



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
Π.Μ.Σ. «Επιστήμη Υπολογιστών»

Πλατφόρμα αναζήτησης και παρακολούθησης αποστολών ταχυμεταφορών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΠΑΠΑΓΕΩΡΓΙΟΥ ΑΛΕΞΑΝΔΡΟΣ

A.M.: 2022202002020

Επιβλέπων Καθηγητής:

ΚΩΝΣΤΑΝΤΙΝΟΣ ΒΑΣΙΛΑΚΗΣ

Τρίπολη | Νοέμβριος 2022

Αφιερώνεται στον πατέρα μου....

Περιεχόμενα

Ευρετήριο Εικόνων	5
Ευρετήριο Πινάκων.....	7
Ευχαριστίες.....	8
Περίληψη	9
ABSTRACT.....	10
1 Εισαγωγή.....	11
2 Υπηρεσίες και τεχνολογίες διαδικτύου	13
2.1 Βασικά στοιχεία διαδικτύου, HTTP Request-Response.....	13
2.2 Υπηρεσίες διαδικτύου	16
2.3 REST (REpresentational State Transfer).....	17
2.3.1 Κανόνες ονοματολογίας	19
2.3.2 Κανόνες ασφαλείας	20
3 Δομικά στοιχεία ιστοσελίδας και προγραμματιστικά εργαλεία	23
3.1 Hyper Text Markup Language (HTML)	23
3.2 Cascading Style Sheet (CSS)	26
3.3 Bootstrap	30
3.4 JavaScript	31
3.5 PHP.....	34
3.6 Βάσεις δεδομένων	38
4 Ταχυμεταφορές, τρόπος λειτουργίας, σχεδιασμός εφαρμογής	41
4.1 Συνοδευτικού Δελτίου Ταχυμεταφοράς ΣΥ.ΔΕ.ΤΑ.....	42
4.1.1 Διεθνής Αριθμός Παρακολούθησης	43
4.1.2 Ραβδωτός ή γραμμωτός κώδικας	46
4.1.3 QR κωδικός	46
4.2 Δεδομένα αποστολών, σχεδιασμός βάσης	48
4.3 Πρόσβαση χρηστών	56
5 Τεχνικές και τεχνολογίες προγραμματισμού.....	59
5.1 Asynchronous JavaScript and XML (AJAX)	59

5.2	JavaScript Object Notation (JSON), αποστολή και λήψη μέσω cURL.....	62
5.3	Διαδικασία πιστοποίησης (Authentication) χρήστη.....	66
5.4	Τεκμηρίωση API συστήματος	72
6	Σενάρια Χρήσης	73
6.1	Περιήγηση στην ιστοσελίδα	73
6.2	Εγγραφή και είσοδος χρηστών	79
6.3	Χώρος διαχειριστή-πιστοποιημένων χρηστών.....	82
6.4	Χώρος Μεταφορέα	88
7	Συμπεράσματα.....	90
8	Πηγές – Βιβλιογραφία	91

Ευρετήριο Εικόνων

Εικόνα 1 Επικοινωνία client-server.....	14
Εικόνα 2 Σύνολο οντοτήτων.....	19
Εικόνα 3 Δομικά στοιχεία ιστοσελίδας.....	23
Εικόνα 4 Δομή ιστοσελίδας	24
Εικόνα 5 Παραδείγματα εννοιών για την PHP.....	38
Εικόνα 6 Επίπεδα βάσης δεδομένων	38
Εικόνα 7 Ενδεικτική εικόνα συνοδευτικού δελτίου ταχυμεταφοράς	43
Εικόνα 8 Εφαρμογές, χρήστες και διασυνδέσεις	47
Εικόνα 9 Διάγραμμα οντοτήτων - συσχετίσεων	48
Εικόνα 10 Σχεσιακό μοντέλο.....	52
Εικόνα 11 Είσοδος χρηστών.....	56
Εικόνα 12 Πρόσβαση μεταφορέα με πιστοποίηση περιηγητή	57
Εικόνα 13 Αλληλουχία γεγονότων στην υλοποίηση με Ajax.....	59
Εικόνα 14 Ασύγχρονη επικοινωνία client-server.....	61
Εικόνα 15 Έλεγχος και είσοδος χρήστη ως μέλος.	66
Εικόνα 16 Είσοδο με χρήση OAuth2 credentials.....	67
Εικόνα 17 Δημιουργία διαπιστευτηρίων OAuth2 στην πλατφόρμα της Google	68
Εικόνα 18 Δημιουργία διαπιστευτηρίων OAuth2 στην πλατφόρμα Facebook.....	71
Εικόνα 19 Αρχική σελίδα περιήγησης ιστοχώρου	73
Εικόνα 20 Αναζήτηση πακέτου τρίτης εταιρίας.	74
Εικόνα 21 Αναζήτηση πακέτου που διεκπεραιώνεται από τον ίδιο τον ιστότοπο.....	75
Εικόνα 22 Το πακέτο δεν βρέθηκε, εμφάνιση σχετικού λάθους.	75
Εικόνα 23 Ενημέρωση και αποδοχή των cookies, όπως προβλέπεται από τον General Data Protection Regulation (GDPR).....	76
Εικόνα 24 Οροι χρήσης ιστοσελίδας , Terms Of Services.....	76
Εικόνα 25 Πληροφορίες για τα Cookies	77
Εικόνα 26 Πληροφορίες εργαλείων ανάπτυξης και συντελεστών.....	77
Εικόνα 27 Πληροφορίες για το προσφερόμενο API	78
Εικόνα 28 Επικοινωνία με διαχειριστές του ιστοχώρου	78
Εικόνα 29 Είσοδος χρήστη με χρήση Κωδικών Facebook	79
Εικόνα 30 Είσοδος χρηστών μέσω Google	80
Εικόνα 31 Δημιουργία νέου χρήστη.....	80
Εικόνα 32 Υπενθύμιση κωδικού πρόσβασης.....	81
Εικόνα 33 Ανάκτηση κωδικού πρόσβασης	81
Εικόνα 34 Έκδοση ΣΥ.ΔΕ.ΤΑ.	82
Εικόνα 35 Επιλογή εκτύπωσης Δελτίου.....	83
Εικόνα 36 Εκτύπωση ΣΥ.ΔΕ.ΤΑ.....	84
Εικόνα 37 Διαχείριση Αποστολών και Παραληπτών	84
Εικόνα 38 Διαχείριση Μηνυμάτων επισκεπτών	85
Εικόνα 39 Διαχείριση καταστάσεων δεμάτων	85

Εικόνα 40 Διαχείριση προφίλ χρήστη.....	86
Εικόνα 41 Κλειδί API	86
Εικόνα 42 Παράμετροι Facebook API και Google API	87
Εικόνα 43 Παράμετροι αποστολής μηνυμάτων ηλεκτρονικού ταχυδρομείου	87
Εικόνα 44 Συγχρονισμός API server με 17Track	87
Εικόνα 45 Πρόσβαση μη πιστοποιημένου browser στον χώρο μεταφορέων	88
Εικόνα 46 Πιστοποίηση browser μεταφορέα.....	88
Εικόνα 47 Tracking και αλλαγή κατάστασης δέματος από μεταφορέα.....	89

Ευρετήριο Πινάκων

Πίνακας 1 HTTP request methods	15
Πίνακας 2 Server response	15
Πίνακας 3 Κωδικοί αποκρίσεων πρωτοκόλλου HTTP.....	19
Πίνακας 4 Βασικές ετικέτες της HTML.....	25
Πίνακας 5 Λίστες και πίνακες στην HTML	25
Πίνακας 6 Βασικότερα γεγονότα (events) στην HTML.....	26
Πίνακας 7 Παραδείγματα CSS.....	29
Πίνακας 8 Βασικές ιδιότητες css.....	30
Πίνακας 9 Βασικά στοιχεία της Javascript	34
Πίνακας 10 Βασικά στοιχεία της PHP	36
Πίνακας 11 Βασικά Σύμβολα διαγράμματος Οντοτήτων Συσχετίσεων	40
Πίνακας 12 Σημειογραφία Crow's foot	40
Πίνακας 13 Πρότυπο UPU S10.....	44
Πίνακας 14 Παρεχόμενες υπηρεσίες, 1 ^ο και 2 ^ο ψηφίο προτύπου S10	45
Πίνακας 15 Πίνακας contactus	49
Πίνακας 16 Πίνακας senders	50
Πίνακας 17 Πίνακας receivers	50
Πίνακας 18 Πίνακας sydetata	51
Πίνακας 19 Πίνακας checkpoints.....	51
Πίνακας 20 Πίνακας users	51
Πίνακας 21 Πίνακας sydetacheckpoints	52
Πίνακας 22 Συσχετίσεις πινάκων εφαρμογής	53
Πίνακας 23 Δικαιώματα χρηστών ανά περίπτωση.....	58
Πίνακας 24 Βασικές μέθοδοι του αντικειμένου XMLHttpRequest	60
Πίνακας 25 Βασικές ιδιότητες του αντικειμένου XMLHttpRequest.....	61
Πίνακας 26 Ομάδες συναρτήσεων υλοποίησης ασύγχρονης επικοινωνίας.....	62
Πίνακας 27 Βασικές παράμετροι curl_setopt()	64

Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής μου εργασίας, αισθάνομαι την ανάγκη να ευχαριστήσω ειλικρινά τον επιβλέποντα Καθηγητή, κ. Βασιλάκη Κωνσταντίνο, για την εμπιστοσύνη που μου έδειξε να προχωρήσω στην εκπόνηση του παρόντος θέματος, καθώς και για την εξαιρετική συνεργασία και την καθοδήγηση που μου παρείχε μέσα από τις επικοινωνητικές παρατηρήσεις του. Θα ήθελα επίσης, να ευχαριστήσω όλους τους καθηγητές του προγράμματος, για την καθοδήγηση και τη συμβολή του καθενός στην ολοκλήρωση των σπουδών μου. Τέλος, ευχαριστώ θερμά την οικογένειά μου για την πολύτιμη υποστήριξη που μου προσέφεραν με κάθε τρόπο, για να καταφέρω να πραγματοποιήσω τον στόχο μου.

Περίληψη

Η παρούσα διπλωματική εργασία εκπονήθηκε, στα πλαίσια της ολοκλήρωσης των σπουδών μου, στο Μεταπτυχιακό Πρόγραμμα Σπουδών «Επιστήμη Υπολογιστών» του τμήματος Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Πελοποννήσου. Σκοπός της παρούσας διπλωματικής εργασίας, αποτελεί η μελέτη, ο σχεδιασμός και η υλοποίηση του συστήματος αναζήτησης και παρακολούθησης αποστολών που εκτελούνται από ταχυμεταφορές. Ο σχεδιασμός του συστήματος έχει πραγματοποιηθεί με τρόπο ώστε να προάγεται η συνεργασία με τα υπάρχοντα συστήματα ταχυμεταφορών με σκοπό την παρακολούθηση αποστολών ανεξάρτητα από την εταιρία που το έχει αναλάβει και ταυτόχρονα να αποτελεί εργαλείο για κάποιον που θέλει να αναπτύξει τη δική του ταχυμεταφορική εταιρία. Η συνεργασία του συστήματος με τα υπόλοιπα συστήματα έχει δομηθεί με βάση την αρχιτεκτονική REST, καθώς αυτή αποτελεί πιο διαδεδομένο αρχιτεκτονικό μοτίβο στην ανάπτυξη web εφαρμογών. Συνεπώς το σύστημα θα μπορεί να αιτείται μέσω κλήσεων προγραμματιστικών διεπαφών (API), πληροφορίες από άλλα συστήματα, αλλά ταυτόχρονα μπορεί και το ίδιο να παρέχει διεπαφή που θα είναι σε θέση να εξυπηρετεί τα αιτήματα άλλων συστημάτων. Η υλοποίηση του συστήματος πραγματοποιήθηκε με χρήση HTML5, CSS3, PHP, MYSQL και JAVASCRIPT.

Λέξεις – κλειδιά: παρακολούθηση αποστολών, δικτυακή εφαρμογή, API, Application Programming, web service.

ABSTRACT

The goal of this M.Sc. thesis is the study, the design and the implementation of a system for tracking shipments that are delivered through courier services. The system design has been formulated in a way that supports the cooperation with existing courier service information systems, aiming to be able to allow tracking of shipments regardless of the specific courier service that has undertaken the delivery, while at the same time the system can be used as a standalone information system, to support the operation of a new courier service. The interoperability of the developed system with third-party systems is based on the REST architectural pattern, which is nowadays the predominant approach for developing web-based applications. The system can also provide its own REST API, in order to accept requests from third-party systems and respond to them. The system has been implemented using the HTML5, CSS3, PHP , MYSQL και JAVASCRIPT technologies.

Keywords: tracking courier shipments, web application, API, Application Programming, web service

1 Εισαγωγή

Το διαδίκτυο ξεκίνησε από ένα μικρό δίκτυο υπολογιστών και σήμερα έχει καταφέρει να είναι εργαλείο οικονομικής ανάπτυξης, πληροφόρησης επικοινωνίας και κοινωνικής δικτύωσης. Από ολοκληρωμένες εφαρμογές με αποκλειστική χρήση από τους πελάτες των εταιριών-ιδιοκτητών έχουμε φτάσει σε υπηρεσίες παρεχόμενες επί πληρωμή ή ελεύθερες, οι οποίες είναι ικανές να ανταλλάσσουν δεδομένα και πληροφορίες και ανά ομάδες να συγκροτούν νέες εφαρμογές. Βασικός παράγοντας για αυτή την εξέλιξη είναι η ανάπτυξη της τεχνολογίας των υπηρεσιών διαδικτύου (web services) για την ανταλλαγή πληροφοριών και δεδομένων ανάμεσα στα πληροφοριακά συστήματα ώστε να μπορούν προγραμματιστές από όλο τον κόσμο να συνθέτουν νέες εφαρμογές κάνοντας χρήση διαφορετικών εργαλείων και γλωσσών προγραμματισμού. Κάθε Web Service ως διεπαφή λογισμικού (software interface) που περιγράφει ένα σύνολο λειτουργιών έχει ένα API (Application Programming Interface) το οποίο παίζει τον ρόλο του διαμεσολαβητή ανάμεσα στην υπηρεσία και τον πελάτη που θέλει να την χρησιμοποιήσει.

Η πλατφόρμα που θα παρουσιαστεί στα επόμενα κεφάλαια έχει ως στόχο να εκτελεί αναζητήσεις αποστολών από τις εταιρίες ταχυμεταφορών μέσω του παρεχόμενου από αυτές API. Πολλές από τις εταιρίες αυτές δεν διαθέτουν το API τους προς ελεύθερη χρήση από άλλες εφαρμογές και ιστοσελίδες. Ταυτόχρονα υπάρχει πολύ μεγάλος αριθμός εταιριών ανά τον κόσμο. Για τους παραπάνω λόγους η πλατφόρμα θα συνεργάζεται με τον ιστότοπο 17TRACK.NET το οποίο παρέχει ελεύθερα το API του και ταυτόχρονα εξυπηρετεί πάνω από 900 ταχυμεταφορικές εταιρίες παγκοσμίως. Παράλληλα με την παραπάνω λειτουργία, η πλατφόρμα μας θα μπορεί να λειτουργήσει αυτόνομα ως εφαρμογή διαχείρισης αποστολών παρέχοντας δικό της API για τη δημιουργία συνοδευτικών δελτίων αποστολών και παρακολούθησής τους σε όλα τα στάδια. Έτσι θα μπορεί να χρησιμοποιείται από μία εταιρία ταχυμεταφορών δίνοντας τη δυνατότητα διασύνδεσης ηλεκτρονικών καταστημάτων στην εφαρμογή.

Στο κεφάλαιο 2 περιγράφεται το θεωρητικό υπόβαθρο των τεχνολογιών που χρησιμοποιούνται στην εφαρμογή που αναπτύχθηκε. Θα αναφερθούμε στον παγκόσμιο ιστό (World Wide Web) ως μέσο ανταλλαγής πληροφοριών και τα πρωτόκολλα που χρησιμοποιούνται προκειμένου να υλοποιηθεί η προαναφερθείσα ανταλλαγή. Θα αναφερθούμε στα Web Services ως τεχνολογία επικοινωνίας διακριτών συστημάτων που έχουν υλοποιηθεί με διαφορετικά εργαλεία και εκτελούνται σε διαφορετικό περιβάλλον υλικού και λογισμικού και τα οποία περιλαμβάνουν μία διεπαφή εφαρμογής (software interface) για χρήση από τρίτα συστήματα και εφαρμογές. Τέλος θα αναφερθούμε στο αρχιτεκτονικό στυλ REST API (REpresentational State Transfer Application Programming Interface) πάνω στο οποίο θα στηριχθεί η λειτουργία της πλατφόρμας μας.

Στο κεφάλαιο 3 θα περιγράψουμε τα εργαλεία υλοποίησης της εφαρμογής. Συγκεκριμένα θα δούμε βασικά στοιχεία της HyperText Markup Language (HTML), της CSS (Cascading Style Sheet), της Javascript αλλά και της PHP. Καθώς η εφαρμογή στηρίζεται στην ασύγχρονη επικοινωνία πελάτη-εξυπηρετητή θα δούμε τον βασικό τρόπο λειτουργίας της τεχνολογίας AJAX ως ένα σύνολο τεχνικών αποστολής και ανάκτησης πληροφοριών με αιτήματα που απευθύνονται από την εφαρμογή ιστού προς τον εξυπηρετητή (server) στον οποίο εκτελείται η επιχειρηματική λογική της εφαρμογής.

