάστηκε με αλλαγές στην ίδια υποβολή διαβάζεται ως μια διαγραφή του παλιού αρχείου και δημιουργία ενός νέου αρχείου. Οι προγραμματιστές μπορούν να παρακάμψουν αυτόν τον περιορισμό υποβάλλοντας τη μετονομασία και τις αλλαγές ξεχωριστά.

Το Git υλοποιεί πολλές στρατηγικές συγχώνευσης· μια μη προεπιλεγμένη μπορεί να επιλεγεί κατά τον χρόνο της συγχώνευσης:

επίλυση: ο παραδοσιακός αλγόριθμος συγχώνευσης τριών δρόμων.

αναδρομικά: Αυτή είναι η προεπιλογή κατά την μετακίνηση ή συγχώνευση ενός κλάδου και είναι μια παραλλαγή του αλγόριθμου συγχώνευσης τριών δρόμων.

Όταν υπάρχουν περισσότεροι από έναν κοινό πρόγονο που μπορεί να χρησιμοποιηθεί για συγχώνευση τριών δρόμων, δημιουργεί ένα συγχωνευμένο δένδρο των κοινών προγόνων και το χρησιμοποιεί ως δένδρο αναφοράς για τη συγχώνευση τριών δρόμων. Αυτό έχει αναφερθεί να καταλήγει σε λιγότερες συγκρούσεις συγχωνεύσεων χωρίς να προκαλεί εσφαλμένες συγχωνεύσεις με δοκιμές που γίνονται σε ενεργές υποβολές συγχώνευσης που ελήφθησαν από το ιστορικό ανάπτυξης του πυρήνα Linux 2.6. Επιπρόσθετα αυτό μπορεί να εντοπίσει και να χειριστεί συγχωνεύσεις που περιέχουν μετονομασίες.

octopus: Αυτή είναι η προεπιλογή κατά τη συγχώνευση περισσότερων από δύο κεφαλίδες [17].

4.11.3. Δομές δεδομένων του Git

Τα βασικά του Git δεν είναι εγγενώς ένα σύστημα διαχείρισης πηγαίου κώδικα (SCM). Ο Τόρβαλντς εξηγεί:

Κατά πολλούς τρόπους μπορείτε να δείτε το git ως ένα σύστημα αρχείων — είναι προσπελάσιμο με το περιεχόμενο (content-addressable) και έχει μια ιδέα αναθεώρησης, αλλά στην πραγματικότητα ο σχεδιασμός του έρχεται στο πρόβλημα από την άποψη ενός προσώπου συστήματος αρχείων (αυτό που κάνω είναι πυρήνες) και στην πραγματικότητα έχω απολύτως μηδενικό ενδιαφέρον να δημιουργήσω ένα παραδοσιακό σύστημα SCM.

Από αυτήν την αρχική σχεδιαστική προσέγγιση, το Git έχει αναπτυχθεί σε ένα πλήρες σύνολο γνωρισμάτων που θα περίμενε κανείς από ένα παραδοσιακό SCM,[19] με γνωρίσματα τα οποία κυρίως δημιουργήθηκαν όταν χρειαζόντουσαν, έπειτα εξευγενίστηκαν και επεκτάθηκαν με την πάροδο του χρόνου. Μερικά δεδομένα ροών και επιπέδων αποθήκευσης στο σύστημα ελέγχου έκδοσης.

Το Git έχει δύο δομές δεδομένων (Structure data): έναν μεταβλητό δείκτη (index) (που λέγεται επίσης καταχώριση ή κρύπτη (stage ή cache)) που αποθηκεύει πληροφορίες για τον κατάλογο εργασίας και την επόμενη έκδοση που θα υποβληθεί· και μία αμετάβλητη, μόνο για προσάρτηση βάση δεδομένων αντικειμένων. Η βάση δεδομένων αντικειμένων περιέχει τέσσερις τύπους αντικειμένων:

Ένα (δυναμικό μεγάλο αντικείμενο) blob είναι το περιεχόμενο ενός αρχείου. Τα Blobs δεν έχουν όνομα αρχείου, χρονοσημάνσεις ή άλλα μεταδεδομένα.

Ένα αντικείμενο δένδρου είναι ισοδύναμο με έναν κατάλογο. Περιέχει έναν κατάλογο των ονομάτων αρχείων, που το καθένα έχει κάποιους τύπος δυαδικών ψηφίων και το όνομα ενός blob ή αντικειμένου δένδρου που είναι αυτό το αρχείο, συμβολικός σύνδεσμος, ή περιεχόμενα καταλόγου. Αυτό το αντικείμενο περιγράφει ένα στιγμιότυπο του πηγαίου δένδρου.

Ένα αντικείμενο υποβολής συνδέει αντικείμενα δένδρου μαζί σε ένα ιστορικό. Περιέχει το όνομα ενός αντικειμένου δένδρου (του ανωτάτου επιπέδου πηγαίου καταλόγου), μια χρονοσήμανση, ένα μήνυμα ημερολογίου και τα ονόματα μηδέν ή περισσότερων γονικών αντικειμένων υποβολής.

Ένα αντικείμενο ετικέτας είναι ένας περιέκτης που περιέχει παραπομπή σε ένα άλλο αντικείμενο και μπορεί να κρατήσει πρόσθετα μεταδεδομένα σχετικά με ένα άλλο αντικείμενο. Κατά κανόνα, χρησιμοποιείται για να αποθηκεύσει μια Ψηφιακή υπογραφή ενός αντικειμένου υποβολής που αντιστοιχεί σε μια συγκεκριμένη έκδοση των δεδομένων που εντοπίζονται από το Git.

Ο δείκτης εξυπηρετεί ως σημείο σύνδεσης μεταξύ της βάσης δεδομένων του αντικειμένου και του δένδρου εργασίας.

Κάθε αντικείμενο ταυτοποιείται από έναν κατακερματισμό (Cryptographic hash function) SHA-1 των περιεχομένων του. Το Git υπολογίζει τον κατακερματισμό και χρησιμοποιεί αυτήν την τιμή για το όνομα του αντικειμένου. Το αντικείμενο μπαίνει σε έναν κατάλογο που ταιριάζει με τους δύο πρώτους χαρακτήρες του κατακερματισμού του. Το υπόλοιπο του κατακερματισμού χρησιμοποιείται ως το όνομα του αρχείου για αυτό το αντικείμενο.

Το Git αποθηκεύει κάθε αναθεώρηση ενός αρχείου ως ένα μοναδικό blob. Η σχέση μεταξύ των blobs μπορεί να βρεθεί μέσα από την εξέταση του δένδρου και των αντικειμένων υποβολής. Νεοπροστιθέμενα αντικείμενα αποθηκεύονται στην ολότητα τους χρησιμοποιώντας μια συμπίεση zlib. Αυτό μπορεί να καταναλώσει ένα μεγάλο ποσό του χώρου του δίσκου γρήγορα, έτσι τα αντικείμενα μπορούν να ενωθούν σε πακέτα, που χρησιμοποιούν δέλτα συμπίεση για να αποθηκεύσουν χώρο, αποθηκεύοντας τα blobs ως αλλαγές τους σχετικά με άλλα blobs.

Οι διακομιστές του Git τυπικά ακροάζονται στη θύρα TCP 9418 [17].



4.12. Spring Framework

Το Spring Framework είναι λογισμικό ανοιχτού κώδικα (ελεύθερο λογισμικό) που σκοπό έχει να διευκολύνει την ανάπτυξη J2EE λογισμικού σε μεγάλη έκταση και βασίζεται στη γλώσσα προγραμματισμού Java.

Το Spring χρησιμοποιεί το Inversion of Control και το dependency injection για την προώθηση καλών πρακτικών για την συγγραφή κώδικα και την επιτάχυνση του χρόνου ανάπτυξης

του λογισμικού. Τα βασικά του αρθρώματα μπορούν να χρησιμοποιηθούν από οποιαδήποτε εφαρμογή Java, αλλά υπάρχουν αρθρώματα με τα οποία κατασκευάζονται εφαρμογές και είναι πάνω από την πλατφόρμα Java EE (Enterprise Edition). Παρόλο που το framework δεν επιβάλλει συγκεκριμένο μοντέλο προγραμματισμού, έχει γίνει δημοφιλές στην κοινότητα Java ως άρθρωμα ή ακόμα και ως αντικατάσταση του μοντέλου Enterprise JavaBeans (EJB).



SPRING
Framework

4.12.1. Spring MVC

Το Spring Framework διαθέτει δικό του model-view-controller (MVC), για εφαρμογές ιστού το οποίο δεν σχεδιάστηκε αρχικά. Οι προγραμματιστές του Spring αποφάσισαν να φτιάξουν το δικό τους framework στο διαδίκτυο ως αντίδραση σε αυτό, που θεωρούσαν ως κακή σχεδίαση του τότε δημοφιλούς Jakarta Struts Web Framework, καθώς και το ότι δεν υπήρχε άλλο διασθεσίμο framework. Συγκεκριμένα, θεώρησαν ότι υπήρχε ανεπαρκής διαχωρισμός μεταξύ των layers παρουσίασης και του χειρισμού των http request και της διαχείρισης των http request και του model. Όπως το Struts, το Spring MVC είναι βασισμένο σε σε http αιτήματα βασιμμένα στο framework.