Στο κεφάλαιο 4 θα αναλύσουμε τις ανάγκες που θα πρέπει να καλύπτει η εφαρμογή, τα βασικά συστατικά της αλλά και τον τρόπο με τον οποίο θα σχεδιαστεί η υλοποίησή της. Θα περιγράψουμε τους κανόνες που διέπουν μία ταχυμεταφορική εταιρεία, αλλά και τον τρόπο που αυτοί οι κανόνες υλοποιούνται προγραμματιστικά. Θα θέσουμε το πλαίσιο λειτουργίας επιμέρους τμημάτων όπως την πρόσβαση των χρηστών την αναπαραγωγή και χρήση QRcode και barcode αλλά και τον σχεδιασμό της βάσης δεδομένων στην οποία θα αποθηκεύονται τα δεδομένα μας

Στο κεφάλαιο 5 θα περιγράψουμε προγραμματιστικές τεχνικές που έχουμε εφαρμόσει κατά την ανάπτυξη της εφαρμογής. Θα αναλύσουμε τμήματα του κώδικα που παρουσιάζουν ενδιαφέρον για τη λειτουργικότητα του συστήματος. Τέλος θα περιγράψουμε τις λειτουργίες API, μαζί με παραδείγματα για τη διασύνδεση της πλατφόρμας με άλλες ιστοσελίδες ως αυτόνομο πρόγραμμα διαχείρισης αποστολών.

Στο κεφάλαιο 6 θα παρουσιάσουμε σενάρια χρήσης της πλατφόρμας μέσα από τις παραγόμενες φόρμες που απεικονίζουν την λειτουργικότητα της εφαρμογής. Τα παραπάνω θα γίνουν για τα εξής σενάρια χρήσης: (1) αναζήτησης αποστολής πακέτου από τρίτη εταιρία, (2) κεντρική διαχείριση αποστολών που παράγονται από την αυτόνομη λειτουργία της στα πλαίσια εταιρίας και τέλος (3) τη διασύνδεση που μπορεί να παρέχει σε e-shops μέσω του API.

Τέλος στο κεφάλαιο 7 παρατίθεται μία σύνοψη της πτυχιακής εργασίας και καταγράφονται πιθανές μελλοντικές επεκτάσεις.

2 Υπηρεσίες και τεχνολογίες διαδικτύου

Σε αυτό το κεφάλαιο θα περιγράψουμε βασικές έννοιες και διαδικασίες που σχετίζονται με τον παγκόσμιο ιστό (World Wide Web), τις υπηρεσίες διαδικτύου (web services) και το αρχιτεκτονικό μοτίβο REST API (REpresentational State Transfer Application Programming Interface) με βάση το οποίο έχει δομηθεί η λειτουργία της πλατφόρμας μας.

2.1 Βασικά στοιχεία διαδικτύου, HTTP Request-Response

Ο παγκόσμιος ιστός είναι ένα πληροφοριακό σύστημα που επιτρέπει την πρόσβαση σε ηλεκτρονικά έγγραφα και υπηρεσίες πάνω από το διαδίκτυο. Τα ηλεκτρονικά έγγραφα περιλαμβάνουν εκτός από κείμενο και πολυμέσα (βίντεο, ήχος, εικόνες) ή αρχεία με άλλες πληροφορίες (π.χ. σχέδια, γεωγραφικούς χάρτες κ.ο.κ.). Κάθε έγγραφο μπορεί, με την σειρά του, να οδηγεί εκ νέου σε νέα ηλεκτρονικά αρχεία. Έτσι δημιουργούνται οι ιστοσελίδες οι οποίες παρουσιάζουν διαδραστικό υλικό και στις οποίες περιηγούμαστε μέσα από τον φυλλομετρητή μας. Τα βασικά συστατικά του web είναι οι web servers, οι χρήστες-πελάτες και τα πρωτόκολλα επικοινωνίας μεταξύ τους.

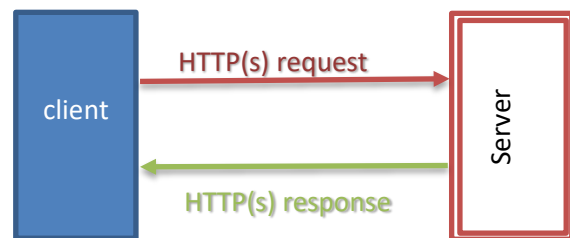
Με τον όρο “Web servers” αναφερόμαστε σε εφαρμογές που είναι εγκατεστημένες στο ένα άκρο της επικοινωνίας, σε ένα κατάλληλο υπολογιστή (που μπορεί επίσης να ονομάζεται “web server” ή απλά server) και βρίσκονται σε συνεχή λειτουργία με σκοπό τη λήψη, επεξεργασία και απάντηση αιτημάτων που υποβάλλονται από τους χρήστες-πελάτες (clients). Σημειώνεται ότι οι χρήστες-πελάτες μπορεί να είναι άνθρωποι που περιηγούνται στο διαδίκτυο χρησιμοποιώντας εφαρμογές πλοήγησης, ή άλλες οντότητες λογισμικού, που ανακτούν σελίδες και υπηρεσίες στο πλαίσιο κάποιας αυτοματοποιημένης επεξεργασίας. Οι servers είναι σε θέση να διευθετούν μία σειρά θεμάτων όπως διαμόρφωση και αποστολή ιστοσελίδων, διαχείριση αρχείων, διαχείριση email και πολλών άλλων υπηρεσιών. Ο client επικοινωνεί με τον server μέσω ειδικών προγραμμάτων όπως φυλλομετρητών, προγράμματα διαχείρισης ταχυδρομείου, αρχείων κλπ. Η επικοινωνία αυτή γίνεται κάνοντας χρήση του πρωτοκόλλου Http ή το κρυπτογραφημένο ανάλογό του, το Https. Το Http/Https είναι ένα πρωτόκολλο μεταφοράς δεδομένων και πληροφοριών μεταξύ δύο συστημάτων, το οποίο χρησιμοποιείται ευρέως στην επικοινωνία των υπολογιστών μέσω διαδικτύου. Η βασική λειτουργία του είναι η εγκαθίδρυση επικοινωνίας από τον client στον Server, η αποστολή ερωτημάτων μέσω του καναλιού επικοινωνίας και η ανάκτηση της απάντησης από αυτόν. Τα ερωτήματα και οι απαντήσεις προς και από τον server μπορεί να περιλαμβάνουν μεγάλο όγκο πληροφοριών και δεδομένων που απαιτούνται για την εκάστοτε εργασία. Το πρωτόκολλο Http χρησιμοποιεί τυπικά την θύρα 80 και δεν παρέχει ασφάλεια στην μετάδοση των δεδομένων, ενώ το Https χρησιμοποιεί τυπικά την θύρα 443 και περιλαμβάνει την κρυπτογράφηση των δεδομένων με τεχνολογία δημοσίου κλειδιού (PKI) με στόχο την ασφαλή ανταλλαγή δεδομένων. Η κρυπτογράφηση μπορεί να πραγματοποιείται με οποιονδήποτε κατάλληλο αλγόριθμο. Τα βασικά χαρακτηριστικά του πρωτοκόλλου είναι τα εξής:

- *Χωρίς σύνδεση (Connectionless)*, καθώς ο πελάτης client ξεκινά την επικοινωνία κάνοντας ένα αίτημα-request στον server και περιμένει από αυτόν την απάντηση. Όταν η παραπάνω επικοινωνία ολοκληρωθεί τότε ο client και ο server δεν έχουν κανένα συνδετικό κρίκο μέχρι να γίνει το επόμενο ερώτημα. Σημειώνεται εδώ ότι σε όρους διαδικτυακής επικοινωνίας εγκαθιδρύεται σύνδεση μεταξύ client και server (καθώς τυπικά η επικοινωνία πραγματοποιείται

με χρήση του πρωτοκόλλου TCP), ωστόσο ο χαρακτηρισμός *Χωρίς σύνδεση* καλύπτει τη συμπεριφορά του πρωτοκόλλου HTTP/HTTPS σε ευρύτερο χρονικό ορίζοντα.

- *Χωρίς μνήμη (Stateless)*. Σύμφωνα με το προηγούμενο, και αφού έχει ολοκληρωθεί η μεταξύ του client και του server επικοινωνία, το πρωτόκολλο δεν διατηρεί καμία πληροφορία για την επικοινωνία που να μπορεί να χρησιμοποιηθεί σε επόμενη επικοινωνία. Ο όρος αναφέρεται στο πρωτόκολλο και τις ιδιότητές του και όχι στο τι μπορεί να κάνει ο προγραμματιστής και πώς μπορεί να συνδυάσει πληροφορίες από διαδοχικά αιτήματα. Ιδιαίτερα γνωρίζουμε ότι οι προγραμματιστές χρησιμοποιούν τεχνικές ώστε οι servers να διατηρούν την κατάσταση της επικοινωνίας με τους clients, κυρίως μέσω της τεχνικής των συνόδων (sessions). Αυτή η διατήρηση ωστόσο γίνεται σε επίπεδο εφαρμογής, και όχι σε επίπεδο πρωτοκόλλου.
- *Ανεξάρτητο από τον τύπο των δεδομένων (media independent)*. Μέσω του πρωτοκόλλου είναι δυνατή η μεταφορά πληροφορίας όλων των δυνατών μέσων αρκεί ο client και ο server να εργάζονται πάνω στην ίδια μορφή μέσου. Χρησιμοποιώντας το MIME¹ type που περιγράφει το είδος του πόρου, client και server μπορούν να γνωρίζουν τον τύπο της μεταδιδόμενης πληροφορίας.

Τα αιτήματα (request) που αποστέλλονται από τον client στον server έχουν τέτοια μορφή που ο client ενημερώνει τον server με ποια μέθοδο του πρωτοκόλλου HTTP θέλει να επικοινωνήσει και ποια δεδομένα θα μετέχουν σε αυτό τον διάλογο. Ένα αίτημα (request) στέλνεται από τον client και με την σειρά του ο server απαντά με μία απόκριση (response).



Εικόνα 1 Επικοινωνία client-server

Το αίτημα (από τον client προς τον server) περιλαμβάνει τη μέθοδο του πρωτοκόλλου HTTP με την οποία θα γίνει η επικοινωνία, καθώς και μία συμβολοσειρά προσδιορισμού του πόρου, έχοντας την μορφή <μέθοδος> <συμβολοσειρά προσδιορισμού του πόρου> όπως η παρακάτω και αποτελείται από:

```
GET /v1/save HTTP/1.1
Host: www.mysite.xx
```

- Τον προσδιορισμό της μεθόδου, μέσω του οποίου ορίζεται η ενέργεια που πρέπει να γίνει. Οι βασικότερες μέθοδοι φαίνονται στον παρακάτω πίνακα.

Μέθοδος	Περιγραφή
GET	Με αυτή την μέθοδο ανακτά δεδομένα ο client από τον server χωρίς να προκαλεί καμία αλλαγή στον server. Όσες φορές και να εκτελεστεί η μέθοδος πάντα θα μας δίνει το ίδιο αποτέλεσμα.
HEAD	Είναι το ίδιο με την GET αλλά επιστρέφει μόνο την κεφαλίδα της απάντησης και όχι δεδομένα.
POST	Αποστέλλονται δεδομένα από τον client προς τον server για την επεξεργασία και αποθήκευσή τους στον τελευταίο.

¹ Το MIME type (Multipurpose Internet Mail Extensions or MIME type) υποδεικνύει τη φύση και τη μορφή ενός εγγράφου, αρχείου ή ακολουθίας από bytes ^[10]

PUT	Η πληροφορία που μεταφέρεται με την αίτηση, αποθηκεύεται στον server ως η νέα τιμή του αντικειμένου που προσδιορίζει το URL.
DELETE	Ο server διαγράφει το αντικείμενο που προσδιορίζει το URL.

Πίνακας 1 HTTP request methods

- Τη συμβολοσειρά προσδιορισμού του πόρου
- Τον προσδιορισμό της έκδοσης του πρωτοκόλλου (τυπικά HTTP/1.1).
- Τις επικεφαλίδες (headers) με τις οποίες μεταφέρεται πρόσθετη πληροφορία όπως (virtual) host, content-type, content-length, user-agent και άλλες.
- Τα πρόσθετα δεδομένα που αποστέλλονται από/προς τον server (ιδιαίτερα με την μέθοδο POST)

Σε επίπεδο αναπαράστασης, οι πληροφορίες για το αν χρησιμοποιείται HTTP ή HTTPS, ο προσδιορισμός του server καθώς και η συμβολοσειρά προσδιορισμού του πόρου περιγράφονται από το Url (Uniform Resource Locator) το οποίο έχει τη γενική μορφή.

<protocol>://<hostname>[:<port>][/<path>]

Κάθε αίτημα από οποιονδήποτε χρήστη πρέπει να έχει απάντηση από τον server. Η απάντηση από τον server (response) έχει την παρακάτω δομή και τα κύρια χαρακτηριστικά περιγράφονται στον παρακάτω πίνακα.

```

HTTP/1.1 200 OK
Date: Mon, 21 Sep 2019 18:46:30 GMT
Content-Length: 1924
Content-Type: text/html; charset=utf-8
{
  "RETURN": "FAILED",
  "MESSAGE": "Person not found in Blacklist"
}

```

Χαρακτηριστικό δομής	Περιγραφή
Status line	(HTTP/1.1 200 OK) Μας επιστρέφει έκδοση πρωτοκόλλου και status code.
Status Code	Εμφανίζεται στην status line και μας δείχνει αν ήταν επιτυχής το αίτημα ή αν προκλήθηκε κάποιο λάθος. Οι κωδικοί ομαδοποιούνται στους: <i>Informational (1xx)</i> <i>Successful (2xx)</i> <i>Redirection (3xx)</i> <i>Client error (4xx)</i> <i>Server error (5xx)</i>
Headers	Όλες οι υπόλοιπες πληροφορίες που σχετίζονται με το σύστημα και με το πακέτο των δεδομένων, εκτός από το σώμα των δεδομένων.
Message Body	Είναι τα δεδομένα τα οποία θα παραλάβει ο client για χρήση, π.χ. ο browser για προβολή στον τελικό χρήστη ή μία εφαρμογή, για περαιτέρω επεξεργασία.

Πίνακας 2 Server response

2.2 Υπηρεσίες διαδικτύου

Ο παγκόσμιος ιστός όπως τον περιγράψαμε πιο πάνω, κατέστησε δυνατή την ανάπτυξη πολλών και διαφορετικών εφαρμογών. Υπήρξαν όμως αρκετά ζητήματα, καθώς κάθε σύστημα και εφαρμογή είχε τη δική της αρχιτεκτονική και τρόπο επικοινωνίας με τρίτους. Αυτό ήταν μεγάλο πρόβλημα, αφού για να επιτευχθεί η επικοινωνία και συνεργασία μίας «κεντρικής» εφαρμογής (π.χ. εφαρμογής βιβλιοθήκης) με τρίτα συστήματα, έπρεπε το κάθε σύστημα-πελάτης να προσαρμοστεί στο «κεντρικό» και να ακολουθήσει τα πρωτόκολλα επικοινωνίας που του επέβαλε εκείνο. Η εισαγωγή των υπηρεσιών διαδικτύου (web services) αντιμετώπισε αυτά τα ζητήματα, προτυποποιώντας την επικοινωνία και τύπους/μορφές ανταλλαγής δεδομένων. Έτσι οι υπηρεσίες διαδικτύου (web services) γνωρίζουν μεγάλη ανάπτυξη και καθημερινά αυξάνει το εύρος των εφαρμογών και των πεδίων που αυτές καλύπτουν.

Για τον όρο «υπηρεσία διαδικτύου» (web service) έχουν δοθεί διαφορετικοί ορισμοί από τους οργανισμούς προτυποποίησης, τις εταιρείες και τους ειδικούς στο πεδίο. Όλοι όμως οι οργανισμοί συγκλίνουν στο ότι, πρόκειται για μία τεχνολογία η οποία επιτρέπει σε εφαρμογές να επικοινωνούν μεταξύ τους, χωρίς περιορισμούς που να αφορούν το υλικό, το λειτουργικό σύστημα ή τη γλώσσα προγραμματισμού στην οποία έχει αναπτυχθεί κάποιο από τα εμπλεκόμενα συστήματα. Περιλαμβάνει μία διεπαφή εφαρμογής (software interface) που υποστηρίζει την επικοινωνία διαφορετικών συστημάτων για υποβολή αιτημάτων εκτέλεσης υπηρεσιών και παραλαβή των σχετικών απαντήσεων. Συνοπτικά ο παραπάνω ορισμός αποδίδεται με τα εξής τέσσερα σημεία:

- Διάθεση μέσω internet ή intranet
- Τυποποίηση ανταλλαγής μηνυμάτων μέσω http
- Δυνατότητα κωδικοποίησης δεδομένων με διάφορα πρότυπα (π.χ. XML, JSON)
- Ανεξάρτητο υλικού, λειτουργικού συστήματος ή γλώσσας προγραμματισμού

Εν κατακλείδι το web service είναι εγκατεστημένο σε ένα server και καλείται από τις εφαρμογές-χρήστες. Το ίδιο το service αποτελεί το «χειριστήριο» της εφαρμογής αφού ο χρήστης με αυτό προσδιορίζει τις ενέργειες που ζητά από το πρόγραμμα. Τα αποτελέσματα του προγράμματος, θα μας επιστραφούν ως αποτελέσματα του service. Τα web service έχουν πλεονεκτήματα και μειονεκτήματα, τα βασικότερα των οποίων αναφέρονται πιο κάτω.

Πλεονεκτήματα των web services.

- Άμεση ενσωμάτωση: Για να μπορέσει να χρησιμοποιηθεί ένα web service από μία ιστοσελίδα ή εφαρμογή, δεν χρειάζεται να γίνει καμία αλλαγή στον μηχανισμό του συστήματος. Η ενσωμάτωση γίνεται σε επίπεδο software χωρίς αλλαγές (ή με ελάχιστες αλλαγές) σε αυτό.
- Επεκτασιμότητα: Η επέκταση ενός web service είναι πολύ εύκολη, αφού γίνεται σε κεντρικό επίπεδο παρέχοντας νέες δυνατότητες στους χρήστες του.
- Διαλειτουργικότητα: Είναι ανεξάρτητο από το υλικό, λειτουργικό σύστημα και το περιβάλλον του χρήστη. Εφόσον ο χρήστης-πρόγραμμα μπορεί να εκτελέσει αιτήματα HTTP, εύκολα μπορεί να ενσωματώσει οποιοδήποτε web service.
- Κόστος: Το web service παρέχει μία διεπαφή εφαρμογής. Αυτό σημαίνει ότι αν η εφαρμογή υπάρχει, το κόστος κατασκευής του χειρισμού της μέσω δικτύου είναι πολύ μικρό. Ομοίως και

για την εφαρμογή που θα ενσωματώσει το service. Υπάρχουν προτυποποιημένα εργαλεία που με εύκολο τρόπο δημιουργούν κώδικα για κλήση web services.

- Τέλος, με βάση την εμπειρία που έχει αποκτηθεί μέχρι τώρα, θα μπορούσαμε να πούμε, ότι ιστοσελίδες οι οποίες παρέχουν πληροφορίες βασιζόμενες σε web services είναι πιο ευέλικτες και εξελίξιμες, ενώ δύνανται να συνδυάζουν τις ροές πληροφορίας από διαφορετικά σημεία και διαφορετικές εφαρμογές του διαδικτύου σε μία πιο ολοκληρωμένη μορφή.

Αναφορικά με τα μειονεκτήματα, ως βασικότερο όλων, αναφέρεται το θέμα της ασφάλειας, αν και για αυτό υπάρχει λύση μέσω της σύνδεσης HTTPS ή/και με χρήση μηχανισμών κρυπτογράφησης σε επίπεδο εφαρμογής. Έτσι με το HTTP η διακίνηση πληροφορίας είναι μη ανασφαλής και ευάλωτη σε κακόβουλες ενέργειες. Το HTTPS παρέχει αυξημένη ασφάλεια, αρκεί να χρησιμοποιηθεί ισχυρός αλγόριθμος κρυπτογράφησης είτε σε ολόκληρη την πληροφορία είτε σε τμήμα της και να πραγματοποιείται αυστηρός έλεγχος στα πιστοποιητικά της υποδομής δημόσιου κλειδιού. Άλλα πιθανά ζητήματα ασφάλειας είναι η μη εξουσιοδοτημένη χρήση, η πλαστογράφηση ταυτότητας, η χρήση επιθέσεων τύπου «man in the middle», οι επιθέσεις υπερφόρτωσης/άρνησης παροχής υπηρεσιών κ.λπ. Μέθοδοι PUT και DELETE (και ενδεχομένως POST), που τροποποιούν τα δεδομένα του server ή γενικότερα την κατάσταση της εφαρμογής που φιλοξενείται σε αυτόν και δεν αφορούν απλή ανάκτηση δεδομένων, θέλουν ιδιαίτερη προσοχή ως προς την ασφάλεια.

2.3 REST (REpresentational State Transfer)

Το REpresentational State Transfer (REST) είναι ένα αρχιτεκτονικό στυλ που διέπει τον σχεδιασμό των web services. Χρησιμοποιεί το πρωτόκολλο HTTP/HTTPS για τη μεταφορά των δεδομένων και προσπαθεί να κρατά απλή και εύχρηστη τη δομή του ερωτήματος και της απάντησης από το service. Όσα web services ακολουθούν τους κανόνες του μοτίβου REST καλούνται Restful. Τα βασικά στοιχεία-περιορισμοί του Rest είναι:

1. Διάκριση ρόλων Client-Server: Είναι πλήρως διακριτοί και ευδιάκριτοι οι ρόλοι client και server. Ο client δεν μετέχει στην αποθήκευση δεδομένων, οι διεργασίες που εκτελεί είναι καθαρά διεργασίες διεπαφής με τον χρήστη ή επεξεργασίας, στέλνει αιτήματα προς τον server και λαμβάνει απαντήσεις από αυτόν. Ο server είναι υπεύθυνος για την αποθήκευση των δεδομένων, την εφαρμογή τυχόν επιχειρηματικών κανόνων ή κανόνων ακεραιότητας, τη συνεργασία με άλλους server, αν τυχόν απαιτείται κ.λπ. Ο server δέχεται ερωτήματα από τον client και απαντά σε αυτά.
2. Statelessness: Ο server αντιδρά στα αιτήματα που λαμβάνει, χωρίς να πρέπει να 'θυμάται' προηγούμενα μηνύματα. Όλη η διαχείριση και αποθήκευση της κατάστασης της πληροφορίας για την επικοινωνία μεταξύ client και server γίνεται από την πλευρά του client^[11].
3. Addressability: Κάθε πόρος θα πρέπει να είναι προσβάσιμος μέσα από μία μοναδική διεύθυνση (μοναδικό URI) πχ οι φωτογραφίες θα έχουν δικό τους μοναδικό URI. Όμως με τις σύγχρονες τεχνικές (π.χ. AJAX) το URI είναι ένα και την μοναδικότητα των πόρων την οργανώνουν τα API.
4. Cacheability: Ο προηγούμενος περιορισμός του statelessness δημιουργεί ένα σημαντικό ζήτημα, τον αυξημένο όγκο δεδομένων επικοινωνίας για την κάθε ανανέωση των δεδομένων. Σε αυτό έρχεται να δώσει λύση το cacheability, υπό την έννοια ότι τα δεδομένα μπορούν να μαρκάρονται

ως cacheable ώστε να καταχωρούνται στην cache του client, ο οποίος θα μπορεί να τα επαναχρησιμοποιήσει ως προηγούμενα αποτελέσματα και να μην κάνει διαδικασία νέας ανάκτησης. Το πλεονέκτημα είναι εμφανές, το μειονέκτημα όμως είναι η υποβάθμιση της αξιοπιστίας και η εισαγωγή του ενδεχομένου τα δεδομένα του client να είναι διαφορετικά από αυτά του server.

5. Layered System: Η αρχιτεκτονική του συστήματος δομείται σε επίπεδα, στα οποία αναπτύσσεται το API. Κάθε επίπεδο είναι υπεύθυνο για ένα στάδιο της υπηρεσίας, η οποία μπορεί να εκτελείται στον ίδιο ή διαφορετικό server: για παράδειγμα, στο πλαίσιο της εκτέλεσης ενός αιτήματος ανάκτησης δεδομένων που υποβάλλεται σε έναν server S1, μπορεί ο S1 να διακριβώνει την ταυτότητα του χρήστη καλώντας μία υπηρεσία στον server S2, να ανακτά δεδομένα από μία βάση δεδομένων που εκτελείται στον server S3, καθώς και πρόσθετα δεδομένα από έναν server S4. Έτσι ο client γνωρίζει ποιος διαμεσολαβεί στο αίτημά του (ο S1), αλλά όχι ποιος server εμπλέκεται στην εκτέλεση της υπηρεσίας (δεν γνωρίζει τίποτε για τους S2, S3 και S4).
6. Uniform Interface: Το πρωτόκολλο επικοινωνίας είναι το HTTP/HTTPS, κατά συνέπεια η επικοινωνία client-server θα στηρίζεται στις μεθόδους του. Το βασικότερο όμως είναι, ότι όλοι όσοι χρησιμοποιούν τις μεθόδους αυτές, τις χρησιμοποιούν με τον ίδιο τρόπο ακολουθώντας τις σχετικές συστάσεις, περιμένοντας τα ίδια αποτελέσματα πάντα. Έτσι τυποποιήθηκαν οι εξής «εντολές» κοινές για όλους^[12]:
 - GET: Ανάκτηση δεδομένων από τον server.
 - PUT: Ενημέρωση/Επεξεργασία δεδομένων ή δημιουργία νέων δεδομένων που αποθηκεύονται στον server ως κύριες οντότητες.
 - POST: Αποστολή δεδομένων προς αποθήκευση στον server. Αφορά δεδομένα που σχετίζονται με εξαρτώμενες (subordinate) οντότητες.
 - DELETE: Διαγραφή δεδομένων από τον server.

Όπως είδαμε προηγουμένως οι εντολές δίνονται μέσω HTTP. Το πρωτόκολλο αυτό κατά την επικοινωνία επιστρέφει στον client κωδικό, επιτυχούς ή όχι έκβασης του ερωτήματος. Οι κωδικοί αυτοί έχουν ομαδοποιηθεί ως εξής:

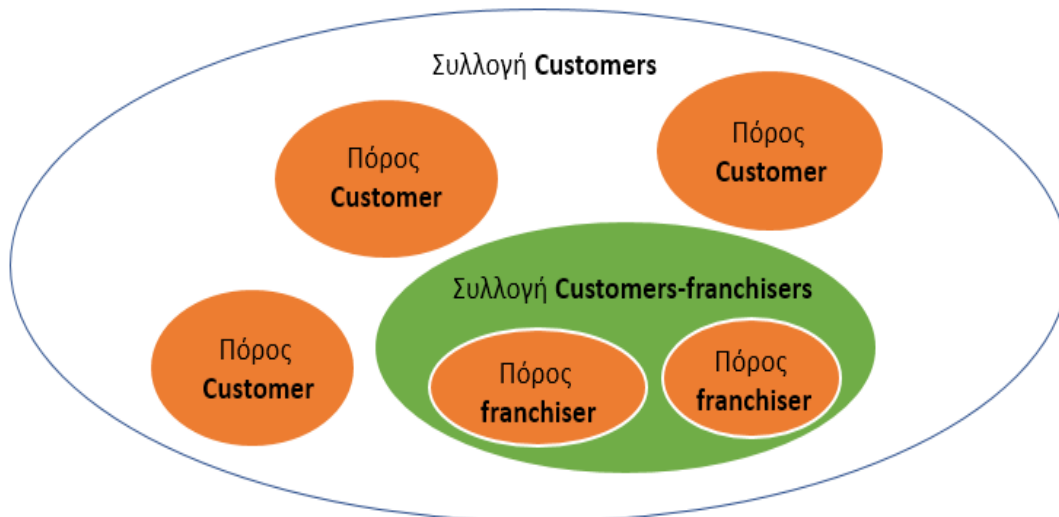
Ομάδα	Περιγραφή
1xx Informational	Ενημερωτικό – Η αίτηση παραλήφθηκε και η διαδικασία εκτελείται
2xx Success	Επιτυχία – Η αίτηση παραλήφθηκε, κατανοήθηκε, έγινε αποδεκτή και εκτελέστηκε με επιτυχία. Ο κωδικός «200: OK» είναι ο πιο γνωστός κωδικός αυτής της κατηγορίας ο οποίος δηλώνει ότι η εκτέλεση του ερωτήματος έγινε με επιτυχία.
3xx Redirection	Ανακατεύθυνση – Απαιτούνται περαιτέρω ενέργειες για την ολοκλήρωση του αιτήματος (υποβολή σε νέα διεύθυνση).
4xx Client Error	Σφάλμα Πελάτη - Η αίτηση περιέχει κακή διατύπωση και δεν μπορεί να τύχει επεξεργασίας ή δεν μπορεί να εξυπηρετηθεί. Τυπικοί λόγοι είναι οι εξής: <ul style="list-style-type: none"> • 400: Bad Request Δηλώνει ότι υπάρχει πρόβλημα στην σύνταξη ή το μέγεθος του αιτήματος.

	<ul style="list-style-type: none"> • 403: Forbidden Ο χρήστης δεν έχει άδεια πρόσβασης στον πόρο. • 404: Not Found Δηλώνει ότι ο πόρος που ζητάμε δεν βρέθηκε
5xx Server Error	Σφάλμα Εξυπηρετητή - Ο εξυπηρετητής απέτυχε να ολοκληρώσει μια έγκυρη αίτηση. Τυπική απάντηση είναι η «500 Internal Server Error», η οποία δηλώνει την ύπαρξη λάθους κατά την εκτέλεση του αιτήματος από τον server η αιτία του οποίου είναι απροσδιόριστη.

Πίνακας 3 Κωδικοί αποκρίσεων πρωτοκόλλου HTTP

2.3.1 Κανόνες ονοματολογίας

Στο υπόδειγμα αρχιτεκτονικής REST, οτιδήποτε θέλουμε να περιγραφεί ως υπηρεσία (συμπεριλαμβάνοντας τα έγγραφα υπερκειμένου), θα αποτελεί έναν πόρο και θα περιγράφεται ως τέτοιος. Έτσι κάθε έγγραφο, εικόνα, χρονική υπηρεσία, ή μία συλλογή άλλων πόρων θα αποτελούν πόρους. Συνεπώς, ένας πόρος είναι μια εννοιολογική χαρτογράφηση σε ένα σύνολο οντοτήτων. Έστω ότι έχουμε την ακόλουθη συλλογή πόρων. Για τη διαμόρφωση της ονοματολογίας ακολουθείται ένα σύνολο κανόνων που αναλυτικά περιγράφεται στο "REST Resource Naming Guide"^[13]. Στη συνέχεια θα δώσουμε τους κανόνες που αφορούν την παρούσα πτυχιακή μέσω παραδειγμάτων, με βάση το σύνολο οντοτήτων που περιγράφεται στην Εικόνα 1.



Εικόνα 2 Σύνολο οντοτήτων

URI	Αντιστοιχία
<i>/customers</i>	<i>προσπέλαση στην συλλογή Customers</i>
<i>/customers/{customerID}</i>	<i>έχουμε προσπέλαση στον στιγμιότυπο της συλλογής Customers με ταυτότητα customerID</i>
<i>/customers/{customerID}/franchisers/{franchiserID}</i>	<i>Από τη συλλογή Customers επιλέγουμε το στιγμιότυπο με ταυτότητα CustomerID και από τους franchisers που ανήκουν σε αυτό το στιγμιότυπο επιλέγουμε αυτόν με ταυτότητα franchiserID</i>

Κατά την ονοματολογία θα πρέπει να τηρούμε κανόνες που συμβαδίζουν με τα παραπάνω. Καλές πρακτικές λοιπόν είναι:

- *Αναφορά στον πόρο και όχι στην ενέργεια/εντολή.*
GET `http://api.example.com/customer/12345`
Σε αντίθεση με τη παραπάνω σωστή γραφή δεν πρέπει να αποκαλύπτουμε την ενέργεια που πρόκειται να συμβεί στο πόρο. Έτσι στη παρακάτω κλήση παρατηρούμε τα εξής:
GET `http://api.example.com/services?op=get_customer&id=12345&format=json`
Προσδιορίζουμε τη μέθοδο GET με την οποία θα ενεργήσουμε πάνω στον πόρο τον οποίο όμως τον αποστέλλουμε ως παράμετρο στο ερώτημά μας. Παράλληλα προσδιορίζουμε ότι η συνάρτηση `get_customer` είναι αυτή που θα μας επιστρέψει τα στοιχεία σε μορφή json. Επιτρεπτό θα ήταν ορίσουμε παραμέτρους οι οποίες θα μας επιτρέψουν να προσεγγίσουμε τον πόρο και όχι να περιγράψουν την ενέργεια σε αυτόν, όπως:
GET `http://api.example.com/customer/12345?region=EU`
- *Περιγραφή συλλογών και πόρων που υπάρχουν σε αυτές.*
URI `http://api.example.com/customers/123/address`
- *Υπαρξη ιεραρχίας για την σαφή περιγραφή του πόρου.*
URI `http://api.example.com/customers/123/address`
- *Χρήση του κόμματος (,) ή της τελείας (.) για περιγραφή λίστας πόρων.*
URI `http://api.example.com/customers/123,456,789`
- *Καθορισμός ερωτημάτων (όπου αυτό είναι απαραίτητο) στην περιγραφή του πόρου.*
URI `http://api.example.com/customers?location=USA`
- *Γενικοί κανόνες συγγραφής ορθού κειμένου στο URI.*
Γραφή με πεζά γράμματα , χρήση (–) αντί για (_), δεν χρησιμοποιούμε το (/) ως τελευταίο χαρακτήρα , χρήση αγγλικών όρων για περιγραφή (airplanes και όχι aeroplana). Τέλος να μην χρησιμοποιούνται επεκτάσεις πχ xml στην γραφή του URL.