Τα πιο σημαντικά interfaces που ορίζονται από το Spring MVC είναι τα εξής :

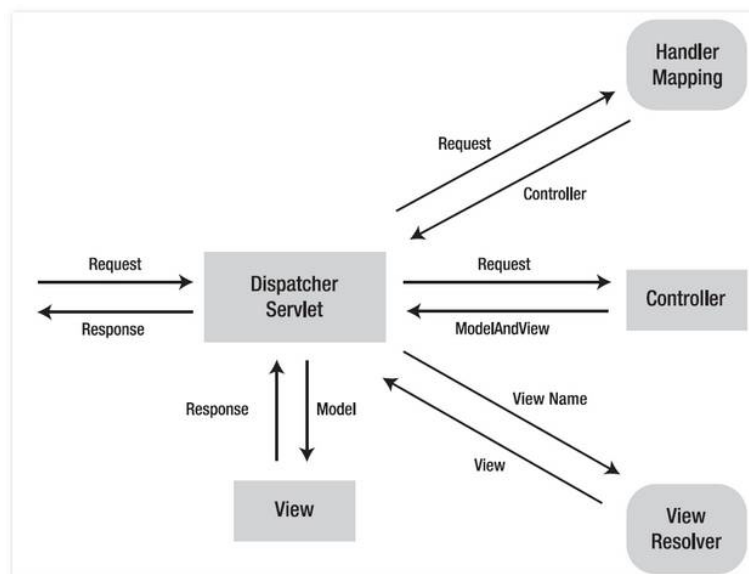
- **HandlerAdapter** : εκτέλεση των αντικειμένων που διαχειρίζονται τα εισερχόμενα http αιτήματα
- **HandlerInterceptor**: η παρακολούθηση των εισερχόμενων αιτημάτων είναι συγκρίσιμη, αλλά όχι ίση με τα φίλτρα Servlet (η χρήση είναι προαιρετική και δεν ελέγχεται από το DispatcherServlet).
- **HandlerMapping**: επιλογή αντικειμένων που χειρίζονται εισερχόμενες αιτήσεις (χειριστές) με βάση οποιοδήποτε χαρακτηριστικό ή προϋπόθεση εσωτερική ή εξωτερική σε αυτά τα αιτήματα
- **LocaleResolver**: επίλυση και προαιρετική αποθήκευση της τοπικής ρύθμισης ενός μεμονωμένου χρήστη

4.12.2. Ο DispatcherServlet

Η τάξη DispatcherServlet είναι ο μπροστινός controller του framework και είναι υπεύθυνος για τον έλεγχο κατά τη διάρκεια των φάσεων εκτέλεσης ενός αιτήματος HTTP.

Το πλαίσιο MVC του Spring της ιστοσελίδας είναι, όπως και πολλά άλλα πλαίσια MVC στο διαδίκτυο, βασισμένο σε αιτήματα, σχεδιασμένο γύρω από ένα κεντρικό Servlet που αποστέλλει αιτήσεις στον controller και προσφέρει άλλες λειτουργίες που διευκολύνουν την ανάπτυξη εφαρμογών ιστού. Ο DispatcherServlet της Spring, ωστόσο, κάνει κάτι περισσότερο από αυτό. Είναι εντελώς ενσωματωμένο με το Spring IoC και ως εκ τούτου σας επιτρέπει να χρησιμοποιείτε κάθε άλλο χαρακτηριστικό που έχει το Spring.

Η ροή εργασιών επεξεργασίας αιτήματος του MVC DispatcherServlet του Spring Web απεικονίζεται στο ακόλουθο διάγραμμα.



Το DispatcherServlet είναι ένα πραγματικό Servlet (κληρονόμησε από την κλάση HttpServlet) και ως τέτοιο δηλώνεται στο web.xml της εφαρμογής σας στο διαδίκτυο. Θα πρέπει να αντιστοιχίσετε τα αιτήματα HTTP που θέλετε να χειριστεί το DispatcherServlet, χρησιμοποιώντας ένα URL mapping στο ίδιο αρχείο web.xml. Αυτή είναι η τυπική διαμόρφωση του Java EE Servlet [18]. Το ακόλουθο παράδειγμα δείχνει μια τέτοια δήλωση και mapping του DispatcherServlet:

```
<web-app>

  <servlet>
    <servlet-name>example</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>example</servlet-name>
    <url-pattern>/example/*</url-pattern>
  </servlet-mapping>

</web-app>
```

4.13. HTML

Η HTML είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων.

Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα <html>), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα <h1> και </h1>), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός web browser είναι να διαβάζει τα έγγραφα HTML και να τα συνθέσει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να παρουσιάσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML και από στατικές τις κάνουν διαδραστικές.

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οργανισμός W3C, ο

οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.

4.14. CSS

Η CSS (Cascading Style Sheets – Διαδοχικά Φύλλα Ύφους) ή (αλληλουχία φύλλων ύφους) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία φύλλων ύφους και χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης.

Χρησιμοποιείται για τον έλεγχο εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο και την εμφάνιση μια ιστοσελίδας. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα στοίχιση και να δίνει περισσότερες δυνατότητες σε σχέση με την html στυλιστικά. Η CSS κρίνεται απαραίτητη για μια ιστοσελίδα όμορφη και καλοσχεδιασμένη.

Η χρήση CSS κάνει πολύ εύκολη την διαχείριση της εμφάνισης των σελίδων μας. Για παράδειγμα, αν σε μια ιστοσελίδα, χωρίς χρήση CSS, με την χρήση html ή xhtml θέλω να αλλάξω το χρώμα του φόντου σε όλες τις επικεφαλίδες όλων των πινάκων θα πρέπει να πηγαίνω σε κάθε μια επικεφαλίδα και να ορίζω το χρώμα που επιθυμώ αλλάζοντας κάθε φορά την τιμή της ιδιότητας bgcolor των ετικετών <th>. Ενώ σε μια σελίδα με χρήση CSS θα χρειαστεί να αλλάξω το χρώμα αυτό μια μόνο φορά και αυτό θα εφαρμοστεί για όλες τις επικεφαλίδες των πινάκων της σελίδας. Παρόμοια μπορώ να ορίσω στυλ, όχι μόνο για στοιχεία μιας σελίδας, αλλά για στοιχεία όλων των σελίδων του Site μου.

Μπορείτε να αλλάζετε το στυλ του οριζόντιου μενού του Site σας, το οποίο επαναλαμβάνεται σε όλες τις σελίδες, με μόνο μια κίνηση.

Σήμερα υπάρχουν πολύ λίγες ιστοσελίδες που δεν χρησιμοποιούν CSS. Η χρήση των στυλ κάνει την ζωή των Web designers πολύ πιο εύκολη δημιουργώντας έτσι ιστοσελίδες τα οποία μπορούν να διαχειρίζονται εύκολα και γρήγορα.

4.15. Javascript

Η Javascript είναι διεργημευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών ιστού, ώστε από την πλευρά του πελάτη (client-side) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενου του εγγράφου που εμφανίζεται.

Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξη της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side).

Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMAScript.

Η Javascript δεν θα πρέπει να συγχέεται με τη Java, που είναι διαφορετική γλώσσα προγραμματισμού και με διαφορετικές εφαρμογές. Η χρήση της λέξης "Java" στο όνομα της γλώσσας έχει περισσότερη σχέση με το προφίλ του προϊόντος που έπρεπε να έχει και λιγότερο με κάποια πιθανή συμβατότητα ή άλλη στενή σχέση με τη Java. Ρόλο σε αυτήν τη σύγχυση έπαιξε και ότι η Java και η Javascript έχουν δεχτεί σημαντικές επιρροές από τη γλώσσα C, ειδικά στο συντακτικό, ενώ είναι και οι δύο αντικειμενοστρεφείς γλώσσες. Τονίζεται ότι ο σωστός τρόπος γραφής της είναι "Javascript" και όχι "Java script" σαν δύο λέξεις, όπως λανθασμένα γράφεται ορισμένες φορές.

Για παράδειγμα, ο ακόλουθος κώδικας Javascript εμφανίζει ένα πλαίσιο διαλόγου με το κείμενο "Γεια σου, κόσμε!":

```
<script type="text/javascript">  
alert("Γεια σου, κόσμε!");  
</script>
```

5. Ανάλυση του κώδικα και του αλγορίθμου στην διαδικτυακή μας εφαρμογή.

Στο κεφάλαιο αυτό θα γίνει ανάλυση των βασικών σημείων του κώδικα, καθώς και του αλγορίθμου που χρησιμοποιήσαμε, ουσιαστικά η διαδικτυακή μας εφαρμογή να καταστεί λειτουργική.

5.1 Εισαγωγή.