2.3.2 Κανόνες ασφαλείας

Το REST API είναι πολύ απλό στην υλοποίησή του και για την υλοποίηση της ασφαλείας του μπορούν να αξιοποιηθούν οι σχετικοί μηχανισμοί που παρέχει το πρωτόκολλο HTTP. Σε όλες τις περιπτώσεις, τα RESTful APIs θα πρέπει να είναι stateless - επομένως το αίτημα ελέγχου ταυτότητας / εξουσιοδότησης δεν πρέπει να εξαρτάται από τα cookies ή τις περιόδους σύνδεσης (sessions), αντιθέτως, κάθε αίτημα API πρέπει να συνοδεύεται από τα απαραίτητα διαπιστευτήρια ελέγχου ταυτότητας, τα οποία θα επικυρώνονται στο διακομιστή για κάθε αίτημα. Τα διαπιστευτήρια ταυτότητας δεν είναι απαραίτητο ότι θα είναι το username και το password του χρήστη ή διαμοιραζόμενο μυστικό, αλλά θα μπορούσαν να είναι ένα token το οποίο έχει αποκτηθεί από προηγούμενη αίτηση και από το οποίο να μπορεί να διακριβωθεί η ταυτότητα του χρήστη (π.χ. τύπος authentication “bearer”). Βασικοί σχεδιαστικοί κανόνες ασφαλείας που πρέπει να τηρούνται περιγράφονται πιο κάτω.

- *Least Privilege:* Η πρόσβαση θα πρέπει να γίνεται με βάση δικαιώματα. Δικαιώματα που έχει ο χρήστης και είναι ακριβώς τα ελάχιστα που απαιτούνται, χωρίς εκχώρηση πρόσθετων δικαιωμάτων.

- *Fail-Safe Defaults*: Δεν θα πρέπει να έχει κανένας χρήστης προεπιλεγμένο επίπεδο πρόσβασης. Το επίπεδο θα αποδίδεται κατά περίπτωση, ανάλογα με την απαιτούμενη πρόσβαση.
- *Economy of Mechanism*: Ο σχεδιασμός πρέπει να είναι απλός για να είναι κατανοητός.
- *Complete Mediation*: Το σύστημα θα πρέπει να επικυρώνει τα δικαιώματα για όλους τους χρήστες, όλους τους πόρους και να διασφαλίζει ότι για κάθε τους αλλαγή θα είναι πάντα ενημερωμένη η αντιστοιχία δικαιωμάτων χρηστών-πόρων (πίνακας δικαιωμάτων).
- *Open Design*: Ο σχεδιασμός θα πρέπει να είναι ανοικτός σε όλους, χωρίς κρυφούς αλγόριθμους.
- *Separation of Privilege*: Η χορήγηση αδειών σε μια οντότητα δεν πρέπει να βασίζεται αποκλειστικά σε μία μόνο συνθήκη, αλλά σε συνδυασμό των συνθηκών βάση τύπου του πόρου.
- *Least Common Mechanism*: Οι κοινοί μηχανισμοί μεταξύ των στοιχείων και πόρων θα πρέπει να είναι ελαχιστοποιημένοι καθώς η παραβίαση του κοινού μηχανισμού σημαίνει αυτομάτως και παραβίαση των επιμέρους στοιχείων που τον αποτελούν.
- *Psychological Acceptability*: Ο μηχανισμός ασφαλείας δεν πρέπει να δυσκολεύει την πρόσβαση και χρήση του χρήστη, αλλά να εκτελείται με το ελάχιστο κόστος σε ευχρηστία.

Με άξονα την ασφάλεια των συστημάτων μπορούμε να κάνουμε κάποιες προτάσεις.

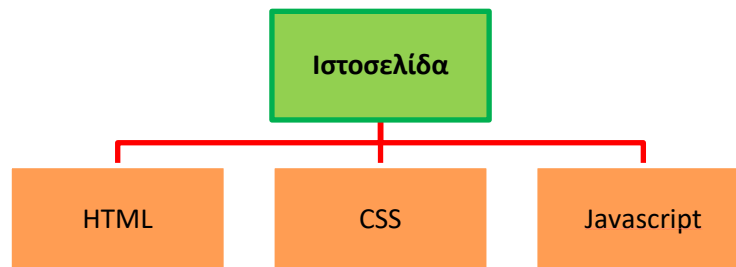
- *Simplicity*: Το σύστημα πρέπει να είναι σχεδιασμένο και υλοποιημένο με τον απλούστερο δυνατό τρόπο, καθώς αχρείαστη πολυπλοκότητα αφ' ενός υποβαθμίζει την ελεγχσιμότητα του συστήματος, αφ' ετέρου μπορεί να οδηγήσει σε σφάλματα τα οποία θα δημιουργήσουν κενά ασφαλείας.
- *Χρήση HTTPS*: Θα πρέπει να χρησιμοποιείται το πρωτόκολλο HTTPS έναντι του HTTP, ώστε να αξιοποιούνται οι πρόσθετες ιδιότητες του πρωτοκόλλου σε σχέση με την ασφάλεια.
- *Hash Password*: Όλοι οι κωδικοί πρέπει να είναι κρυπτογραφημένοι με μονόδρομο τρόπο (hash) κατά την αποθήκευση. Η χρήση κατάλληλων πρωτοκόλλων κατά τη μετάδοση για την αποφυγή χρήσης plaintext passwords, είναι επιθυμητή.
- *URLs Information*: Στο URL δεν πρέπει να συμπεριλαμβάνονται ποτέ οι κάτωθι πληροφορίες: usernames, passwords, session tokens, και API keys. Τα URL μπορεί να υποκλαπούν ή να εξαχθούν από log files κ.λπ. με συνέπεια να διαρρεύσουν οι ευαίσθητες αυτές πληροφορίες.
- *OAuth*: Παρόλο που η απλή εξουσιοδότηση είναι ασφαλής εάν υλοποιηθεί σωστά, η χρήση του πρωτοκόλλου OAuth2.0^[14] μας δίνει τη δυνατότητα ένας τρίτος φορέας (Facebook, Google κ.λπ.) να επικυρώνει την ύπαρξη και αυθεντικοποίηση των στοιχείων ενός χρήστη. Τα στοιχεία αυτά δεν υπάρχουν στο σύστημά μας και έτσι μειώνεται το εύρος των επιθέσεων που μπορεί να πραγματοποιηθούν έναντι του συστήματός μας.
- *Χρονική σήμανση στο request*: Η ιδέα εδώ είναι να προσθέσουμε στο head του HTTP request μία χρονική σήμανση (timestamp). Όταν ο server θα λάβει το αίτημα θα το συγκρίνει με την τρέχουσα χρονική σήμανση. Αν η διαφορά είναι πολύ μικρή τότε το αίτημα είναι έγκυρο αλλιώς απορρίπτεται. Με τον τρόπο αυτό μπορεί να διαγνωστεί η περίπτωση επανειλημμένης αποστολής υποκλαπέντων αιτημάτων (replay attacks), τα αιτήματα αυτά να μην εκτελούνται και να λαμβάνονται τυχόν πρόσθετα μέτρα.
- *Input Parameter Validation*: Το σύστημα πρέπει πάντα να ελέγχει αν τα στοιχεία που έρχονται παρέχονται ως input είναι αποδεκτά και σε μορφή που επιτρέπεται. Τότε και μόνο μπορεί να

προχωρήσει η περαιτέρω επεξεργασία και εκτέλεση του αιτήματος. Για παράδειγμα, στην PHP η εισαγωγή κώδικα αντί για δεδομένα σε πεδίο input μπορεί να μας οδηγήσει σε κενό ασφαλείας με εκτέλεση κώδικα που έχει παρέχει ο επιτιθέμενος. Αντίστοιχα, όταν ο κώδικάς μας διαλειτουργεί με μία βάση δεδομένων μπορεί να υπάρξει επίθεση τύπου SQL injection^[15]. Ο έλεγχος της μορφής των δεδομένων μπορεί να αποτρέψει την επιτυχία επιθέσεων αυτού του τύπου.

3 Δομικά στοιχεία ιστοσελίδας και προγραμματιστικά εργαλεία

Σε αυτό το κεφάλαιο θα περιγράψουμε τα βασικά εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Θα παρουσιαστούν βασικοί κανόνες και δομές των γλωσσών προγραμματισμού που έχουν σαν σκοπό την καλύτερη κατανόηση του κώδικα. Θα αναφερθούμε στην HTML , CSS, σε βασικές εντολές και τεχνικές της Javascript και της PHP. Τέλος θα παρουσιάσουμε την τεχνική ασύγχρονης επικοινωνίας μεταξύ client και server (AJAX) αλλά και το πρότυπο ανταλλαγής δεδομένων JSON που χρησιμοποιείται στο πλαίσιο των κλήσεων API.

Τα βασικά δομικά στοιχεία μιας ιστοσελίδας παρουσιάζονται στην Εικόνα 2 και περιγράφονται αμέσως μετά, ενώ στη συνέχεια θα παρουσιαστούν προγραμματιστικές δομές.



Εικόνα 3 Δομικά στοιχεία ιστοσελίδας

3.1 Hyper Text Markup Language (HTML)

Η Hyper Text Markup Language (HTML) ή αλλιώς Γλώσσα Μορφοποίησης Υπερκειμένου, είναι μία περιγραφική γλώσσα η οποία χρησιμοποιείται ιδίως στο πλαίσιο του παγκόσμιου ιστού για να επισημάνει τον τρόπο δόμησης της εμφάνισης εικόνων, κείμενου και ήχων τα οποία αποστέλλονται από τον server. Η HTML μπορεί να χρησιμοποιείται και στο πλαίσιο τοπικών εφαρμογών. Η HTML έχει απλή δομή και είναι αρκετά ανεκτική σε λάθη. Έτσι:

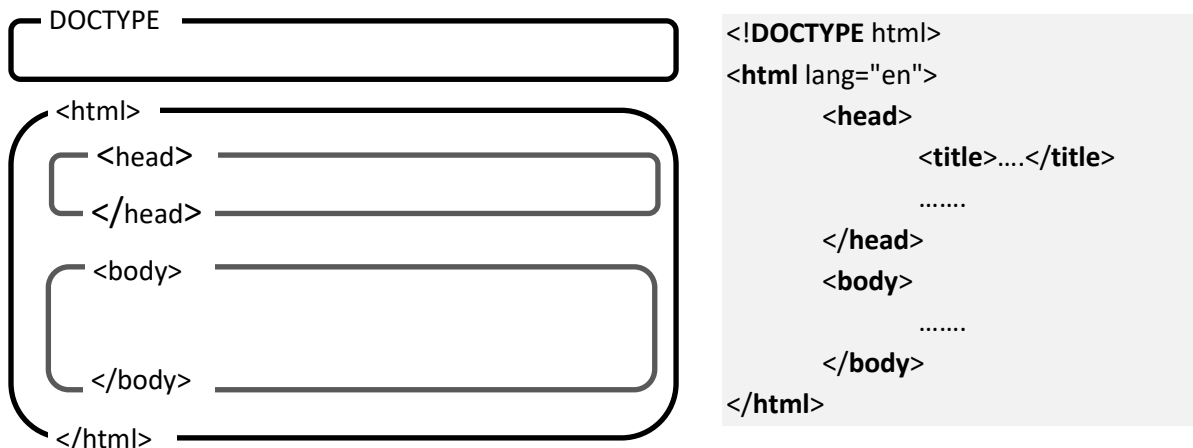
- Η γραφή των ετικετών στην HTML μπορεί να γίνεται με κεφαλαία ή μικρά.
- Η HTML αγνοεί τα κενά.
- Οι εντολές της είναι ετικέτες (tags) που τους δίνουμε ιδιότητες.
- Είναι ανεκτική στα συντακτικά λάθη και προσπαθεί να αγνοεί τα λάθη προς όφελος της λειτουργίας. Για παράδειγμα η απουσία μίας ετικέτας κλεισίματος δεν οδηγεί σε αποτυχία εμφάνισης της σελίδας (αν και μπορεί να οδηγεί σε μη αναμενόμενη εμφάνιση).

Η βασική γραφή των ετικετών είναι

`<ΕΤΙΚΕΤΑ ιδιότητες> ΠΕΡΙΧΟΜΕΝΟ </ΤΕΛΟΣ ΕΤΙΚΕΤΑΣ>` ή `<ΕΤΙΚΕΤΑ ιδιότητες/>`

όπου το ΠΕΡΙΧΟΜΕΝΟ μπορεί να περιέχει με τη σειρά του, τόσο κείμενο όσο και άλλες ετικέτες, οι οποίες θα έχουν πάλι την ίδια σύνταξη.

Μια ιστοσελίδα αποτελείται από το τμήμα πληροφόρησης του περιηγητή για το είδος του εγγράφου (<DOCTYPE>), καθώς και από το τμήμα <html>...</html>, το οποίο με την σειρά του αποτελείται από το τμήμα επικεφαλίδων (<head>...</head>) και το κυρίως σώμα της ιστοσελίδας (<body>...</body>).



Εικόνα 4 Δομή ιστοσελίδας

- Το **DOCTYPE** ενημερώνει τον browser σε ποια έκδοση της HTML είναι γραμμένη η ιστοσελίδα ώστε να εφαρμόζονται οι κατάλληλοι κανόνες του προτύπου. Πολλές φορές όμως οι ίδιοι οι browser δεν ακολουθούν το σύνολο των οδηγιών για τις εκδόσεις αυτές των προτύπων, με αποτέλεσμα να υπάρχουν αποκλίνουσες εμφανίσεις ή συμπεριφορές. Κάνοντας χρήση της HTML5 δεν χρειάζεται να κάνουμε αναφορά σε πρότυπα κειμένου όπως σε παλαιότερες γλώσσες και συνεπώς η σύνταξη θα είναι:

```
<!DOCTYPE html>
```

- Στο **head** υπάρχει συγκεκριμένη ομάδα εντολών-ετικετών η οποία μας διαμορφώνει τον τρόπο συμπεριφοράς του κειμένου μας, ορίζοντας ιδιότητες της ιστοσελίδας και τις συνεργασίες με άλλα αρχεία εντολών. Θα αναφέρουμε επιγραμματικά τις εντολές, αλλά χρήζουν περαιτέρω μελέτης αφού είναι υπεύθυνες για τα keywords της ιστοσελίδας, τη γλώσσα, τη κωδικοποίηση αλλά και τον τύπο του κειμένου (εφαρμογή, εικόνα, ήχος, μήνυμα κλπ). Οι κύριες ετικέτες της περιοχής head είναι οι εξής:

`<title>ΠΕΡΙΕΧΟΜΕΝΟ </title>`: Ορίζει τον τίτλο της ιστοσελίδας, που τυπικά εμφανίζει ο browser στη γραμμή τίτλου.

`<style>ΠΕΡΙΕΧΟΜΕΝΟ </style>`: Ορίζει την μορφή (χρώματα, γραμματοσειρά και άλλες ιδιότητες) που θα έχουν ως πρότυπο καθολικά στην σελίδα άλλες ετικέτες. Η οδηγία `h1{color:green;}` ορίζει ότι το κείμενο που χαρακτηρίζεται ως `h1` (κεφαλίδα επιπέδου 1) θα έχει καθολικά πράσινο χρώμα.

`<link ιδιότητες />`: Ορίζει μία σύνδεση της ιστοσελίδας μας με ένα εξωτερικό URL του ίδιου ή διαφορετικού ιστοτόπου ως τμήμα της ιστοσελίδας μας. Κάνει ενσωμάτωση και όχι απλή αναφορά, συνεπώς κατά τη διερμηνεία του κώδικα της ιστοσελίδας μας, λαμβάνεται υπ' όψιν και η αναφερόμενη σελίδα.

`<script href=".." />` ή `<script ιδιότητες> ...</script>`: Μας συνδέει ένα εξωτερικό αρχείο με κώδικα που συνήθως περιέχει πρόγραμμα γραμμένο σε javascript. Ανάλογα με την μορφή που θα έχει η ετικέτα μπορεί να υπάρχουν εντός του τμήματος εντολές javascript με τη μορφή ελεύθερου κώδικα ο οποίος εκτελείται κατά την φόρτωση της σελίδας ή συναρτήσεις που καλούνται αργότερα από γεγονότα που προσδιορίζονται σε ετικέτες.

`<meta ιδιότητες/>`: Καθορίζονται ιδιότητες character set, description, keywords, author και viewport. Τα στοιχεία αυτά δεν εμφανίζονται αλλά χρησιμοποιούνται από τον ίδιο τον browser, από μηχανές αναζήτησης και άλλους server.

- Το **body** αποτελεί το κυρίως σώμα της ιστοσελίδας. Σε αυτό γράφουμε το περιεχόμενο που θα εμφανιστεί και τις ετικέτες που θα το διαμορφώσουν και θα δώσουν την τελική μορφή του κειμένου στον περιηγητή του χρήστη. Οι ετικέτες αυτές έχουν ιδιότητες που καθορίζουν τον τρόπο προβολής, αλλά και γεγονότα που δημιουργούν την διάδραση με τον χρήστη. Όπως θα παρουσιάσουμε και στη συνέχεια, οι ιδιότητες μπορούν να ρυθμιστούν μέσω της CSS και τα γεγονότα είναι τμήματα κώδικα Javascript που ενεργοποιούνται όταν υπάρξει κάποιο συμβάν, όπως για παράδειγμα το πάτημα του πλήκτρου του ποντικιού. Οι βασικές ετικέτες παρατίθενται πιο κάτω:

Το **body** αποτελεί το κυρίως τμήμα του προγράμματος μας. Σε αυτό γράφουμε τις ετικέτες που θα διαμορφώσουν και θα δώσουν την τελική μορφή του κειμένου στον περιηγητή του χρήστη. Οι ετικέτες αυτές έχουν ιδιότητες που καθορίζουν τον τρόπο προβολής τους αλλά και γεγονότα που δημιουργούν την διάδραση με τον χρήστη. Όπως θα δούμε και παρακάτω οι ιδιότητες μπορούν να ρυθμιστούν μέσω της CSS και τα γεγονότα είναι τμήματα JavaScript κώδικα που ενεργοποιούνται όταν υπάρξει κάποιο συμβάν όπως για παράδειγμα το πάτημα του πλήκτρου του ποντικιού. Οι βασικές ετικέτες παρατίθενται πιο κάτω:

<code>bold</code> <code><u>Underline</u></code> <code><i>Italics</i></code> <code>emphasized</code> <code><sup>κείμενο δείκτης</sup></code>	<code><small>smaller text</small></code> <code><p>paragraph</p></code> <code>
</code> : Αλλαγή γραμμής <code><hr></code> : Εμφάνιση γραμμής <code><h1>headings</h1>(h1 to h6)</code>	<code>Υπερσύνδεσμος</code> <code></code> <code><div>division or a section</div></code> <code>section</code> <code><iframe src='ιστοσελίδα'></iframe></code>
---	---	--

Πίνακας 4 Βασικές ετικέτες της HTML

Λίστες <code><dl></code> <code><dt>όρος</dt></code> <code><dd>περιγραφή</dd></code> <code></dl></code>	<code></code> ή <code></code> <code>κουκίδα</code> <code></code> ή <code></code>	Πίνακες <code><table></code> <code><tr><th>Column1</th> </tr></code> <code><tr><td>Κείμενο</td> </tr></code> <code></table></code>
--	--	---

Πίνακας 5 Λίστες και πίνακες στην HTML

Τέλος έχουμε και τη δυνατότητα δημιουργίας φορμών οι οποίες στο εσωτερικό τους περιέχουν μία σειρά από πεδία εισόδου δεδομένων και χειριστήρια με τα οποία αποστέλλονται τα δεδομένα από τον περιηγητή του χρήστη στον server για περαιτέρω επεξεργασία τους ή και αποθήκευση.

```
<form action='URL αποστολής δεδομένων' method='POST ή GET'>
    ... πεδία εισαγωγής δεδομένων ...
    ... στοιχεία ελέγχου για την αποστολή των δεδομένων
</form>
```

Τα πεδία εισαγωγής δεδομένων που μπορούν να χρησιμοποιηθούν είναι:

- `<input type='Type Of input' />` το οποίο ανάλογα με το type προσαρμόζεται σε button, checkbox, color, date, email, file, hidden, image, password, radio, submit, range, text κλπ
- `<textarea> </textarea>`, που αποτελεί πεδίο για εισαγωγή και επεξεργασία κειμένου μεγάλης έκτασης.
- `<select>` Πεδίο το οποίο χρησιμοποιούμε για να προβάλλουμε λίστα επιλογών, από την οποία μπορούν να επιλεγθούν κανένα, ένα ή περισσότερα στοιχεία.
`<option>Επιλογή</option>`
`</select>`

Σε όλες τις παραπάνω ετικέτες μπορούμε να εφαρμόσουμε ιδιότητες εμφάνισης οι οποίες θα διαμορφώσουν τον τρόπο με τον οποίο οι ετικέτες θα παρουσιαστούν στον περιηγητή μας. Τις ιδιότητες αυτές της ορίζουμε είτε μέσω της `style="ιδιότητες"` είτε μέσω ομάδας ιδιοτήτων (πρότυπα) που ορίζονται με την `class="classname"`. Υπάρχουν και πρόσθετες ιδιότητες που αφορούν λοιπά στοιχεία, όπως π.χ. εξορισμού τιμές, ορισμός πεδίων ως ανάγνωσης μόνο κ.λπ.

Τέλος η HTML, υποστηρίζει διαδραστικότητα με τον χρήστη, την οποία επιτυγχάνει μέσω των γεγονότων (events) που υποστηρίζουν όλες οι ετικέτες. Τα γεγονότα αυτά ενεργοποιούνται όταν υπάρξει κάποιο συμβάν το οποίο ταιριάζει με το συμβάν που προδιαγράφει η ετικέτα. Κατά την ενεργοποίησή τους μπορούν να εκτελεστούν εντολές JavaScript που παρατίθενται άμεσα εντός της ετικέτας ή και να κληθεί συνάρτηση που περιέχει κώδικα σε JavaScript. Ένας συγκεντρωτικός πίνακας βασικών γεγονότων είναι:

Ετικέτα/εμβέλεια	Προσδιορισμός συμβάντος
Body	onafterprint, onbeforeprint, onbeforeunload, onerror, onresize, onload
Συνηθέστερα στα πεδία μίας φόρμας.	onblur, onchange, onfocus, oninput, oninvalid, onreset, onselect, onsubmit, onkeydown, onkeypress, onkeyup
Γεγονότα σε σχέση με την λειτουργία του ποντικιού.	onclick, ondblclick, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onwheel
Στις περισσότερες ετικέτες με έμφαση στην μετακίνηση αντικειμένων	ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onscroll

Πίνακας 6 Βασικότερα γεγονότα (events) στην HTML

3.2 Cascading Style Sheet (CSS)

Το CSS (Cascading Style Sheets) είναι γλώσσα διαμόρφωσης των χαρακτηριστικών εμφάνισης της HTML στον περιηγητή. Το βασικό πλεονέκτημα είναι ότι μία αλλαγή σε αυτά τα χαρακτηριστικά αλλάζει

όλες τις σχετιζόμενες ετικέτες και το αντίστοιχο περιεχόμενο σε όλη ή όλες τις ιστοσελίδες και δεν χρειάζεται να αλλάξουμε κάθε ένα χαρακτηριστικό μεμονωμένα. Επιπλέον διαχωρίζει τους κανόνες εμφάνισης από την περιγραφή της δόμησης της σελίδας. Συνεπώς μας παρέχει ευελιξία σε δύσκολες διαμορφώσεις στυλ, ευκολότερη συντήρηση των ιστοσελίδων, αφού όλες οι ιδιότητες είναι μαζεμένες σε ένα αρχείο, αποφυγή επανάληψης και εύκολη επίτευξη ομοιομορφίας, κώδικα αφού χρησιμοποιούμε το ίδιο στυλ για πολλές ιστοσελίδες.

Οι οδηγίες CSS ενσωματώνονται στη σελίδα με τρεις διαφορετικούς τρόπους:

- Εντός της ετικέτας ως ιδιότητα που παίρνει κάποια τιμή. Η ιδιότητα αυτή και η τιμή της είναι ένα συστατικό των χαρακτηριστικών (attribute) που αποδίδονται εντός της ετικέτας στο `style="..."`. Έχει εφαρμογή μόνο στην συγκεκριμένη ετικέτα.

```
<p style=" margin:10px; color:red; margin:10px; border:1px solid black;">
```

```
....
```

```
</p>
```

- Στην αρχή της ιστοσελίδας εντός του τμήματος `<head>` δημιουργώντας ένα υποτμήμα `<style>...</style>`. Εφαρμόζει τις ιδιότητες μόνο στις ετικέτες αυτής της σελίδας.

```
<head>
```

```
<style>
```

```
.closebtn {
```

```
margin-left: 5px;
```

```
color: white;
```

```
font-weight: bold;
```

```
font-size: 17px;
```

```
cursor: pointer;
```

```
}
```

```
</style>
```

```
...
```

```
</head>
```

- Σε διαφορετικό αρχείο που τυπικά έχει επέκταση `.css` το οποίο μπορεί να βρίσκεται στον ίδιο ή διαφορετικό ιστότοπο. Στο αρχείο αυτό δημιουργούμε ένα πρότυπο στυλ για όλες τις ίδιες ετικέτες ή και μοναδικά για κάποιες ιδιαίτερες ετικέτες. Οι ιδιότητες θα έχουν καθολική εφαρμογή στις ιστοσελίδες, οι οποίες ενσωματώνουν μέσω της ετικέτας `<link >` το συγκεκριμένο αρχείο `.css`.

```
<link href="mainconfig.css" type="text/css" />
```

Το στυλ το οποίο διαμορφώνουμε έχει προτεραιότητα ανάλογα με τον τρόπο που το δημιουργούμε. Έτσι οι ιδιότητες που εφαρμόζονται απευθείας στην ετικέτα έχουν μεγαλύτερη προτεραιότητα έναντι αυτών που είναι τοποθετημένες στην περιοχή `<style>` της ενότητας `<head>` και αυτές με την σειρά τους έχουν μεγαλύτερη προτεραιότητα, έναντι αυτών που βρίσκονται σε εξωτερικό αρχείο. Τέλος αν έχουμε πολλές

φορές δηλωμένη την ίδια ιδιότητα τότε υπερισχύει και εφαρμόζεται η τελευταία κατά σειρά δήλωση έναντι όλων των υπολοίπων.

Εκτός από την πρώτη περίπτωση συγγραφής, όπου οι ιδιότητες εγγράφονται εντός του χαρακτηριστικού της ετικέτας, ο γενικός κανόνας συγγραφής ιδιοτήτων στυλ είναι:

```

Επιλογέας {
    ιδιότητα:τιμή;
    ιδιότητα:τιμή;
    .....
}
    
```

Με τον όρο «επιλογέας» ορίζουμε τον γενικό προσδιορισμό μίας ή περισσότερων ετικετών ή μίας ή περισσότερων κλάσεων. Η ομάδα ιδιοτήτων που ακολουθεί αποδίδεται και επηρεάζει όλο το περιεχόμενο που ταιριάζει με έναν προσδιορισμό ετικέτας ή κλάσης, από αυτές που παρατίθενται στον επιλογέα. Περιπτώσεις γραφής είναι:

Επιλογέας	Περιγραφή
*	Εφαρμόζει την ιδιότητα σε όλες τις ετικέτες της ιστοσελίδας. Παράδειγμα: <i>* {color:blue;} όλες οι ετικέτες να έχουν χρώμα μπλε.</i>
Ετικέτα	Εφαρμόζει την ιδιότητα σε όλες τις εμφανίσεις της ετικέτας. Παράδειγμα: <i>p {color:blue;} όλες οι ετικέτες <p> να έχουν χρώμα μπλε.</i>
.όνομα κλάσης	Εφαρμόζει την ιδιότητα σε όλα τα ίδια στοιχεία που έχουν αυτή την κλάση. Παράδειγμα: <i>.cyan_link {color:cyan;} οι ετικέτες που έχουν class="cyan_link" (π.χ.) να έχουν χρώμα θαλασσί.</i>
ετικέτα.όνομα κλάσης	Εφαρμόζει την ιδιότητα στις ετικέτες «ετικέτα» που έχουν την ίδια κλάση. Παράδειγμα: <i>p.cyan_link {color:cyan;} οι ετικέτες <p> που έχουν class="cyan_link" (π.χ. <p class="cyan_link">) να έχουν χρώμα θαλασσί.</i>
#id_στοιχείου	Εφαρμόζει την ιδιότητα σε εκείνη την ετικέτα που έχει μοναδικό id το οριζόμενο. Παράδειγμα: <i>#item_one {color:cyan;} η ετικέτα που έχει id="item_one" (π.χ. <p id="item_one " >...</p>) θα έχει χρώμα θαλασσί. Κλήση στο <body> γίνεται ως <p id="item_one " >...</p></i>

ετικέτα[ιδιότητα=τιμή]	Εφαρμόζει την ιδιότητα στις ετικέτες «ετικέτα» στις οποίες η ιδιότητα «ιδιότητα» έχει την τιμή που προσδιορίζεται. Παράδειγμα: <code>p.[color="cyan"] {color:white;}</code> οι ετικέτες <code><p></code> με ιδιότητα <code>color="cyan"</code> (π.χ. <code><p style="color:cyan;">...</p></code>) θα αποκτήσουν λευκό χρώμα. Κλήση στο <code><body></code> γίνεται ως <code><p style="color:cyan;">...</p></code> όπου το χρώμα θα γίνει άσπρο.
Επιλογέας1, Επιλογέας2... {ιδιότητα:τιμή; ...}	Όλοι οι επιλογείς θα αποκτήσουν τις ίδιες ρυθμίσεις. Παράδειγμα: <code>input[type="text"], textarea {color:white;}</code> οι ετικέτες <code>input</code> που είναι τύπου <code>text</code> καθώς και οι ετικέτες <code>textarea</code> θα αποκτήσουν άσπρο χρώμα.
Επιλογέας1 επιλογέας2 {ιδιότητα:τιμή; ...}	Σε αυτή την έκφραση ορίζουμε ότι οι αλλαγές θα γίνουν στον επιλογέα 2 που περιέχεται ως περιεχόμενο εντός του επιλογέα 1 Παράδειγμα: <code>p img {border:2px solid red;}</code> το οποίο ορίζει ότι όλες οι ετικέτες <code>img</code> (εικόνες) που περιέχονται μέσα σε οποιαδήποτε παράγραφο θα έχουν συγκεκριμένες ιδιότητες στο περίγραμμά τους. Αυτό δεν ισχύει για τις εικόνες που βρίσκονται εκτός παραγράφων <code><p></code> .

Πίνακας 7 Παραδείγματα CSS

Όταν σε μία ετικέτα εφαρμόζεται στυλ που ορίζεται από επιλογέα με κλάση, από επιλογέα με `id` και με επιλογέα ετικέτας τότε υπάρχει επικάλυψη των στυλ. Αν τα επικαλυπτόμενα στυλ ορίζουν διαφορετικές τιμές για χαρακτηριστικά, επικρατούν οι τιμές του ορισμού με τη μεγαλύτερη προτεραιότητα ως ακολούθως: ο επιλογέας με `id` προηγείται έναντι αυτού με όνομα κλάσης, ο οποίος με την σειρά του προηγείται έναντι αυτού με ετικέτα.

Παράδειγμα:

```
#item_one {border:1px dotted blue;}
.yellowitems{border: double;}
p {border:2px solid red;}
```

Τότε η ετικέτα `<p id="item_one" class="yellowitems">..</p>` θα έχει σαν αποτέλεσμα το περίγραμμα να είναι με την μορφή τελείας μπλε χρώματος και πάχους 1 px καθώς η ιδιότητα `border` του CSS με επιλογέα ταυτότητας (`#item_one`) έχει προτεραιότητα έναντι των άλλων δύο προσδιορισμών CSS.

Κάποιες από τις πιο βασικές ιδιότητες οι οποίες μπορούν να συνδυαστούν και μεταξύ τους ακολουθούν στον παρακάτω πίνακα. Υπάρχει μεγάλο ακόμα πλήθος ιδιοτήτων των οποίων η ονομασία είναι αρκετά περιγραφική σε σχέση με την εργασία που εκτελούν, με αποτέλεσμα να είναι εύκολο να τις καταλάβουμε.

color	background-color	padding
font	background-image	margin
font-size	background-repeat	width
font-family	background-position	height
font-style	background	float
font-weight	border-color	clear: both;
text-decoration	border-width	
text-align	border-style	

Πίνακας 8 Βασικές ιδιότητες CSS

Κατά τη συγγραφή ενός στυλ είναι πολύ χρήσιμο και ευέλικτο να μπορούμε να προσαρμόσουμε την διαμόρφωσή μας με βάση κάποιο στοιχείο των ετικετών (ψευδοστοιχεία) ή κάποιες καταστάσεις (ψευδοκλάσεις) στις οποίες μπορεί να βρεθεί. Οι πιο σημαντικές και ευρέως χρησιμοποιούμενες είναι:

- *a.link* Εφαρμόζει την ιδιότητα στους συνδέσμους (<a>), τους προορισμούς των οποίων ο χρήστης δεν έχει επισκεφθεί.
- *a.visited* Εφαρμόζει την ιδιότητα στους συνδέσμους (<a>), τους προορισμούς των οποίων ο χρήστης έχει επισκεφθεί.
- *στοιχείο.active* Εφαρμογή στα «στοιχεία» που ο χρήστης έχει πατημένο το ποντίκι πάνω σε αυτά.
- *στοιχείο.hover* Εφαρμογή στα «στοιχεία» πάνω από τα οποία βρίσκεται ο δρομέας του ποντικού.
- *στοιχείο.focus* Εφαρμογή στα «στοιχεία» που έχουν την εστίαση.
- *στοιχείο:first-letter* Χρησιμοποιείται για την δημιουργία αρχιγράμματος και έχει εφαρμογή στο πρώτο γράμμα του «στοιχείου».
- *στοιχείο:first-line* Το ίδιο με το *στοιχείο:first-letter*, αλλά έχει εφαρμογή στην πρώτη γραμμή του «στοιχείου».

3.3 Bootstrap

Όπως είδαμε προηγουμένως, η CSS είναι μία γλώσσα διαμόρφωσης των χαρακτηριστικών με τα οποία προβάλλονται οι ετικέτες της HTML στον περιηγητή. Η HTML μπορεί να παράγει την προβολή της ιστοσελίδας στον περιηγητή αλλά δεν μπορεί να το διαμορφώσει με φιλικό τρόπο. Μία εργαλειοθήκη που θα περιελάμβανε τμήματα της HTML αλλά και την απαραίτητη CSS, θα μπορούσε να δώσει λύση στο πρόβλημα της σωστής και φιλικής προς τον χρήστη διαμόρφωσης μίας ιστοσελίδας. Ακριβώς αυτόν τον σκοπό υπηρετεί το Bootstrap, το οποίο λειτουργεί ως τυπογραφική εργαλειοθήκη παρέχοντάς μας, τυπογραφικές μορφές των ετικετών HTML (κουμπιά, φόρμες, μορφές κ.λπ.). Όταν η εργαλειοθήκη αυτή συνδυαστεί με κάποιες βιβλιοθήκες JavaScript (π.χ. JQuery και άλλες) τότε πετυχαίνουμε, η ιστοσελίδα να είναι και φιλική προς τον χρήστη αλλά και διαδραστική με αυτόν. Οι εργασίες που εκτελούνται με τον παραπάνω τρόπο λαμβάνουν χώρα στον περιηγητή του χρήστη και μόνο σε αυτόν, συνεπώς το βάρος της εκτέλεσής του το έχει ο χρήστης και όχι ο server που φιλοξενεί την ιστοσελίδα.