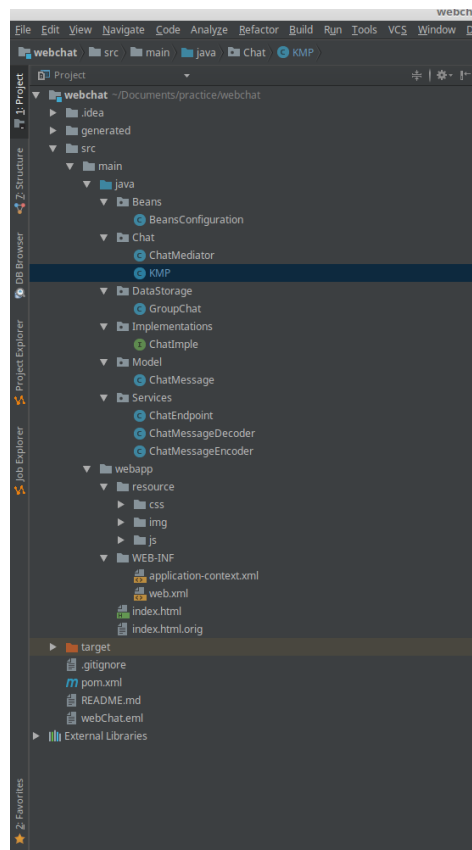
Η διαδικτυακή μας εφαρμογή είναι χωρισμένη σε δύο μέρη τα οποία είναι αλληλένδετα :

- Η αμφίδρομη επικοινωνία των χρηστών, μέσω της φυσικής γλώσσας εντός της διαδικτυακής μας εφαρμογής.
- Υλοποίηση του αλγορίθμου για την ανάλυση των λέξεων που μαρτυρούν θυματοποίηση.

Μπορείτε παρακάτω να δείτε την διάρθρωση των πακέτων και των κλάσεων, τα οποία είναι με τέτοιο τρόπο χωρισμένα προκειμένου η εφαρμογή μας να είναι εύκολα διαχειρίσιμη, και αν χρειαστούν κάποιες αλλαγές να μπορούν να γίνουν χωρίς ιδιαίτερο κόπο.

Στο πρώτο πακέτο με όνομα Beans μπορεί να δει κάποιος ότι βρίσκονται τα beans μας , στο δεύτερο πακέτο με όνομα Chat είναι ο διαμοιραστής και ο αλγόριθμός μας. Στο τρίτο μέρος DataStorage είναι η αποθήκευση της <<κακής>> λέξης του χρήστη σε αρχείο. Στο τέταρτο μέρος είναι ένα interface με όνομα Implementations. Έπειτα το πακέτο Model που εκεί βρίσκονται τα δεδομένα μας. Στην συνέχεια είναι το πακέτο Services εκεί γίνεται ο διαμοιρασμός των δεδομένων μεταξύ του server και του client.

Τέλος στο πακέτο webapp βρίσκεται το configuration της εφαρμογής μας γραμμένο σε xml, μαζί με το pom αρχείο με τα dependencies του. Μπορούμε επίσης να δούμε και το html αρχείο μας, στο οποίο έχουμε την css και την Javascript.



5.2. Ο Αλγόριθμος Levenshtein distance

Ο αλγόριθμος ονομάστηκε Levenshtein distance από τον Vladimir Levenshtein, ο οποίος ανακάλυψε αυτή την εξίσωση το 1965. Στον αλγόριθμο Levenshtein distance χρησιμοποιούμε δύο αλφαριθμητικά, όπου μετράμε την διαφορά μεταξύ των δύο αυτών ακολουθιών. Συμπληρωματικά, στον αλγόριθμο Levenshtein distance τροποποιούμε τους χαρακτήρες με εισαγωγές διαγραφές ή αντικαταστάσεις, που απαιτούνται για την ταύτιση της μιας λέξης στην άλλη.

Ο αλγόριθμος μπορεί να αναφέρεται στην απόσταση επεξεργασίας των χαρακτήρων, αλλά επίσης μπορεί να χρησιμοποιηθεί σε μεγαλύτερες μετρήσεις και όχι μόνο. Συνδέεται στενά με τις ευθυγραμμίσεις κατά ζεύγη. Για παράδειγμα στην βιοπληροφορική, η ευθυγράμμιση αλληλουχίας είναι ένας τρόπος διαμόρφωσης του DNA, του RNA ή της πρωτεΐνης για τον προσδιορισμό ομοιότητας που μπορεί να είναι συνέπεια λειτουργικών, δομικών ή εξελικτικών σχέσεων μεταξύ ακολουθιών.

5.2.1 Χρήση του αλγορίθμου Levenshtein distance στην εφαρμογή μας

Στην διαδικτυακή μας εφαρμογή όπως αναφέραμε γίνεται επικοινωνία των χρηστών μέσω του WebChat, όπου ένας χρήστης μπορεί να γράψει στο γκρουπ οτιδήποτε θέλει. Για να δούμε αν ένα παιδί υφίσταται εκφοβισμό (bullying), δημιουργήσαμε ένα λεξικό με τις λεγόμενες <<κακές>> ή <<απαγορευμένες>> λέξεις που μπορεί να γράψει μέσα από την εφαρμογή μας ένας χρήστης σε έναν άλλον, στις οποίες μέσω του αλγόριθμου κάνουμε σύγκριση των λέξεων που βρίσκονται στο λεξικό μας, με τις λέξεις που γράφει ο χρήστης στο γκρουπ.

Θα γίνει κατανοητό αν προχωρήσουμε στην ανάλυση της κλάσης του αλγορίθμου μας. Για παράδειγμα, όπως βλέπουμε στην εικόνα εντός του πακέτου chat και στην κλάση KMP, η οποία είναι ένα component του Spring framework μας. Υπάρχει μία μέθοδο similarity στην οποία της περνάμε ως παραμέτρους δύο string, την λέξη απ το λεξικό μας και την λέξη του χρήστη.

Προχωράμε βρίσκοντας ποιά λέξη είναι πιο μεγάλη σε μήκος από την άλλη, έπειτα τις περνάμε ως παραμέτρους στην μέθοδο editDistance, στην οποία θα γίνει η αντιμεταθέση των γραμμάτων αντικατάσταση η και διαγραφή τους, στο τέλος θα δούμε κατά πόσο μοιάζουν μεταξύ τους.

```

package Chat;

import ...

@Component
public class KMP {

    /**
     * Calculates the similarity (a number within 0 and 1) between two strings.
     */
    public static double similarity(String s1, String s2) {
        String longer = s1, shorter = s2;
        if (s1.length() < s2.length()) { // longer should always have greater length
            longer = s2; shorter = s1;
        }
        int longerLength = longer.length();
        if (longerLength == 0) { return 1.0; /* both strings are zero length */ }
        return (longerLength - editDistance(longer, shorter)) / (double) longerLength;
    }

    public static int editDistance(String s1, String s2) {
        s1 = s1.toLowerCase();
        s2 = s2.toLowerCase();

        int[] costs = new int[s2.length() + 1];
        for (int i = 0; i <= s1.length(); i++) {
            int lastValue = i;
            for (int j = 0; j <= s2.length(); j++) {
                if (i == 0)
                    costs[j] = j;
                else {
                    if (j > 0) {
                        int newValue = costs[j - 1];
                        if (s1.charAt(i - 1) != s2.charAt(j - 1))
                            newValue = Math.min(Math.min(newValue, lastValue),
                                costs[j]) + 1;
                        costs[j - 1] = lastValue;
                        lastValue = newValue;
                    }
                }
            }
            if (i > 0)
                costs[s2.length()] = lastValue;
        }
        return costs[s2.length()];
    }
}

```

Σχεδιασμός, Ανάπτυξη και Εφαρμογή Προηγμένου Λογισμικού Διεπαφής χρηστών διαδικτυακών συζητήσεων (webChat), με τη χρήση της γλώσσας Java, για το σκοπό ανίχνευσης λέξεων της φυσικής γλώσσας που μαρτυρούν θυματοποίηση (Bullying).

Όπως βλέπουμε στην παρακάτω εικόνα, τυπώνουμε το αποτέλεσμα σε ποσοστιαίες μονάδες το οποίο και το επιστρέφουμε, για να καταχωρηθεί στο αρχείο μας με τα στοιχεία του χρήστη την ημερομηνία ώρα και την ανάρμοστη λέξη που έγραψε ο χρήστης.

Η μέθοδος `getWords` παίρνει το `String` του χρήστη και αφαιρεί τους ειδικούς χαρακτήρες, επιστρέφοντας την λέξη απαλλαγμένη από χαρακτήρες και αριθμούς που δεν θέλουμε.

```
public static double printSimilarity(String s, String t) {
    System.out.println(String.format(
        s: "%.2f is the similarity between \"%s\" and \"%s\"", Objects: 100 * similarity(s, t), s, t));

    return 100 * similarity(s, t);
}

public static List<String> getWords(String text) {
    List<String> words = new ArrayList<String>();
    BreakIterator breakIterator = BreakIterator.getWordInstance();
    breakIterator.setText(text);
    int lastIndex = breakIterator.first();
    while (BreakIterator.DONE != lastIndex) {
        int firstIndex = lastIndex;
        lastIndex = breakIterator.next();
        if (lastIndex != BreakIterator.DONE && Character.isLetterOrDigit(text.charAt(firstIndex))) {
            words.add(text.substring(firstIndex, lastIndex));
        }
    }

    return words;
}
}
```

5.3. Χρήση του design pattern Mediator (Διαμοιραστή) στην εφαρμογή μας.

Στην επιστήμη της πληροφορικής υπάρχουν διάφορα design patterns ένα απ το οποίο είναι ο mediator (διαμοιραστής), ο mediator ορίζει ένα αντικείμενο στο οποίο ενθυλακώνει τα υπόλοιπα αντικείμενα με τα οποία αλληλεπιδρά. Αυτή η λογική είναι συμπεριφοριστική (behavioral design patterns), η οποία αναγνωρίζει μία κοινή επικοινωνία ανάμεσα στα αντικείμενα, η οποία μπορεί να αλλάξει την συμπεριφορά του προγράμματος το οποίο τρέχει.