Το Bootstrap δεν υλοποιεί το σύνολο των εντολών της HTML και CSS, επιδιώκοντας με αυτόν τον τρόπο αυξημένη συμβατότητα με όλους τους περιηγητές. Το υποσύνολο των εντολών HTML και CSS που δεν περιλαμβάνονται, είναι αυτό που προκαλεί ασυμβατότητα ιστοσελίδων μεταξύ των περιηγητών και έχει ως αποτέλεσμα ιστοσελίδες που εμφανίζονταν με συγκεκριμένο τρόπο, σε κάποιο περιηγητή, να μην μπορούν να εμφανιστούν σωστά σε άλλους. Συνεπώς αποτελεί ένα προδιαμορφωμένο στυλ

εμφάνισης και λειτουργίας στο οποίο ο προγραμματιστής μπορεί να επέμβει σε περιορισμένη έκταση επικαλύπτοντας ιδιότητες όπως είδαμε και στην λειτουργία της CSS. Η τεχνική της επικάλυψης των ιδιοτήτων έχει ως αποτέλεσμα την επαναχρησιμοποίηση των τυπογραφικών στοιχείων που μας παρέχει το bootstrap και σε συνδυασμό με την εύκολη προσαρμογή των στοιχείων αυτών, σε κάθε είδους ανάλυσης οθόνης χρήστη, το μετέτρεψαν σε ένα εξαιρετικό εργαλείο ανάπτυξης διεπαφής χρήστη.

Η εργαλειοθήκη αυτή έχει χρησιμοποιηθεί στην ανάπτυξη της πλατφόρμας μας για την δημιουργία τόσο της κεντρικής ιστοσελίδας που θα είναι διαθέσιμη στους επισκέπτες-χρήστες όσο και στο τμήμα της διαχείρισης, που είναι διαθέσιμο στους κατάλληλα πιστοποιημένους χρήστες.

3.4 JavaScript

Η JavaScript είναι μία γλώσσα σεναρίων (scripting language) με την οποία μπορούμε να κατασκευάσουμε σεναρία που θα εκτελούνται στον περιηγητή του χρήστη. Αποτελεί μία γλώσσα στην οποία μπορούμε να στηρίξουμε την διάδραση και τη φιλικότητα της ιστοσελίδας μας προς τον χρήστη. Έχει τη δυνατότητα να αλληλεπιδρά με τον server στέλνοντας σε αυτόν ερωτήματα, τις απαντήσεις των οποίων παρουσιάζει στον browser του χρήστη μέσω της HTML και CSS. Δεν έχει καμία πρόσβαση στα αρχεία του server και η εκτέλεσή της περιορίζεται αποκλειστικά εντός του browser του χρήστη. Ακριβώς αυτή η ιδιότητα, της εκτέλεσής της στον Browser του χρήστη, δημιουργεί κενά ασφαλείας για κακόβουλα λογισμικά που επιδιώκουν να έχουν πρόσβαση στον υπολογιστή του χρήστη μέσω του περιηγητή ή στα αποθηκευμένα στοιχεία όπως αριθμούς καρτών, κωδικούς κλπ.

Ο κώδικας JavaScript μπορεί να βρίσκεται στην ενότητα <head> της ιστοσελίδας, σε οποιοδήποτε σημείο εντός του <body>, στους προσδιορισμούς γεγονότων των ετικετών ή σε αρχεία τύπου .js, στα οποία τοποθετούμε κώδικα και συναρτήσεις που επιτελούν συγκεκριμένες εργασίες. Όταν ο κώδικας JavaScript είναι γραμμένος ως «ελεύθερες εντολές» εκτός συναρτήσεων, αλλά πάντα εντός των ετικετών <script>...</script>, όπως θα δούμε, εκτελείται κατά την φόρτωση της ιστοσελίδας. Συγκεκριμένα όταν τοποθετείται στην αρχή, τότε πρώτα εκτελείται ο κώδικας και έπειτα φορτώνεται στον περιηγητή η υπόλοιπη ιστοσελίδα, αντίθετα όταν τοποθετείται στο τέλος πρώτα φορτώνεται η ιστοσελίδα και έπειτα εκτελείται ο κώδικας JavaScript. Η γραφή της JavaScript γίνεται με τους εξής τρόπους:

- Στο <head> ή στο <body> τμήμα μίας ιστοσελίδας όπου μπορούμε είτε να γράψουμε τον κώδικα εντολών είτε να ενσωματώσουμε κώδικα που υπάρχει σε ένα εξωτερικό αρχείο. Αυτό το πετυχαίνουμε με χρήση της ετικέτας <script>.

```
<script src="mylib.js" type="text/javascript">
```

Τη μορφή αυτή τη χρησιμοποιούμε όταν επιθυμούμε να ενσωματώσουμε κώδικα που βρίσκεται σε εξωτερικό αρχείο (πχ mylib.js), δηλώνοντας ότι το αρχείο είναι τύπου εντολών JavaScript.

```
<script language = "javascript" type = "text/javascript">
```

```
....
```

```
</script>
```

Σε αυτή τη δήλωση ενημερώνουμε ότι ανάμεσα στην αρχή και το τέλος της ετικέτας <script> υπάρχει κείμενο κώδικα με εντολές JavaScript. Όταν στο κείμενο υπάρχουν συναρτήσεις, τότε

αυτές μπορούν να κληθούν από οποιοδήποτε μέρος του κώδικα ή κατά την εκτέλεση συμβάντων των ετικετών.

- Κάθε ετικέτα έχει μία σειρά από συμβάντα (Global Event Attributes) τα οποία ενεργοποιούνται υπό συνθήκες. Όταν ενεργοποιούνται μπορεί να εκτελεστεί κώδικας σε JavaScript είτε απευθείας είτε καλώντας κάποια συνάρτηση.

```
<a href="#" onclick="javascript:OpenNewLink();">
```

```
<button onclick="javascript>window.location.href='a.php'">Go</button>
```

Το σύνολο των γεγονότων που μπορεί να υπάρχουν δεν ορίζεται στα πλαίσια της JavaScript αλλά αποτελούν τμήμα του πρότυπου αντικειμένου εγγράφου (DOM-Document Object Model^[16]). Δυστυχώς το πρότυπο αυτό δεν είναι ίδιο σε όλους τους περιηγητές, με αποτέλεσμα να υπάρχουν ιστοσελίδες που κάνουν χρήση συμβάντων τα οποία δεν εκτελούνται στον περιηγητή του χρήστη λόγω ασυμβατότητάς τους με αυτόν.

Η γραφή της JavaScript έχει πολλές ομοιότητες με τη C/C++ και υποστηρίζει τη δημιουργία αντικειμένων χωρίς όμως να είναι μία αντικειμενοστρεφής γλώσσα όπως άλλες (πχ Java), αφού δεν ενσωματώνει την έννοια της κληρονομικότητας και οι έννοιες των κλάσεων είναι πολύ απλοποιημένες. Τα βασικά στοιχεία είναι:

Κανόνες	<p>Οι μεταβλητές έχουν τα παρακάτω χαρακτηριστικά.</p> <ul style="list-style-type: none">• Ξεκινούν από a-z A-Z ή ‘_’• Υπάρχει διαφορά πεζών-κεφαλαίων (case sensitive).• Δεν επιτρέπονται κενά και δεσμευμένες λέξεις. <p>Ανάθεση τιμής μπορεί να γίνει με τον τελεστή = σε οποιοδήποτε σημείο.</p> <pre>var ονομα = τιμή;</pre>
Μεταβλητές	<p>Η javascript δεν απαιτεί να κάνουμε δήλωση του τύπου μίας μεταβλητής. Φυσικά η μη δήλωση μίας μεταβλητής, εγκυμονεί κινδύνους και συνίσταται κάθε μεταβλητή να δηλώνεται μέσω της var, π.χ.:</p> <pre>var ονομα1, ονομα2;</pre> <p>Ο τύπος της μεταβλητής αποδίδεται αυτόματα από την JavaScript κατά την πρώτη ανάθεση τιμής σε αυτή. Όταν μία μεταβλητή δηλώνεται εντός μίας συνάρτησης, τότε αυτή είναι μία τοπική μεταβλητή (local variable) με εμβέλεια μόνο την έκταση της συνάρτησης. Αντίθετα αν δηλωθεί εκτός συνάρτησης τότε θεωρείται καθολική μεταβλητή (global variable) και είναι προσβάσιμη από όλο το πρόγραμμα. Οι βασικοί τύποι είναι:</p> <ul style="list-style-type: none">• Αριθμοί: Ακέραιοι, πραγματικοί• Χαρακτήρες ή αλφαριθμητικοί χαρακτήρες• Λογικές (Boolean): True, false <p>Τα αντικείμενα τα οποία μπορούμε να χρησιμοποιήσουμε κατηγοριοποιούνται ως εξής:</p> <ul style="list-style-type: none">• αντικείμενα της ίδιας της γλώσσας (π.χ. Math, Date, String, Array)

	<ul style="list-style-type: none"> • αντικείμενα του περιηγητή, Browser objects (π.χ. Window, Navigator, History) • αντικείμενα του HTML DOM (Document, Elements, Attributes, Events) • αντικείμενα της HTML (form, canvas, image, table) • αντικείμενα οριζόμενα από τον χρήστη
Τελεστές	<p>Οι τελεστές μπορούν να κατηγοριοποιηθούν σε λογικούς, αριθμητικούς, δυαδικών ψηφίων, ανάθεσης, σύγκρισης.</p> <p>Σύγκρισης ==, !=, <=, >=, <, ></p> <p>Δυαδικών ψηφίων &, , ^, <<, >>, >>></p> <p>Λογικοί &&, , !</p> <p>Αριθμητικοί +, -, *, /, %, ++, --</p> <p>Αντικατάστασης +=, -=, *=, /=, %=, <<=, >>=, >>>=, &=, ^=, =</p>
Εντολές ελέγχου	<ul style="list-style-type: none"> • <i>switch</i> (έκφραση) { <i>case</i> ετικέτα1: εντολές1; <i>break</i>; <i>case</i> ετικέτα2: εντολές2; <i>break</i>; <i>default</i>:εντολές; } • <i>if</i> (συνθήκη) {εντολές1;} <i>else</i> {εντολές2;}
Εντολές επανάληψης	<ul style="list-style-type: none"> • <i>for</i> (αρχική τιμή; συνθήκη; βήμα) {εντολές;} <i>for</i> (μεταβλητή <i>in</i> αντικείμενο) {εντολές;} • <i>while</i> (συνθήκη) {εντολές} <p><i>do</i> {εντολές;} <i>while</i> (συνθήκη)</p> <p>Εντός των παραπάνω δομών επανάληψης μπορούμε να χρησιμοποιήσουμε εντολές ελέγχου επανάληψης ως εξής:</p> <ul style="list-style-type: none"> • <i>break</i>; Τερματίζει την εκτέλεση του βρόγχου.

	<ul style="list-style-type: none"> • <i>continue</i>; Τερματίζει την συγκεκριμένη επανάληψη και συνεχίζει στην επόμενη, σύμφωνα με την συνθήκη
Συναρτήσεις	<p>Η JavaScript έχει ιδιαίτερο χειρισμό στις παραμέτρους που μπορεί να δεχτεί κατά την κλήση μίας συνάρτησης. Ανεξάρτητα από τις δηλωμένες παραμέτρους, μπορούμε να καλέσουμε μία συνάρτηση με λιγότερες ή περισσότερες παραμέτρους. Όταν παρατίθενται λιγότεροι παράμετροι το σύστημα αποδίδει μηδενική τιμή στις ελλείπουσες, ενώ όταν παρατίθενται περισσότερες τότε το σύστημα δημιουργεί ένα πίνακα με όνομα <i>arguments</i> ο οποίος περιέχει όλες τις τιμές των παραμέτρων.</p> <pre>function όνομα_συνάρτησης (παράμετρος1, παράμετρος2) { εντολές; return τιμή; }</pre>

Πίνακας 9 Βασικά στοιχεία της Javascript

3.5 PHP

Η PHP (Hypertext Preprocessor) είναι μια scripting γλώσσα προγραμματισμού, ανοιχτού κώδικα για τη δημιουργία σελίδων και εφαρμογών ιστού. Είναι scripting καθώς είναι διερμηνευόμενη. Αναμειγνύεται πλήρως με την HTML, υπό την έννοια ότι τμήματα κώδικα PHP μπορούμε, να τα ενσωματώσουμε σε μία ιστοσελίδα HTML και κατά την επεξεργασία της σελίδας στον server να εκτελεστεί ο κώδικας PHP και να αντικατασταθεί ο κώδικας με το αποτέλεσμα της εκτέλεσής του. Έχει γνωρίσει μεγάλη απήχηση στους κατασκευαστές προγραμμάτων για web, με αποτέλεσμα να έχουν δημιουργηθεί μία πλειάδα από βιβλιοθήκες που μπορούν να ενσωματωθούν σε ένα πρόγραμμα και συνεργάζονται μεταξύ τους με ιδιαίτερη ευκολία. Μία τέτοια βιβλιοθήκη υποστηρίζει τη σύνδεση της PHP με την MySQL, η οποία αποτελεί μία πολύ διαδεδομένη σχεσιακή βάση δεδομένων. Μία ακόμα χρήσιμη βιβλιοθήκη είναι η cURL η οποία δίνει τη δυνατότητα στην PHP να μπορεί να επικοινωνήσει και να συνδεθεί με διαφορετικού τύπου διακομιστές, κάνοντας χρήση διαφορετικών τύπων πρωτοκόλλων. Οι δύο παραπάνω βιβλιοθήκες θα μας βοηθήσουν στην εφαρμογή μας, τόσο στο να χειριστούμε τα δεδομένα που πρέπει να αποθηκεύονται στον server, όσο και την επικοινωνία της εφαρμογής μας με άλλους διακομιστές με τους οποίους θα επικοινωνεί μέσω API.

Από το σύνολο των χαρακτηριστικών και των δυνατοτήτων της PHP, εδώ θα αναφερθούμε μόνο στα σημαντικότερα χαρακτηριστικά γραφής προγράμματος. Όπως και η JavaScript έχει δομή η οποία μοιάζει πολύ με τη C και ταυτόχρονα έχει τη δυνατότητα δημιουργίας αντικειμένων όπως και η C++. Συνεπώς ο τρόπος γραφής βρόχων επανάληψης, εντολών ελέγχου αλλά και άλλων βασικών εντολών θα μοιάζουν πολύ με αυτές της C ή της JavaScript. Τα βασικά στοιχεία της PHP θα μπορούσαν να ομαδοποιηθούν στον παρακάτω πίνακα:

Γραφή κώδικα	<p>Σε οποιοδήποτε αρχείο .html ή σε αρχεία .php η γραφή έχει την μορφή</p> <pre><?php // κώδικας PHP ?></pre>
--------------	---

<p>Μεταβλητές</p>	<p>Η PHP είναι dynamically and weakly (loosely) typed γλώσσα, δηλαδή οι μεταβλητές της ορίζονται δυναμικά με βάση την τιμή τους και τυχόν σφάλματα αναζητούνται κατά τη διάρκεια της εκτέλεσης.</p> <p>Η εμβέλεια κάθε μεταβλητής καθορίζεται από τον τρόπο δήλωσής της. Έτσι:</p> <ul style="list-style-type: none"> • <i>Local</i>: Έχουν εμβέλεια μόνο μέσα στη συνάρτηση ή στο αρχείο κώδικα που εμφανίζονται • <i>Global</i>: Έχουν εμβέλεια μόνο μέσα στη συνάρτηση ή στο αρχείο κώδικα που εμφανίζονται • <i>Static</i>: Όμοια με την local εντός μιας συνάρτησης, αλλά διατηρεί την τιμή της μέχρι την επόμενη εκτέλεση της συνάρτησης.
<p>Πίνακες</p>	<p>Οι πίνακες στην PHP μπορεί να είναι μονοδιάστατοι ή και πολυδιάστατοι και ο χειρισμός τους είναι πολύ εύκολος. Ορίζονται με την εντολή Array() με εύκολο τρόπο, είτε ως πίνακες δεικτοδοτούμενοι με θέση, είτε με τον παρακάτω τρόπο ο οποίος περιλαμβάνει και το κλειδί δεικτοδότησης και την τιμή (συσχετιστικοί πίνακες).</p> <p style="text-align: center;"><i>array(key => value,...)</i></p> <p>Υπάρχει μεγάλη ποικιλία ενσωματωμένων συναρτήσεων της PHP για τις εργασίες που χρειαζόμαστε να κάνουμε με πίνακες. Η διαχείρισή τους και η εργασία με αυτούς είναι ένα από τα ισχυρά σημεία της γλώσσας.</p>
<p>Super Global μεταβλητές</p>	<p>Οι Super Global μεταβλητές είναι πίνακες, η τιμή των οποίων είναι προσπελάσιμη από οποιοδήποτε σημείο του προγράμματος. Σχετίζονται με πληροφορίες και χαρακτηριστικά του server, της PHP αλλά και της τρέχουσας αίτησης και η τιμή τους μπορεί να είναι για ανάγνωση μόνο ή ανάγνωση και εγγραφή, κατά περίπτωση. Οι πίνακες αυτοί έχουν την μορφή <i>Πίνακας[κλειδί]=τιμή</i>. Οι βασικότεροι είναι:</p> <ul style="list-style-type: none"> • <i>\$_SERVER[]</i>: πληροφορίες για τον server. • <i>\$_GET[]</i>, <i>\$_POST[]</i>: πίνακες με πληροφορία όλων των στοιχείων παραμέτρων που μεταβιβάζονται στην αίτηση. Οι παράμετροι δίδονται είτε μέσω φόρμας, είτε μέσω προγραμματιστικής κλήσης. • <i>\$_REQUEST[]</i>: πίνακας με στοιχεία που μεταβιβάζονται με την αίτηση, είτε ως παράμετροι GET, είτε ως παράμετροι POST, είτε ως cookies. • <i>\$_FILES[]</i>: πίνακας με τα στοιχεία των αρχείων που δόθηκαν για upload. • <i>\$_SESSION[]</i>: πίνακας με στοιχεία που σχετίζονται με καταχωρήσεις της περιόδου σύνδεσης (session). • <i>\$_ENV[]</i>: πίνακας με στοιχεία που σχετίζονται με μεταβλητές του περιβάλλοντος.
<p>Εντολές ελέγχου Εντολές επανάληψης</p>	<p>Ακολουθούν την ίδια δομή και γραφή όπως αυτές της JavaScript, καθώς και οι δύο αυτές γλώσσες ακολουθούν ως πρότυπο γραφής αυτό της C.</p>

	<p>Διαφοροποίηση παρουσιάζει η <code>foreach</code> η οποία εκτελεί τον κώδικα κάθε φορά και για διαφορετικό στοιχείο του πίνακα διατρέχοντάς τον σειριακά.</p> <pre>foreach (\$array as \$value) { κώδικας... }</pre>
<p>Συναρτήσεις</p>	<p>Ο χειρισμός και η γραφή των συναρτήσεων από την PHP είναι παρόμοιος με αυτόν της JavaScript. Μία παράμετρος μπορεί να αρχικοποιηθεί κατά τον ορισμό της συνάρτησης και να λάβει αυτή την προκαθορισμένη τιμή στην περίπτωση που δεν δοθεί άλλη τιμή κατά την κλήση της συνάρτησης. Στην περίπτωση που η συνάρτηση κληθεί με παραμέτρους χωρίς αυτές να έχουν οριστεί, τότε ενεργοποιείται ένας πίνακας παραμέτρων εντός της συνάρτησης, η προσπέλαση του οποίου γίνεται μέσω των συναρτήσεων <code>func_get_arg()</code>, <code>func_get_args()</code>, <code>func_num_args()</code>.</p> <p>Κατά την κλήση μίας συνάρτησης θα πρέπει να τηρείται η σειρά ορισμού των παραμέτρων. Μια δόκιμη τεχνική είναι να καλούμε την συνάρτηση με όρισμα ένα πίνακα παραμέτρων, με κλειδί το όνομα της παραμέτρου και τιμή την τιμή της παραμέτρου. Στην συνέχεια αυτός ο πίνακας θα μπορεί να γίνει συνένωση (συνάρτηση <code>array_merge()</code>) με ένα πίνακα αρχικών τιμών και όποιο κλειδί δεν έχει ορισθεί θα αρχικοποιείται εντός της συνάρτησης. Ως παράδειγμα έχουμε την κλήση της συνάρτησης <code>CreteICon(array('parname2'=>'100px'))</code>; η οποία έχει ορισθεί ως:</p> <pre>function CreteICon(\$agivenconfig=Array()) { \$aconfig=array('parname1'=>'DefVal1', 'parname2'=>'DefVal2' ); \$aconfig=array_merge(\$aconfig, \$agivenconfig); return null; }</pre>

Πίνακας 10 Βασικά στοιχεία της PHP

Η PHP μας δίνει τη δυνατότητα να συμπεριλαμβάνουμε κώδικα που υπάρχει σε ένα ή πολλά αρχεία μέσα στο τρέχον αρχείο προγράμματος. Με αυτό τον τρόπο κάνουμε επαναχρησιμοποίηση κώδικα (Code Reusability) και μπορούμε να οργανώσουμε τα αρχεία με βάση τις γενικές εργασίες που επιτελούν. Έτσι δημιουργούμε βιβλιοθήκες με συναρτήσεις που εκτελούν εργασίες γραμμένες με γενικό τρόπο και τις καλούμε από πολλά σημεία. Η συμπερίληψη του κώδικα από βιβλιοθήκες που έχουμε φτιάξει μπορούν να γίνουν με τις συναρτήσεις `include`, `require` και `require_once`.

- Η `include` συναντάται στην αρχή συνήθως του κυρίως αρχείου και αντικαθιστά τη γραμμή που βρίσκεται με τον αντίστοιχο κώδικα του προς συμπερίληψη αρχείου σαν να είχε ενσωματωθεί με συνένωση κατά τη γραφή. Αυτό θα συμβεί μόνο όταν ο έλεγχος εκτέλεσης φτάσει σε αυτό το σημείο. Αν η συνάρτηση `include` καλείται μέσα σε ένα `if` στο οποίο δεν πληρείται η συνθήκη, τότε η συμπερίληψη δεν θα συμβεί. Μπορούμε να κάνουμε συμπερίληψη απομακρυσμένου αρχείου,

πρακτική όμως που ενέχει πολλά θέματα ασφάλειας και θα πρέπει να υπάρχει ειδικός σχεδιασμός για αυτό.

- Η *require* έχει πολλές ομοιότητες με την *include* αλλά η βασική διαφορά είναι ότι κατά την έναρξη εκτέλεσης του κυρίως προγράμματος η *require* αντικαθίσταται από το αντίστοιχο κομμάτι κώδικα, ακόμα και αν δεν χρειαστεί ποτέ να εκτελεσθεί. Δεν μπορεί να χρησιμοποιηθεί για συμπερίληψη υπό συνθήκη, αφού πάντα θα συμπεριλαμβάνεται. Όποια μεταβλητή έχει ορισθεί στην εμβέλεια στην οποία λαμβάνει χώρα η συμπερίληψη θα είναι διαθέσιμη αυτόματα μέσα στο συμπεριλαμβανόμενο αρχείο, εφόσον έχει γίνει ουσιαστικά μέρος του καλούντος αρχείου.
- Η *require_once* λειτουργεί ακριβώς το ίδιο με την *require*, αλλά ελέγχει ώστε η συμπερίληψη να γίνει μόνο μία φορά από το αρχικό αρχείο.

Κλείνοντας αυτή τη σύντομη περιήγησή μας στην γλώσσα προγραμματισμού PHP θα πρέπει να κάνουμε αναφορά και στη δημιουργία κλάσεων, καθώς η PHP είναι μια αντικειμενοστρεφής γλώσσα προγραμματισμού. Η αναφορά μας θα σταθεί σε απλές μορφές δημιουργίας αντικειμένων και δεν θα επεκταθεί σε όλες τις δυνατότητες της γλώσσας ώστε να μπορούμε να διαβάσουμε τον κώδικα της εφαρμογής με ευκολία.

- **Class (κλάση):** Πρόκειται για έναν τύπο δεδομένων που καθορίζεται από τον προγραμματιστή, ο οποίος περιλαμβάνει τοπικές λειτουργίες καθώς και τοπικά δεδομένα. Μία κλάση αποτελεί το πρότυπο (template) για την παραγωγή πολλών αντιτύπων του αρχικού τύπου.
- **Object (Αντικείμενο):** Ένα αντίτυπο μίας κλάσης και έχει τη δομή που προσδιορίζεται από αυτή. Μετά τον ορισμό της κλάσης, μπορούν να δημιουργηθούν πολλά στιγμιότυπα της ως objects.
- **Member Variable (μεταβλητή):** Μεταβλητές που ορίζονται μέσα σε μια κλάση και είναι αόρατες στο εξωτερικό της, και καθορίζουν την κατάσταση του αντικειμένου. Οι member variables είναι προσβάσιμες μόνο από τις Member functions. Αυτές οι μεταβλητές ονομάζονται και χαρακτηριστικά.
- **Member function:** Συναρτήσεις που ορίζονται μέσα σε μια κλάση και χρησιμοποιούνται για την υλοποίηση συμπεριφοράς του αντικειμένου, συμπεριλαμβανομένης της πρόσβασης σε δεδομένα και μεταβλητές των αντικειμένων.

Οι παραπάνω ορισμοί μπορούν να συνοψισθούν στα παρακάτω παραδείγματα:

```

class Student{
    private $stuName = "Dimitrios";

    public function getName() {
        return $this-> stuName;
    }
}

```



```

$studentObj= new Student();
echo "Student name!:". $studentObj-> getName();

```

```

class Student{
    private $personal;

    function __construct() {
        $this->person=array("Alex","24","gr");
    }

    function getAllPerson() {
        foreach($this->person as $p) {
            print $p;
        }
    }
}

```



```

$stuObj= new Student();
$stuObj -> getAllPerson();

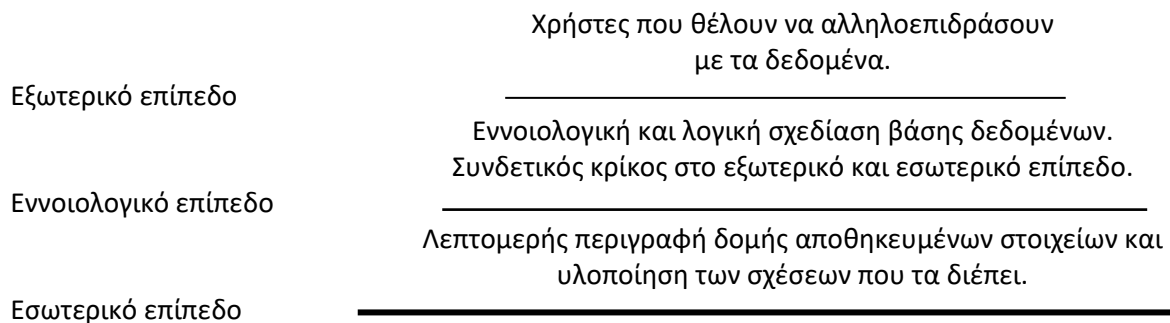
```

Εικόνα 5 Παραδείγματα εννοιών για την PHP

3.6 Βάσεις δεδομένων

Στο πεδίο της πληροφορικής, τα δεδομένα είναι όλα τα στοιχεία τα οποία μπορούν να λάβουν ψηφιακή μορφή και να καταχωρηθούν σε ένα υπολογιστή. Η καταγραφή και η αποθήκευση των δεδομένων σε ένα πληροφοριακό σύστημα γίνεται σε αρχεία με ένα οργανωμένο τρόπο εγγραφών ώστε να είναι δυνατή η εύκολη επεξεργασία και η συσχέτισή τους. Η οργανωμένη αποθήκευση, διαχείριση και προσπέλαση των δεδομένων γίνεται μέσα από ειδικά λογισμικά, τις βάσεις δεδομένων.

Ένα σύστημα διαχείρισης βάσεων δεδομένων έχει σαν βασικό στόχο την επικοινωνία του εξωτερικού χρήστη με τη φυσική αποθήκη των δεδομένων, αποκρύπτοντας τις τεχνικές λεπτομέρειες της αποθήκευσης, της επεξεργασίας των αιτημάτων και παρέχοντας ένα εύχρηστο και αποτελεσματικό πλαίσιο αποθήκευσης, ανάκτησης και διαχείρισης των δεδομένων. Σε μία βάση δεδομένων υπάρχουν τα εξής επίπεδα:



Εικόνα 6 Επίπεδα βάσης δεδομένων

Στο σύστημα αυτό εφαρμόζεται η αρχή της ανεξαρτησίας των δεδομένων, κατά την οποία κάθε αλλαγή που συμβαίνει σε ένα επίπεδο δεν θα πρέπει να επηρεάζει τα υψηλότερα επίπεδα από αυτό στο οποίο συνέβη η αλλαγή. Συνεπώς οι αλλαγές μπορούν να συμβούν στα δύο τελευταία επίπεδα εσωτερικό και εννοιολογικό χωρίς ο χρήστης (διαχειριστής, προγραμματιστής ή άλλος) που βρίσκεται στο εξωτερικό επίπεδο να γνωρίζει για αυτές τις αλλαγές. Έτσι έχουμε τη φυσική ανεξαρτησία των δεδομένων με μεταβολές στο εσωτερικό επίπεδο (φυσικό επίπεδο) χωρίς να επηρεάζεται το εννοιολογικό επίπεδο. Παράλληλα έχουμε και τη λογική ανεξαρτησία των δεδομένων στην οποία μπορούμε να έχουμε αλλαγές στη λογική αναπαράσταση και διασύνδεση των δεδομένων, χωρίς ο χρήστης του εξωτερικού επιπέδου να υποστεί κάποια αλλαγή στον τρόπο με τον οποίο εκτελεί τα αιτήματα προς το σύστημα.

Το **μοντέλο Οντοτήτων Συσχετίσεων** αναπτύχθηκε με σκοπό να παρέχει μία εννοιολογική απεικόνιση της δομής των δεδομένων και των σχέσεων που διέπουν τα δεδομένα, λαμβάνοντας υπόψιν και περιορισμούς που μπορεί να προκύψουν στις σχέσεις αυτές. Οποιοσδήποτε δομές δεδομένων από τον πραγματικό κόσμο που έχουν ιδιότητες τέτοιες ώστε να χαρακτηρίζουν ομάδες δεδομένων, θα αποτελούν τις οντότητες. Οι ιδιότητες – χαρακτηριστικά θα αποτελούν τα γνωρίσματα της οντότητας. Κάθε οντότητα θα χαρακτηρίζει μοναδικά τα στοιχεία της με ένα ή περισσότερα ξεχωριστά χαρακτηριστικά τα οποία θα είναι τα μοναδικά κλειδιά των στοιχείων. Οι οντότητες σχετίζονται μεταξύ τους μέσω σχέσεων, οι σχέσεις αυτές αποτελούν μία εννοιολογική σύνδεση των οντοτήτων. Κάθε συσχέτιση έχει ως στοιχεία τον αριθμός των οντοτήτων κάθε πλευράς που συμμετέχουν στη συσχέτιση αυτή (βαθμός). Ο βαθμός μιας συσχέτισης μπορεί να είναι:

- 1-1 (ένα-προς-ένα): Για κάθε ένα στοιχείο της μίας οντότητας έχουμε ένα και μόνο στοιχείο στην προς συσχέτιση οντότητα.
- 1-N (ένα-προς-πολλά): Για κάθε ένα στοιχείο της μίας οντότητας, έχουμε ένα ή πολλά στοιχεία στην προς συσχέτιση οντότητα.
- M-N (πολλά-προς-πολλά): Για κάθε στοιχείο της μίας οντότητας, έχουμε κανένα, ένα ή πολλά στοιχεία στην προς συσχέτιση οντότητα και αντίστροφα.

Για τη σχηματική απεικόνιση των παραπάνω κάνουμε χρήση των κάτωθι σχημάτων:

Παραλληλόγραμμο



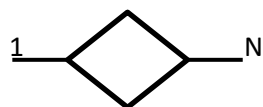
Οντότητα: Μας ορίζει μία οντότητα, το όνομα της οποίας το αναγράφουμε στο εσωτερικό του.

Κύκλος ή έλλειψη



Γνώρισμα: Αποτελεί ένα γνώρισμα, δηλαδή ένα χαρακτηριστικό που περιγράφει την οντότητα. Η ονομασία του γνωρίσματος αναγράφεται στο εσωτερικό του κύκλου-έλλειψης και συνδέεται με την οντότητα. Αν το γνώρισμα αποτελεί τμήμα του κλειδιού, τότε αυτό υπογραμμίζεται.

Ρόμβος



Συσχέτιση: Περιγράφει τη σχέση που υπάρχει μεταξύ των οντοτήτων και σε κάθε σκέλος της αναγράφεται ο βαθμός της συσχέτισης 1-1,1-N,N-M. Συνήθως η σχέση που περιγράφει ο ρόμβος, εννοιολογικά είναι το ρήμα που χρησιμοποιούμε στην περιγραφή των δεδομένων που ανήκουν σε δύο οντότητες. Π.χ. στην έκφραση «ο αποστολέας στέλνει αποστολή» έχουμε δύο

οντότητες (αποστολέας , αποστολή) με σχέση «στέλνει» όπου ένας αποστολέας μπορεί να στείλει κανένα, ένα ή και περισσότερα δέματα.

Πίνακας 11 Βασικά Σύμβολα διαγράμματος Οντοτήτων Συσχετίσεων

Όπως αναφέρθηκε, ο βαθμός της συσχέτισης μπορεί να αναγραφεί στις γραμμές με τις οποίες συνδέεται ο ρόμβος με τα σχετιζόμενα παραλληλόγραμμα. Υπάρχουν όμως περιπτώσεις στις οποίες το είδος συσχέτισης μπορεί να ορίζει κανένα ή ένα, ή αυστηρά ένα. Έτσι μπορούμε να κάνουμε χρήση των παρακάτω συμβόλων τα οποία αποτελούν τη σημειογραφία Crow's foot. Σε αυτό τον τρόπο αναπαράστασης στις άκρες των γραμμών μίας σχέσης σημειώνουμε σύμβολα που δείχνουν χαρακτηριστικά, όπως πολλαπλότητα και συσχέτιση.