Σχεδιασμός, Ανάπτυξη και Εφαρμογή Προηγμένου Λογισμικού
Διεπαφής χρηστών διαδικτυακών συζητήσεων (webChat), με τη
χρήση της γλώσσας Java, για το σκοπό ανίχνευσης λέξεων της
φυσικής γλώσσας που μαρτυρούν θυματοποίηση (Bullying).

Με το πρότυπο μεσολαβητή, η επικοινωνία μεταξύ των αντικειμένων περιορίζεται μέσα σε ένα αντικείμενο μεσολαβητή. Τα αντικείμενα δεν επικοινωνούν απευθείας μεταξύ τους, αλλά επικοινωνούν μέσω του διαμεσολαβητή. Με αυτό τον τρόπο μειώνεται η εξάρτηση, ανάμεσα στα αντικείμενα επικοινωνίας, μειώνοντας έτσι την σύζευξη (coupling).

Παρακάτω στην εικόνα βλέπουμε, την κλάση ChatMediator όπου διαπιστώνουμε την λογική με το πρότυπο του μεσολαβητή. Στην μέθοδο sendMessage περνάμε ως μεταβλητή τα δεδομένα μας. Στην συνέχεια, αποστέλλουμε τα δεδομένα μας στην μέθοδο που φτιάξαμε για να απαλλαχθεί από ειδικούς χαρακτήρες και αριθμούς.

Έπειτα δημιουργούμε έναν πίνακα με το λεξικό μας (για λόγους ανάλυσης του κώδικα βάζουμε μόνο τρεις λέξεις, στην πραγματικότητα υπάρχει το λεξικό σε αρχείο) και σπάμε τις προτάσεις σε λέξεις, ουσιαστικά να μπορούμε να διαβάσουμε λέξη προς λέξη. Παράλληλα, ελέγχουμε την λέξη του χρήστη με το λεξικό μας.

Τέλος, εάν ξεπερνά το 50% η ομοιότητα ανάμεσα στις δύο λέξεις, τότε στέλνουμε τα δεδομένα αυτά προς αποθήκευση για την ενημέρωση του δασκάλου ή του administrator (διαχειριστή συστήματος), διαφορετικά τα προσπερνάμε για να μην αποθηκευτούν.

```
import DataStorage.GroupChat;
import Model.ChatMessage;
import Implementations.ChatImple;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;
import java.text.Normalizer;

@Component
public class ChatMediator implements ChatImple{

    @Autowired
    GroupChat saveData;

    @Autowired
    KMP kmp;

    @Override
    public void sendMessage(ChatMessage message) {

        kmp.getWords(message.getMessage());

        String[] array = new String[]{"ΓΑΙΔΟΥΡΙ", "ΑΣΧΗΜΟΣ", "ΜΑΠΑΣ"};
        Double resultPercentage;

        for(String result : kmp.getWords(stripAccents(message.getMessage().toUpperCase())))
        {
            for(String wordArray : array) {
                resultPercentage = kmp.printSimilarity(result, wordArray);

                if(kmp.printSimilarity(result, wordArray) >= 50)
                {
                    saveData.showMessage(message, resultPercentage);
                }
            }
        }
    }
}
```

5.4. Οργάνωση των δεδομένων (Data model).

Στο πακέτο Model υπάρχει η κλάση ChatMessage, εκεί βρίσκονται τα δεδομένα μας τα οποία καλούμε η περνάμε την πληροφορία, ώστε να είναι διαθέσιμη από τα αντικείμενά μας.

Αναλυτικά υπάρχει το μήνυμα του χρήστη, το όνομα του χρήστη, η ημερομηνία και ώρα μηνύματος, το session του καθώς και το όνομα του γκρουπ που έκανε σύνδεση και η φωτογραφία του.

```

package Model;

import java.util.Date;

public class ChatMessage {

    private String message;
    private String sender;
    private Date received;
    private String session;
    private String room;
    private String image;

    public ChatMessage(String message, String sender, Date received, String session, String room, String image) {
        this.message = message;
        this.sender = sender;
        this.received = received;
        this.session = session;
        this.room = room;
        this.image = image;
    }

    public String getImage() { return image; }

    public void setImage(String image) { this.image = image; }

    public String getRoom() { return room; }

    public void setRoom(String room) { this.room = room; }

    public String getSession() { return session; }

    public void setSession(String session) { this.session = session; }

    public final String getMessage() { return message; }

    public final void setMessage(final String message) { this.message = message; }

    public final String getSender() { return sender; }

    public final void setSender(final String sender) { this.sender = sender; }

    public final Date getReceived() { return received; }

    public final void setReceived(final Date received) { this.received = received; }

    @Override
    public String toString() {
        return "ChatMessage [message=" + message + ",image=" + image+", sender=" + sender
            + ", received=" + received + "];"
    }
}

```

5.5. Αποθήκευση των δεδομένων.

Την λειτουργία της αποθήκευσης που βρίσκεται στο πακέτο `DataStorage`, την υλοποιήσαμε στην κλάση με όνομα `GroupChat`. Εντός της κλάσης, βρίσκονται δύο μέθοδοι με όνομα `showMessage` και `writeCsvFile`. Στην μέθοδο `showMessage` περνάμε δύο παραμέτρους, η μία παράμετρος θα είναι τα δεδομένα της κλάσης `ChatMessage` και η άλλη το ποσοστό ομοιότητας των λέξεων που δοκιμάσαμε, σκοπός της μεθόδου είναι η αποτύπωση των δεδομένων στα logs του server μας, κυρίως για λόγους δοκιμής. Στην ίδια μέθοδο περνάμε ακριβώς τις ίδιες παραμέτρους με αποστολή των δεδομένων στην `writeCsvFile`, όπου σκοπός της είναι η αποθήκευση των δεδομένων μας.

Συγκεκριμένα, βλέπουμε ένα `try-catch` block μέσα στο οποίο θα διαχειριστούμε, τα δεδομένα μας τα οποία όπως βλέπουμε είναι το `session` του χρήστη, το γκρουπ στο οποίο ανήκει, το μήνυμα του, η ημερομηνία και ώρα και τέλος το ποσοστό ομοιότητας με το λεξικό μας.


```
package DataStorage;
import Model.ChatMessage;
import org.springframework.stereotype.Component;

import java.io.*;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;

@Component
public class GroupChat {

    public static void showMessage(ChatMessage message, Double percentage) {
        System.out.println(new Date().toString() + " [" + message.getSession() + "] : " + message.getMessage());
        writeCsvFile(message, percentage);
    }

    private static void writeCsvFile(ChatMessage message, Double percentage) {
        try {
            DateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
            Date date = new Date();
            BufferedWriter writer = new BufferedWriter(new FileWriter("./file.csv", true));
            writer.append("\n"+message.getSession() + "," + message.getRoom() + "," + message.getMessage() + "," + message.getSender() + ","
                + dateFormat.format(date) + ", Το ποσοστό ομοιότητας είναι = " + percentage + " receiver");

            System.out.println(writer);

            writer.flush();
            writer.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

5.6. Η υπηρεσία Web Socket.

Δημιουργήσαμε το πακέτο υπηρεσία όπου μέσα εκεί βρίσκεται η κλάση ChatEndPoint, όπου έχει το websocket container το οποίο μας δίνει την δυνατότητα επικοινωνία μας μέσω ενός url που του θέτουμε ως σημείο αναφοράς μεταξύ του client και του server. Υπάρχουν δύο μέθοδοι, πρώτα η open ελέγχει αν υπάρχει ανοικτή γραμμή επικοινωνίας. Δεύτερη είναι η onMessage η οποία διαχειρίζεται την πληροφορία μας, εφόσον υπάρχουν ανοιχτά session των χρηστών.

```

package Services;
import ...

@ServerEndpoint(value = "/chat/{room}", encoders = ChatMessageEncoder.class, decoders = ChatMessageDecoder.class)
public class ChatEndpoint {
    private final Logger log = Logger.getLogger(getClass().getName());

    @OnOpen
    public void open(final Session session, @PathParam("room") final String room) {
        log.info("session opening and bound to room: " + room);
        session.getUserProperties().put("room", room);
    }

    @OnMessage
    public void onMessage(final Session session, final ChatMessage chatMessage) {
        String room = (String) session.getUserProperties().get("room");
        ApplicationContext applicationContext = new AnnotationConfigApplicationContext(BeansConfiguration.class);
        ChatMediator getIt = applicationContext.getBean(ChatMediator.class);
        int i = 0;

        try {
            for (Session s : session.getOpenSessions()) {
                if (s.isOpen() && room.equals(s.getUserProperties().get("room"))) {
                    s.getBasicRemote().sendObject(chatMessage);
                    chatMessage.setSession(session.getId());
                    chatMessage.setRoom(room);

                    if(i==0)
                        getIt.sendMessage(chatMessage);
                }
                i++;
            }
        } catch (IOException | EncodeException e) {
            log.log(Level.WARNING, "onMessage failed", e);
        }
    }
}

```

5.7. Η υπηρεσία chatMessageEncoder και chatMessageDecoder.

Στο σημείο αυτό δημιουργήσαμε δύο κλάσεις οι οποίες είναι άρρηκτα συνδεδεμένες με την κλάση ChatEndPoint, πρώτη είναι η chatMessageEncoder η οποία χρησιμοποιείται για να στέλνει τα μηνύματα, μαζί με όλες τις πληροφορίες του χρήστη. Δεύτερη είναι η chatMessageDecoder η οποία επι της ουσίας αποκωδικοποιεί το μήνυμα του χρήστη και όλες τις απαραίτητες πληροφορίες του.

Μέσα στην κλάση chatMessageEncoder, βρίσκεται η μέθοδος encode η οποία επιστρέφει ένα String, εκεί αναθέτουμε όλες τις πληροφορίες μας σε ένα json object. Με τον ίδιο ακριβώς τρόπο δουλεύουμε και στην κλάση chatMessageDecoder, όπου μέσα εκεί βρίσκεται η μέθοδος decode η οποία επιστρέφει ένα αντικείμενο chatMessage με όλες τις πληροφορίες, τις οποίες τις αναθέτουμε στο μοντέλο μας. Παρακάτω θα δείτε στις εικόνες αναλυτικά, τις δύο κλάσεις.

```
package Services;

import ...

public class ChatMessageEncoder implements Encoder.Text<ChatMessage> {
    @Override
    public void init(final EndpointConfig config) {
    }

    @Override
    public void destroy() {
    }

    @Override
    public String encode(final ChatMessage chatMessage) throws EncodeException {
        return Json.createObjectBuilder()
            .add( name: "message", chatMessage.getMessage())
            .add( name: "image", chatMessage.getImage())
            .add( name: "sender", chatMessage.getSender())
            .add( name: "received", chatMessage.getReceived().toString()).build()
            .toString();
    }
}
```

```
public class ChatMessageDecoder implements Decoder.Text<ChatMessage> {

    @Override
    public void init(final EndpointConfig config) {
    }

    @Override
    public void destroy() {
    }

    @Override
    public ChatMessage decode(final String textMessage) throws DecodeException {
        ChatMessage chatMessage = new ChatMessage();
        JsonObject obj = Json.createReader(new StringReader(textMessage))
            .readObject();
        chatMessage.setMessage(obj.getString( name: "message"));
        chatMessage.setImage(obj.getString( name: "image"));
        chatMessage.setSender(obj.getString( name: "sender"));
        chatMessage.setReceived(new Date());
        return chatMessage;
    }

    @Override
    public boolean willDecode(final String s) { return true; }
}
```

Σχεδιασμός, Ανάπτυξη και Εφαρμογή Προηγμένου Λογισμικού
Διεπαφής χρηστών διαδικτυακών συζητήσεων (webChat), με τη
χρήση της γλώσσας Java, για το σκοπό ανίχνευσης λέξεων της
φυσικής γλώσσας που μαρτυρούν θυματοποίηση (Bullying).

5.8. Η Html σελίδα της εφαρμογής και η χρήση της Javascript.

Με την html ιστοσελίδα μας υπήρξε προσπάθεια να κάνουμε την εφαρμογή όσο το δυνατόν πιο ελκυστική για τον χρήστη και πιο εύκολα διαχειρίσιμη απ' τον προγραμματιστή της εφαρμογής.

Χρησιμοποιήσαμε την έκδοση Html5 με τη χρήση του Bootstrap, ως css framework το οποίο και μας βοήθησε στην δημιουργία μίας εφαρμογής 'One Page' . Μέσω αυτού η εφαρμογή μας, είναι εύκολα προσαρμόσιμη απο έξυπνα κινητά και Tablet.

```

});
});
</script>
</head>
<body>
<div class="container chat-signin">
<form class="form-signin">
<h2 class="form-signin-heading">WebChat</h2>
<label for="nickname">Nickname</label> <input type="text" class="input-block-level" placeholder="θα είναι το όνομα που θα συνδεθείς" id="nickname">
<div class="btn-group">
<label for="chatroom">Chatroom</label>
<select size="1" id="chatroom">
<option>Μαθηματικά</option>
<option>Ιστορία</option>
<option>Γεωγραφία</option>
<option>Θύκια</option>
</select>
</div>
<!--button class="btn btn-large btn-primary" type="submit"
id="enterRoom">Sign in</button-->
<div id="firebaseui-auth-container"></div>
</form>
</div>
<!-- /container -->
<div class="container chat-wrapper">
<form id="do-chat">
<div id="image" class="face back">
<h2 class="alert alert-success"></h2>
</div>
<table id="response" class="table table-bordered"></table>
<fieldset>
<legend>Enter your message..</legend>
<div class="controls">
<input type="text" class="input-block-level" placeholder="Your message..." id="message" style="width: 100%;"/>
<input type="hidden" name="image" id="imageChat" value="" />
<input type="submit" class="btn btn-large btn-block btn-primary"
value="Send message" />
<button class="btn btn-large btn-block" type="button" id="leave-room">Leave room</button>
</div>
</fieldset>
</form>
</div>
</body>
</html>

```

Στην παρακάτω εικόνα θα δείτε ότι δημιουργήσαμε συναρτήσεις στο header της ιστοσελίδας μας μέσω της Javascript, όπου πετύχαμε την αλληλομεταφορά δεδομένων σε μορφή Json απ' τον Client στο Server. Το login και η εγγραφή του χρήστη στο σύστημα γίνεται με την Javascript μέσω της Firebase.

Στην αρχή δημιουργούμε ένα αντικείμενο για να ανοίξουμε μία σύνδεση για την επικοινωνία απ τον Client στο Server μέσω Web Socket με την συνάρτηση connectToChatServer, έπειτα στέλνουμε και λαμβάνουμε μηνύματα σε μορφή Json όπου τα κάνουμε encode και decode και μέσω του div id τα αποτυπώνουμε και τα εμφανίζουμε στην ιστοσελίδα.

Με την συνάρτηση leaveRoom διακόπτουμε την σύνδεση του Web Socket και ανακατευθύνουμε τον χρήστη στην αρχική σελίδα.

Σχεδιασμός, Ανάπτυξη και Εφαρμογή Προηγμένου Λογισμικού Διεπαφής χρηστών διαδικτυακών συζητήσεων (webChat), με τη χρήση της γλώσσας Java, για το σκοπό ανίχνευσης λέξεων της φυσικής γλώσσας που μαρτυρούν θυματοποίηση (Bullying).

```

var websocket;
//var serviceLocation = "ws://83.212.99.30/webChatSocial/chat/";
var serviceLocation = "ws://192.168.1.118:4549/webChatSocial/chat/";
//var serviceLocation = "ws://localhost:8080/webChatSocial/chat/";

var SnickName;
var $message;
var $image;
var $imageIcon;
var $chatWindow;
var room = '';
var b = 0;

function onMessageReceived(evt) {
    var msg = JSON.parse(evt.data); // native API
    var $messageLine = $('<tr class="change"><td class="received">' + msg.received
        + '</td><td class="imageN"></img>'
        + '</td><td class="user label label-info">' + msg.sender
        + '</td><td class="message badge">' + msg.message
        + '</td></tr>');
    $chatWindow.append($messageLine);

    var img = $("#checkIt");

    if (img.attr('src') != $image.val()) {
        $('#image').append('');
        console.log(" " +msg.image + " " + $image.val());
    }
}

function sendMessage() {
    var msg = '{"message":"' + $message.val() + '", "image":"'
        + $image.val() + '", "sender":"'
        + $nickName.val() + '", "received":""}';
    websocket.send(msg);
    $message.val('').focus();
}

function connectToChatserver() {
    room = $('#chatroom option:selected').val();
    websocket = new WebSocket(serviceLocation + room);
    websocket.onmessage = onMessageReceived;
}

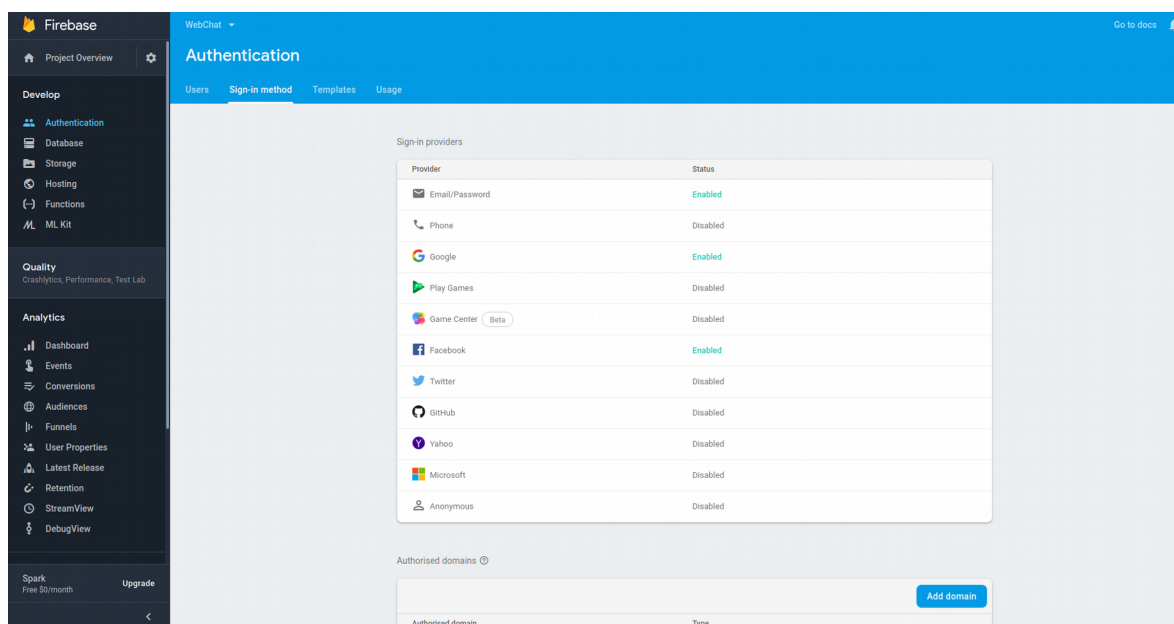
function leaveRoom() {
    websocket.close();
    $chatWindow.empty();
    $('#chat-wrapper').hide();
    $('#chat-signin').show();
    $nickName.focus();
    window.location = "http://192.168.1.118:4549/webChatSocial/";
    //window.location = "http://83.212.99.30/webChatSocial/";
}
}

```

5.9. Η Firebase στην εφαρμογή.

Με την χρήση της Firebase και των βιβλιοθηκών μπορούμε να διαχειριζόμαστε τους χρήστες, στην Firebase υπάρχει οδηγός εγκατάστασης μέσω μια βιβλιοθήκης Javascript, που μας επιτρέπει πολύ εύκολα την σύνδεση για την προστασία της εφαρμογής μας και την σύνδεση μέσω των εφαρμογών που θέλουμε. Χρησιμοποιήσαμε τις πιο διαδεδομένες εφαρμογές όπως το Facebook και το Gmail για την σύνδεση login του χρήστη στην εφαρμογή μας.

Σχεδιασμός, Ανάπτυξη και Εφαρμογή Προηγμένου Λογισμικού Διεπαφής χρηστών διαδικτυακών συζητήσεων (webChat), με τη χρήση της γλώσσας Java, για το σκοπό ανίχνευσης λέξεων της φυσικής γλώσσας που μαρτυρούν θυματοποίηση (Bullying).



6. Η λειτουργία της εφαρμογής.

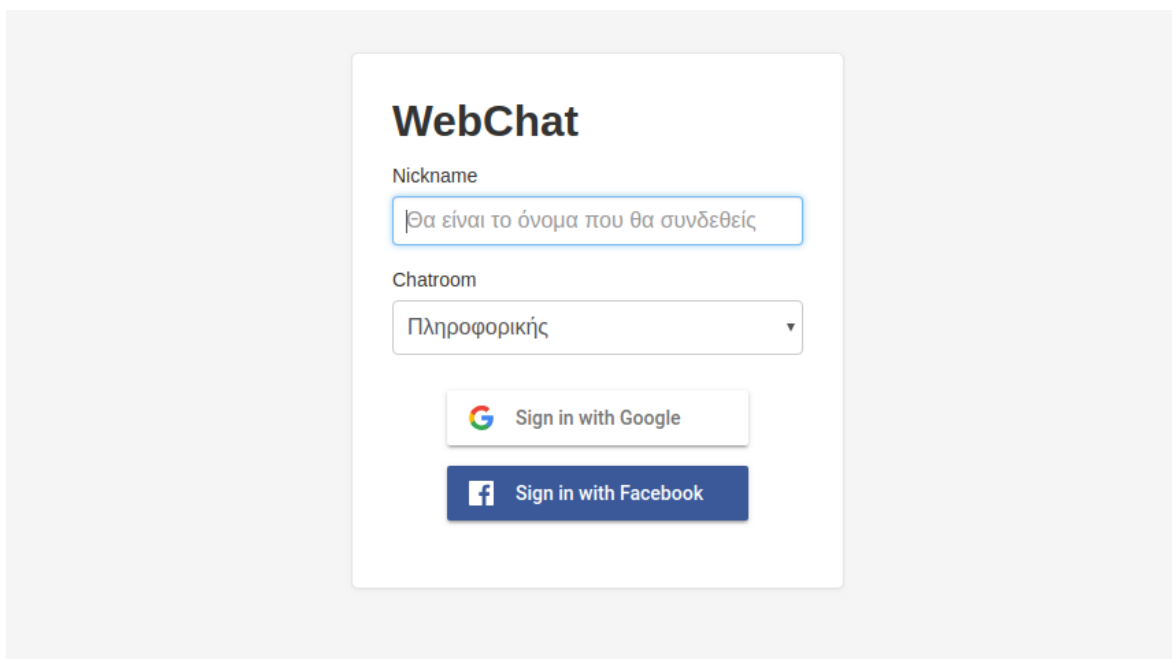
6.1. Εισαγωγή

Στην παρούσα ενότητα θα δούμε την εφαρμογή στην πράξη. Δημιουργήσαμε ένα σενάριο δύο χρηστών και των συνομιλιών τους, μέσα απ' το οποίο θα εξάγουμε τα συμπερασμάτα μας.

Ο χρήστης στην αρχή θα κάνει σύνδεση (login) μέσω του facebook, αφού πρώτα επιλέξει το γκρουπ του μαθήματος που επιθυμεί και αφού συνδεθεί επιτυχώς θα τον ανακατευθύνει στην σελίδα της εφαρμογής, όπου θα μπορεί να συνομιλεί με τους υπόλοιπους χρήστες.

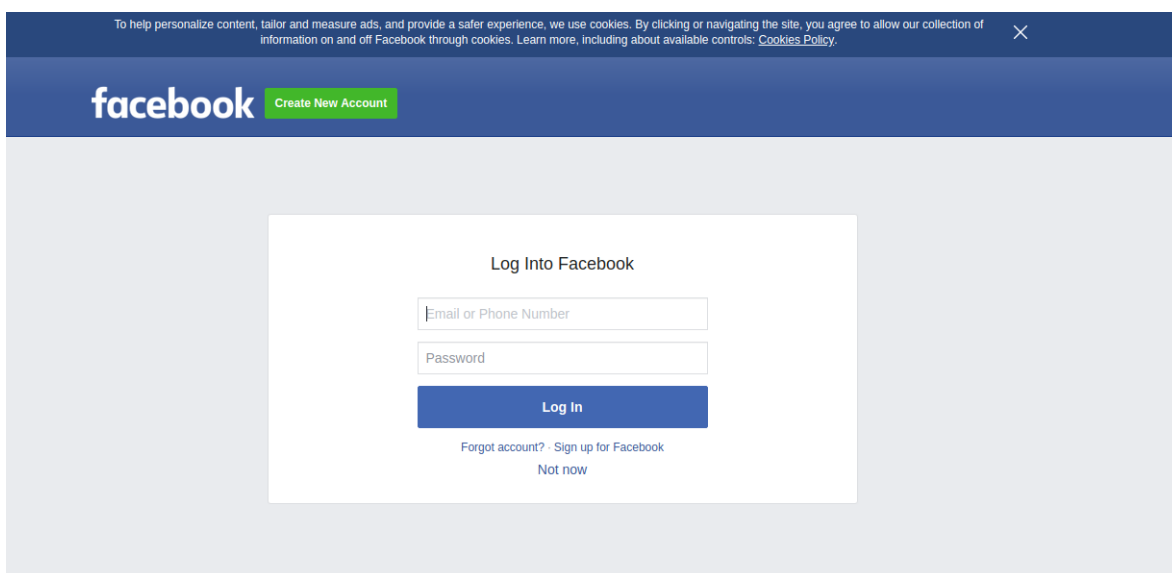
6.2. Σύνδεση χρήστη στην εφαρμογή

Ο χρήστης δεν χρειάζεται να επιλέξει το όνομα (username) για την εφαρμογή, ούτε να δώσει κωδικό password. Το authentication του χρήστη και η ταυτοποίησή του, θα γίνει μέσω των εφαρμογών του Facebook ή του Gmail. Το μόνο που θα επιλέξει είναι το γκρουπ στην περίπτωση μας, επιλέξαμε το γκρουπ της πληροφορικής.



The screenshot shows a registration form titled "WebChat". It includes a "Nickname" input field with the placeholder text "Θα είναι το όνομα που θα συνδεθείς". Below it is a "Chatroom" dropdown menu currently set to "Πληροφορικής". At the bottom, there are two buttons: "Sign in with Google" and "Sign in with Facebook".

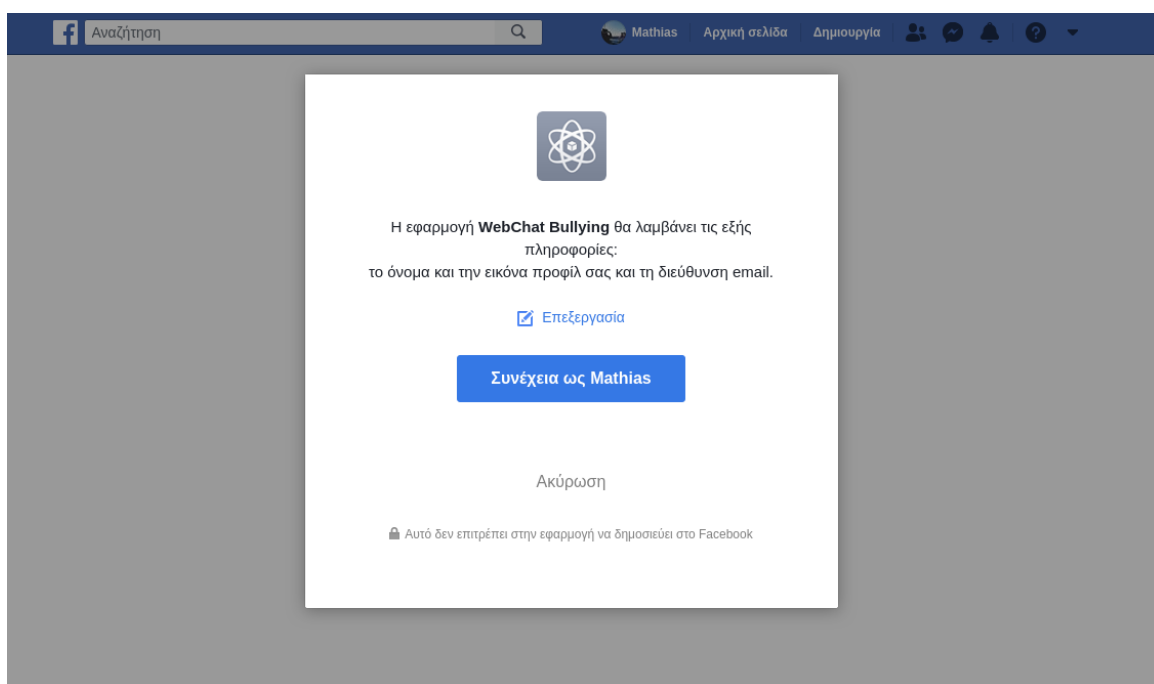
Υπάρχουν δύο εικονίδια ένα του Facebook και ένα του Gmail, ο χρήστης αφού επιλέξει και πατήσει πάνω στο εικονίδιο θα ανακατευθυνθεί για παράδειγμα στην σελίδα του Facebook, όπου θα πρέπει να δώσει τα στοιχεία του το username και το password του.



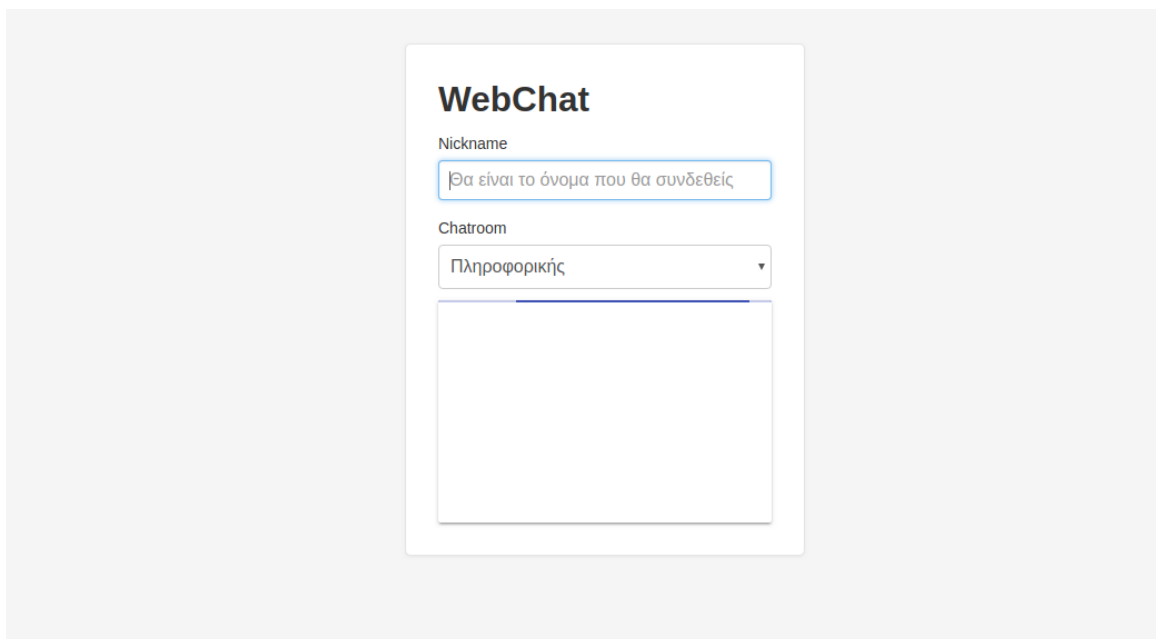
The screenshot shows the Facebook login page. At the top, there is a cookie consent banner. Below it is the Facebook logo and a "Create New Account" button. The main content is a "Log Into Facebook" form with two input fields: "Email or Phone Number" and "Password". Below the fields is a blue "Log In" button. At the bottom of the form, there are links for "Forgot account?" and "Sign up for Facebook", and a "Not now" option.

Σχεδιασμός, Ανάπτυξη και Εφαρμογή Προηγμένου Λογισμικού Διεπαφής χρηστών διαδικτυακών συζητήσεων (webChat), με τη χρήση της γλώσσας Java, για το σκοπό ανίχνευσης λέξεων της φυσικής γλώσσας που μαρτυρούν θυματοποίηση (Bullying).

Αφού κάνει login στην εφαρμογή του facebook ή ίδια η εφαρμογή θα του δείξει ότι το WebChat Bullying ζητάει κάποιες πληροφορίες, όπως το email του χρήστη το username του και την φωτογραφία του, όπου και θα πρέπει να πατήσει συνέχεια.



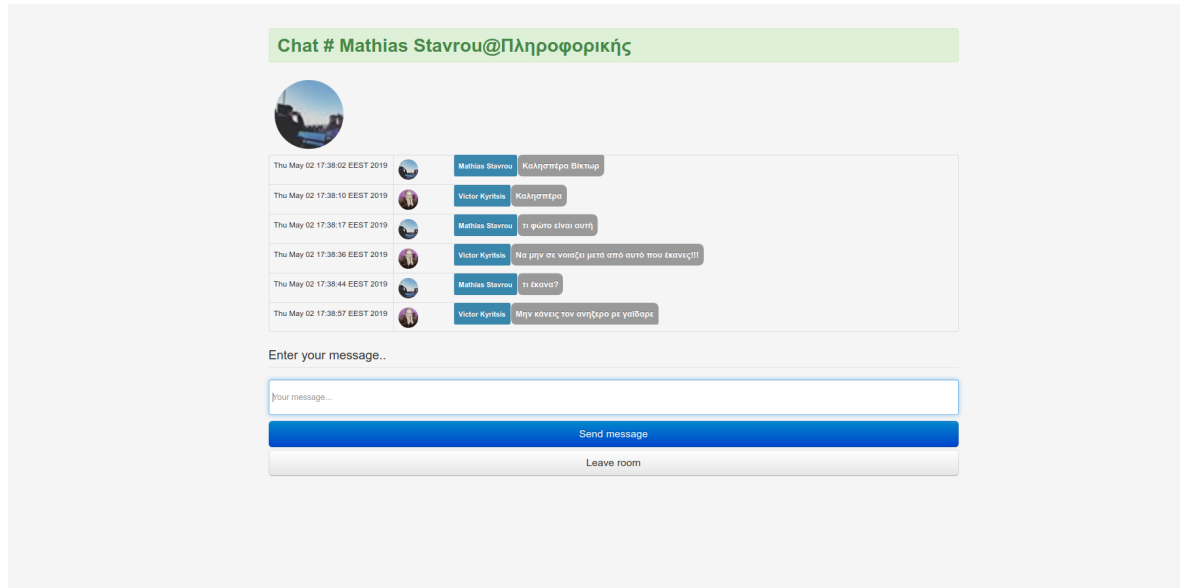
Έπειτα θα ανακατευθυνθεί στην εφαρμογή μας και θα μπορεί να συνομιλεί με άλλους χρήστες, οι οποίοι και βρίσκονται εκείνη τη στιγμή εντός της εφαρμογής.



6.3. Εικονική συνομιλία μεταξύ χρηστών.

Δημιουργήσαμε μία εικονική συνομιλία μεταξύ δύο χρηστών, με βάση τις λέξεις που χρησιμοποίησαν στην συνομιλία εφόσον ταυτίζονται με το λεξικό μας, αποθηκεύονται σε ένα αρχείο το οποίο αποστέλλεται στον δάσκαλο ή τον διαχειριστή του συστήματος.

Στην παρακάτω εικόνα υπάρχει η συνομιλία, καθώς και η καταγραφή σε αρχείο csv των λέξεων που δηλώνουν θυματοποίηση (bullying).



Αυτό που βλέπουμε είναι το username του χρήστη στην επικεφαλίδα, καθώς και το γκρουπ πληροφορικής στο οποίο και βρίσκεται, τα αντίστοιχα βλέπει και ο δεύτερος χρήστης απ την μεριά την δική του. Ακριβώς από κάτω είναι η φωτογραφία του χρήστη, και από κάτω κάθε γραμμή της συνομιλίας, όπου είναι σε μικρογραφία η φώτο το username απ το facebook του χρήστη, ημερομηνία και ώρα καθώς και το μήνυμα, το οποίο και αναλύει εφαρμογή μας.

238	e92f0929-c190-4b44-b91c-49f6b7b549@Πληροφορικής	Τι θέλεις ρε και με εντολεις!!!	Victor Kyritsis		
239	eafe87a2-b113-4747-8152-1bf96d106b@Πληροφορικής	τι φώτο είναι αυτή	Mathias Stavrou		
240	eafe87a2-b113-4747-8152-1bf96d106b@Πληροφορικής	χαχαχαχα	Mathias Stavrou		
241	e92f0929-c190-4b44-b91c-49f6b7b549@Πληροφορικής	Μην μου ξαναμιλήσεις, Γέλω κιάλας	Victor Kyritsis		
242	3b839eec-760b-414b-b0a0-23a1ba27f@Πληροφορικής	Καλησπέρα Βίκτωρ	Mathias Stavrou		
243	1bf96d106b-4705-8b31-5090e6e2eb@Πληροφορικής	Καλησπέρα Βίκτωρ	Mathias Stavrou		
244	0d61ec11-6a4f-42f1-9160-9c0615deb2@Πληροφορικής	τι φώτο είναι αυτή	Mathias Stavrou		
245	828036d8-a6c7-423d-99c3-da25f5bcb@Πληροφορικής	Καλησπέρα	Victor Kyritsis		
246	0d61ec11-6a4f-42f1-9160-9c0615deb2@Πληροφορικής	τι φώτο είναι αυτή	Mathias Stavrou		
247	828036d8-a6c7-423d-99c3-da25f5bcb@Πληροφορικής	Να μην σε νοιάζει μετά από αυτό που έκανες!!!	Victor Kyritsis		
248	0d61ec11-6a4f-42f1-9160-9c0615deb2@Πληροφορικής	τι έκανα?	Mathias Stavrou		
249	828036d8-a6c7-423d-99c3-da25f5bcb@Πληροφορικής	Μην κάνεις τον ανήξερο με γαϊδαρε	Victor Kyritsis	02/06/19 11:16 AM	Το ποσοστό ομοιότητας είναι = 50.0 receiver
250					
251					
252					
253					
254					
255					
256					
257					
258					
259					
260					
261					
262					

Σχεδιασμός, Ανάπτυξη και Εφαρμογή Προηγμένου Λογισμικού Διεπαφής χρηστών διαδικτυακών συζητήσεων (webChat), με τη χρήση της γλώσσας Java, για το σκοπό ανίχνευσης λέξεων της φυσικής γλώσσας που μαρτυρούν θυματοποίηση (Bullying).

Η λέξη στο σημείο αυτό που υποδηλώνει θυματοποίηση είναι η λέξη 'ρε γαιδαρέ' απο τον χρήστη και η οποία αφού αναλύθηκε αποθηκεύτηκε σε αρχείο csv, εκφράζοντας ένα ποσοστό ομοιότητας 50 τοις εκατό, στο λεξικό μας υπάρχει η λέξη 'Γαιδούρι'.

Στην πρώτη στήλη βλέπουμε το session id του χρήστη στην δεύτερη στήλη το γκρουπ, στην τρίτη στήλη το μήνυμα, στην τέταρτη το όνομα του χρήστη , στην πέμπτη ημερομηνία και ώρα και στην έκτη το ποσοστό ανάλυσης ομοιότητας μετά την ανάλυση απο τον αλγοριθμό.

Τέλος το αρχείο αυτό θα λάβει ο δάσκαλος ή ο διαχειριστής του συστήματος, προκειμένου να κρίνει εάν ο μαθητής, φοιτητής έχει προχωρήσει σε bullying εναντίον συμμαθητή του.

7. Συμπεράσματα.

Στην παρούσα εργασία με τη χρήση του εργαλείου της διαδικτυακής εφαρμογής, θα μπορούν οι χρήστες να συνομιλούν μεταξύ τους στο πλαίσιο ενός γκρουπ. Θα μπορεί να αποφευχθεί η θυματοποίηση ενός ατόμου – μαθητή από έναν συνομιληκό του. Τον τελευταίο λόγο τον έχει ο δάσκαλος όπου θα μπορεί να ενημερώσει τους γονείς, ουσιαστικά να μην υπάρξει επανάληψη του περιστατικού θυματοποίησης (bullying). Με αυτό τον τρόπο δημιουργείται η πρόληψη τέτοιων φαινομένων.

Το πεδίο των κοινωνικών επιστημών, είναι τεράστιο όπως και η θυματοποίηση που υφίστανται αρκετοί άνθρωποι καθημερινά, όπου αισθάνονται αίσθημα ντροπής. Δεν το αναφέρουν στο δάσκαλο, ή ακόμα και σε κάποια γραμμή υποστήριξης τέτοιων φαινομένων. Το διαδικτυακό εργαλείο, προσπαθεί να προβλέψει και να προφυλάξει τα παιδιά αυτά.

Σε ένα πλαίσιο διδακτορικής διατριβής και ενδελεχούς επιτόπιας έρευνας σε ένα σχολείο, θα μπορεί το εργαλείο αυτο να εξελιχθεί και να μετατραπεί σε ένα εργαλείο πρόληψης. Μπορεί να γίνω αυτό μέσω της μηχανικής μάθησης, για παράδειγμα το λεξικό θα δημιουργείται αυτόματα μέσω της τεχνητής νοημοσύνης.

Θα μπορέσουν να χρησιμοποιηθούν οι αλγόριθμοι αυτοί οι οποίοι θα μπορούν να μαθαίνουν απ τα δεδομένα και να κάνουν προβλέψεις, ή να εξάγουν αποφάσεις που θα μπορούν να εκφραστούν ως προς το αποτέλεσμα.

8. Βιβλιογραφία

1. Dan Olweus. Εκφοβισμός Και Βία Στο Σχολείο. 1999
2. Ελένη Δ. Τρίγκα-Μερτίκα. Ενδοσχολική Βία και σχολικός εκφοβισμός. Αθήνα 2015
3. Η συμβολή της Συναισθηματικής Νοημοσύνης στην πρόληψη του Σχολικού Εκφοβισμού. (2012, January). Retrieved from http://lyk-ialys.dod.sch.gr/attachments/article/8/EQ_and_bullying.pdf
ΓΕΝΙΚΟ ΛΥΚΕΙΟ ΙΑΛΥΣΟΥ ΕΡΕΥΝΗΤΙΚΗ ΕΡΓΑΣΙΑ Α' ΛΥΚΕΙΟΥ
4. Victim, (n.d.). Retrieved from <http://eyesonbullying.org/victim.html>
5. BULLYING – ΣΧΟΛΙΚΟΣ ΕΚΦΟΒΙΣΜΟΣ, Πανεπιστήμιο Μακεδονίας, Μάρτιος, 2016
http://bullying.uom.gr/files_eiconf/Bullying-praktika%2027-3-2018.pdf
6. How to Prevent Bullying. (n.d.). Retrieved from <https://www.stopbullying.gov/prevention/index.html>
7. Lyness, D. (Ed.). (2013, July). Helping Kids Deal With Bullies (for Parents). Retrieved from <https://kidshealth.org/en/parents/bullies.html>
8. Διαδικτυακός εκφοβισμός. (n.d.). Retrieved from https://el.wikipedia.org/wiki/Διαδικτυακός_εκφοβισμός#Μορφές_Διαδικτυακού_εκφοβισμού
9. Görzig, A., & Frumkin, L. A. (2013, March 01). Cyberbullying experiences on-the-go: When social media can become distressing. Retrieved from <https://cyberpsychology.eu/article/view/4283/3328>
10. Deptment, P. E. (n.d.). Retrieved from http://www.pi.ac.cy/InternetSafety/kindinoi_akatalperiex.html
11. Grooming. Retrieved from http://www.help-line.gr/wp-content/uploads/2015/01/Brochure_Grooming_web.pdf
12. Java. (2018, November 11). Retrieved from <https://el.wikipedia.org/wiki/Java>

13. Java Platform, Enterprise Edition. (2013, March 21). Retrieved from https://el.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition
14. XML. (2017, May 07). Retrieved from <https://el.wikipedia.org/wiki/XML>
15. WebSocket. (2018, November 13). Retrieved from <https://el.wikipedia.org/wiki/WebSocket>
16. Firebase. (2019, January 01). Retrieved from <https://en.wikipedia.org/wiki/Firebase>
17. Git (λογισμικό). (2017, December 19). Retrieved from [https://el.wikipedia.org/wiki/Git_\(λογισμικό\)](https://el.wikipedia.org/wiki/Git_(λογισμικό))
18. Spring Framework. (2019, January 08). Retrieved from https://en.wikipedia.org/wiki/Spring_Framework
19. HTML. (2018, December 31). Retrieved from <https://el.wikipedia.org/wiki/HTML>
20. Καργιοφύλλης, Α. (n.d.). Εισαγωγή. Retrieved from <https://www.wlearn.gr/index.php/css-84>
21. CSS. (2018, September 26). Retrieved from <https://el.wikipedia.org/wiki/CSS>
22. JavaScript. (2019, February 27). Retrieved from <https://el.wikipedia.org/wiki/JavaScript>
23. Menesini, E., Nocentini, A., Palladino, B., & Blaya, C. (n.d.). Definitions of cyberbullying. Retrieved January/February, 2013, from https://www.researchgate.net/publication/262584397_Definitions_of_cyberbullying