Σύμβολο	Χαρακτηρισμός
	Κανένα (0)
	Ένα (1)
	Ένα ή κανένα
	Ένα και μόνο ένα
	Πολλά
	Ένα ή πολλά
	Κανένα ή πολλά

Πίνακας 12 Σημειογραφία Crow's foot

Το **Σχεσιακό μοντέλο** είναι μία προσέγγιση στην δόμηση των δεδομένων και των σχέσεων που διέπει τα δεδομένα και χαρακτηρίζεται από την απλότητά της. Οι οντότητες και τα γνωρίσματα που περιγράψαμε στο μοντέλο οντοτήτων-συσχετίσεων, σε αυτό το επίπεδο απεικονίζονται σε δομές που θα ονομάζονται πίνακες/σχέσεις και τα γνωρίσματα απεικονίζονται σε στήλες/χαρακτηριστικά του πίνακα. Με αυτό τον τρόπο έχουμε την αναπαράσταση της δομής με πίνακα οι στήλες του οποίου (γνωρίσματα) θα αποτελούν τα χαρακτηριστικά των δεδομένων μας. Κάθε σύνολο στοιχείων που αναπαρίσταται σε μία γραμμή εντός του πίνακα θα αποτελεί μία πλειάδα. Σημαντικό στοιχείο των γνωρισμάτων είναι ο τύπος τους αλλά και το πεδίο τιμών τους. Ένα σύνολο από τέτοιους πίνακες-σχέσεις θα αποτελούν τη βάση δεδομένων της εφαρμογής που αναπτύχθηκε στο πλαίσιο της πτυχιακής εργασίας.

Όπως και στο μοντέλο οντοτήτων συσχετίσεων οι πλειάδες θα χαρακτηρίζονται από τα κλειδιά τα οποία μπορούμε να θέσουμε και τα οποία είναι μοναδικά. Οι σχέσεις-πίνακες συσχετίζονται μέσω των κλειδιών τους και δημιουργούν νέα στιγμιότυπα από πλειάδες. Η συσχέτιση αυτή εύκολα μπορεί να αναπαρασταθεί μέσω της σημειογραφίας Crow's foot.

4 Ταχυμεταφορές, τρόπος λειτουργίας, σχεδιασμός εφαρμογής

Οι ταχυμεταφορές είναι ένας κλάδος που δραστηριοποιείται στον χώρο του παραδοσιακού ταχυδρομείου και μίας μεταφορικής εταιρίας. Δανείζεται στοιχεία των εργασιών κάθε κατηγορίας ξεχωριστά δημιουργώντας τον δικό του τρόπο εργασίας. Τα αντικείμενα που μπορούν να μεταφερθούν από μία ταχυμεταφορική μπορεί να είναι από μία μικρή επιστολή έως ένα μεγάλο κιβώτιο με διαστάσεις τέτοιες που να μην μπορεί να εξυπηρετηθεί από το παραδοσιακό ταχυδρομείο και σε άλλη περίπτωση θα έπρεπε να αποσταλεί με μία μεταφορική. Η μεταφορά του αντικειμένου θα πρέπει να γίνεται με αυξημένη ασφάλεια και με μεγάλη ταχύτητα και κατά την διάρκεια της μεταφοράς να ενημερώνεται τόσο ο παραλήπτης όσο και ο αποστολέας για την εκάστοτε θέση του. Η παροχή της υπηρεσίας αυτής έχει αυξημένο κόστος σε σχέση με το ταχυδρομείο ή τη μεταφορική καθώς οι υπηρεσίες που προσφέρονται τελικά είναι διαφορετικές και πολλές φορές εξειδικευμένες ανά πελάτη. Σημαντικό παράδειγμα εξειδικευμένης παροχής υπηρεσιών από την πλευρά των ταχυμεταφορών αποτελεί η μεταφορά βιολογικού υλικού από ένα μέρος σε άλλο αφού θα πρέπει να γίνει με ασφάλεια, ταχύτητα και χωρίς λάθη κατά τη μεταφορά, με άμεση ενημέρωση των εμπλεκόμενων, αποστολέα και παραλήπτη, για τη θέση του σε πραγματικό (ή «σχεδόν πραγματικό») χρόνο. Γεωγραφικά οι εταιρίες αυτές μπορούν να δραστηριοποιούνται στα όρια πόλης, νομού, χώρας ή και παγκόσμια.

Ρυθμιστική αρχή για τον τρόπο λειτουργίας των ταχυμεταφορών είναι η Εθνική Επιτροπή Τηλεπικοινωνιών και Ταχυδρομείων (ΕΕΤΤ). Από την ιστοσελίδα της μπορούμε να αντλήσουμε τον τρόπο λειτουργίας των εταιριών και συγκεκριμένα σύμφωνα με το έγγραφο «Υποχρεώσεις αδειοδοτημένων επιχειρήσεων» στην παράγραφο 3 και 4 ^[1] προκύπτουν τα ακόλουθα στοιχεία:

“ 3. τηρούν Ειδικό Σύστημα Παρακολούθησης και Εντοπισμού Ταχυδρομικών Αντικειμένων (Ε.Σ.Π.Ε.Τ.Α.) το οποίο:

α) να επιτρέπει την παρακολούθηση και τον εντοπισμό του ταχυδρομικού αντικειμένου ταχυμεταφορών και την ενημέρωση των χρηστών/καταναλωτών για τη διακίνηση της αποστολής τους, μέσω ιδίως διαδικτύου, ηλεκτρονικού ταχυδρομείου και τηλεφώνου και να ενημερώνεται για κάθε μεταβολή των στοιχείων που αφορούν την ταυτότητα ή τη διακίνηση του ταχυδρομικού αντικειμένου, κατά τη διαδρομή του από τον αποστολέα έως τον παραλήπτη. Η καταχώρηση του ταχυδρομικού αντικειμένου είναι μονοσήμαντα ορισμένη στο Ε.Σ.Π.Ε.Τ.Α. της αδειοδοτημένης επιχείρησης και του δικτύου της,

β) σε περίπτωση συνεργασίας με άλλες ταχυδρομικές επιχειρήσεις, να ανταλλάσσει με τα πληροφοριακά συστήματα αυτών πληροφορίες που έχουν σχέση με την παρακολούθηση και τον εντοπισμό του ταχυδρομικού αντικειμένου και με την πληροφόρηση των χρηστών/καταναλωτών

γ) να έχει τη δυνατότητα ηλεκτρονικής καταχώρησης, διαχείρισης και διακράτησης για χρονικό διάστημα όχι μικρότερο των έξι (6) μηνών, ιδίως, των παρακάτω πληροφοριών για κάθε ταχυδρομικό αντικείμενο:

- αριθμό ή κωδικό ταυτοποίησης ταχυδρομικού αντικειμένου,
- στοιχεία αποστολέα/ παραλήπτη,
- ημερομηνία, τόπο και ώρα παραλαβής,
- ημερομηνία, τόπο και ώρα επίδοσης και όνομα ατόμου που παρέλαβε το ταχυδρομικό αντικείμενο,
- βάρος ή κλιμάκιο βάρους ταχυδρομικού αντικειμένου,- είδος ταχυδρομικού αντικειμένου με ελάχιστη διάκριση σε φάκελο/δέμα και
- ημερομηνία/ες και ώρα/ες αποτυχημένων προσπαθειών επίδοσης.

4. προσαρτούν Συνοδευτικό Δελτίο Ταχυμεταφοράς (ΣΥ.ΔΕ.ΤΑ.) στο ταχυδρομικό αντικείμενο το οποίο πρέπει να φέρει, κατ' ελάχιστο:

α) την επωνυμία ή/ και το διακριτικό τίτλο, τον Αριθμό Φορολογικού Μητρώου και τα στοιχεία επικοινωνίας της επιχείρησης-κατόχου της Γενικής Άδειας,

β) την ένδειξη ΕΕΤΤ: “Γενική Άδεια Ταχυδρομικών Υπηρεσιών, ΕΕΤΤ, ΑΜ: ΧΧ-ΧΧΧ”, με κατάλληλου μεγέθους γραμματοσειρά, ώστε να διακρίνεται ευκρινώς από τους χρήστες,

γ) το μοναδικό αριθμό ή κωδικό ταυτοποίησης ταχυδρομικού αντικειμένου που είναι απαραίτητος για την παρακολούθηση και τον εντοπισμό του ταχυδρομικού αντικειμένου από το Ε.Σ.Π.Ε.Τ.Α. της επιχείρησης,

δ) τα στοιχεία του αποστολέα και του παραλήπτη,

ε) στις περιπτώσεις που αποστέλλονται ταχυδρομικά αντικείμενα μαζικώς, βάσει σύμβασης, είναι δυνατή η χρήση απλοποιημένου ΣΥ.ΔΕ.ΤΑ., το οποίο θα προσαρτάται σε κάθε ένα από τα αντικείμενα, και θα φέρει υποχρεωτικά τουλάχιστον το μοναδιαίο αριθμό αποστολής, το διακριτικό τίτλο και την ένδειξη ΕΕΤΤ. Οι επιχειρήσεις ταχυμεταφορών υποχρεούνται να διατηρούν τα αποκόμματα ΣΥ.ΔΕ.ΤΑ. στο αρχείο τους, σε ηλεκτρονική ή έντυπη μορφή, για ελάχιστη χρονική περίοδο έξι (6) μηνών, εκτός αν ορίζεται διαφορετικά σε άλλες διατάξεις. ”

Τα παραπάνω μας παρέχουν μία αρχική κατεύθυνση του τρόπου εργασίας της πλατφόρμας μας η οποία θα πρέπει να είναι σύννομη. Ακόμα και στην περίπτωση που κάποιες από τις παραπάνω εργασίες δεν υλοποιηθούν, θα πρέπει το σύστημα να είναι ανοικτό σε αναβάθμιση όταν αυτό κριθεί αναγκαίο.

4.1 Συνοδευτικού Δελτίου Ταχυμεταφοράς ΣΥ.ΔΕ.ΤΑ.

Το συνοδευτικό δελτίο ταχυμεταφοράς (ΣΥ.ΔΕ.ΤΑ.) περιλαμβάνει όλη την πληροφορία που πρέπει να γνωρίζει ο αποστολέας, ο παραλήπτης αλλά και οι ενδιάμεσα εμπλεκόμενοι φορείς (εταιρίες), ώστε το δέμα να παραδοθεί σωστά. Οι πληροφορίες που πρέπει να έχει, περιγράφονται στο έγγραφο περιγραφής προτύπου S10, “Identification of postal items – 13-character identifier”². Η όποια παρέμβαση

² <https://www.upu.int/UPU/media/upu/files/postalSolutions/programmesAndServices/standards/S10-12.pdf>

γίνει στα στοιχεία του εντύπου θα πρέπει να είναι τέτοια που να μην αλλοιώνει τον χαρακτήρα του εντύπου και των στοιχείων του, αλλά να προσθέτει περαιτέρω πληροφορία για χρήση από κάθε εμπλεκόμενο. Από τα στοιχεία αυτά ξεχωρίζουμε δύο στοιχεία του προτύπου και ένα νέο που θα προσθέσουμε εμείς για χρήση από νεότερα μέσα τεχνολογίας.

Το πρώτο στοιχείο είναι ο διεθνής αριθμός παρακολούθησης, ο οποίος είναι μία απαραίτητη πληροφορία που έχει συγκεκριμένη μορφή, και ο τρόπος δημιουργίας του πρέπει να εγγυάται ώστε η τιμή του να είναι μοναδική μέσα σε όλο το δίκτυο διανομής του δέματος για μία χρονική περίοδο τουλάχιστον δώδεκα μηνών.

Το δεύτερο στοιχείο είναι η εκτύπωση του αριθμού παρακολούθησης, η οποία θα πρέπει να έχει συγκεκριμένο τρόπο και να αποτυπώνεται με την μορφή γραμμωτού κώδικα (bar code) σε τύπους ISO/IEC 16388 code 39 ή UCC/EAN-128. Στην παρούσα εργασία θα κάνουμε χρήση του ISO/IEC 16388 code 39.

Τέλος θα προσθέσουμε ένα ακόμα πληροφοριακό στοιχείο, την εκτύπωση του QR-code το οποίο θα μας κατευθύνει στην κεντρική σελίδα της πλατφόρμας. Τη λειτουργία του θα την παρουσιάσουμε σε μεταγενέστερο σημείο αφού η λειτουργικότητα που παρέχεται με τη σάρωσή του θα είναι διαφορετική για κάποιο χρήστη-επισκέπτη και διαφορετική για χρήστη εγκεκριμένο από το σύστημα.

Συνεπώς η βασική μορφή που θα πρέπει να έχει ένα τέτοιο συνοδευτικό δελτίο είναι η κάτωθι.

ΣΥΝΟΔΕΥΤΙΚΟ ΔΕΛΤΙΟ ΤΑΧΥΜΕΤΑΦΟΡΑΣ

ΑΠΟΣΤΟΛΕΑΣ	ΠΑΡΑΛΗΠΤΗΣ
ΠΑΡΑΤΗΡΗΣΕΙΣ	ΠΑΡΑΚΟΛΟΥΘΗΣΗ 
 L A 0 0 0 0 0 0 1 4 G R	

Εικόνα 7 Ενδεικτική εικόνα συνοδευτικού δελτίου ταχυμεταφοράς

4.1.1 Διεθνής Αριθμός Παρακολούθησης

Η παρακολούθηση ενός πακέτου είναι μία βασική διαδικασία που κάνει τον πελάτη μιας μεταφορικής εταιρίας να είναι βέβαιος για την ορθή και έγκαιρη μεταφορά των αποστολών του, αλλά ταυτόχρονα του δίνει τη δυνατότητα να οργανώνει καλύτερα τις εργασίες του, γνωρίζοντας την ακριβή

διαδρομή και θέση των αποστολών του. Οι μεταφορές αυτές πραγματοποιούνται σε παγκόσμια κλίμακα, συνεπώς είναι απαραίτητο να χρησιμοποιείται αριθμοδότηση του προς αποστολή πακέτου με έναν ενιαίο τρόπο, ώστε να είναι άμεσα ανιχνεύσιμο από όλους. Η λύση σε αυτό το πρόβλημα για τις διεθνείς αριθμοδοτήσεις δόθηκε από τη Universal Postal Union με το πρότυπο «UPU S10 standard». Το πρότυπο αυτό δίνει τη δυνατότητα σε κάθε χώρα να αποδίδει αριθμούς κεντρικά ένας οργανισμός (για την Ελλάδα υπεύθυνος για αυτό είναι τα Ελληνικά Ταχυδρομεία) και ο αριθμός αυτός θα έχει συγκεκριμένη μορφή και δεν θα επαναχρησιμοποιείται σε διάστημα μικρότερο των 12 μηνών. Το πρότυπο αυτό παράγει αριθμούς παρακολούθησης δεκατριών αλφαριθμητικών ψηφίων, η μορφή των οποίων φαίνεται στον παρακάτω πίνακα:

Πρότυπο S10 (13-ψηφία)			
ΧΧ	(0...9) – 8 Αριθμητικά	1 ψηφίο	ΥΥ
Κωδικός παρεχόμενης υπηρεσίας	Σειριακός αριθμός πακέτου	Ψηφίο ελέγχου	Κωδικός χώρας με την μορφή 2 ψηφίων ISO

Πίνακας 13 Πρότυπο UPU S10

Οι κωδικοί για τις παρεχόμενες υπηρεσίες είναι:

Παρεχόμενες υπηρεσίες (1 ^ο και 2 ^ο ψηφίο)	
Κωδικός ΧΧ	Υπηρεσία
AV-AZ	Χρήση εντός χώρας (όχι σε διεθνή εμβέλεια) με εφαρμογή στον εντοπισμό πακέτων μέσω ετικετών RFID (Radio Frequency Identification)
BA-BZ, DA-DZ GA, GD, NA-NZ PA-PZ, ZA-ZZ	Χρήση εντός χώρας (όχι σε διεθνή εμβέλεια)
CA-CZ	Αφορά διεθνή μεταφορά πακέτου βαρύτερου των 2Kgr.
EA-EZ	Express Mail Service (EMS). Οι κωδικοί EX-EZ λειτουργούν έπειτα από διμερείς συμφωνίες ανάμεσα σε διαφορετικές εταιρίες.
HA-HZ	Αφορούν δέματα ηλεκτρονικού εμπορίου. Οι κωδικοί HX-HY απαιτούν πολυμερείς συμφωνίες ανάμεσα σε εταιρίες ενώ οι HZ διμερείς συμφωνίες.
JA-JZ, KA-KZ, SA-SZ, TA-TZ WA-AZ	Κατοχυρωμένοι κωδικοί άλλων εταιριών-οργανισμών και δεν μπορούν να χρησιμοποιηθούν από το πρότυπο.
LA-LZ	Χρήση για επιστολές και απλά γράμματα (επιστολικά) τα οποία όμως μπορούν να έχουν την δυνατότητα παρακολούθησης. Η χρήση του κωδικού LZ απαιτεί διμερή συμφωνία εταιριών.
MA-MZ	Δηλώνει διακίνηση έντυπου υλικού με τη μορφή επιστολής/δέματος (π.χ. αποστολή βιβλίου)
QA-QM	Επιστολές που υπόκεινται σε IBRS (International Business Reply Service), δηλαδή αποστέλλονται στον παραλήπτη ο οποίος αφού κάνει κάποιες εργασίες το επιστρέφει όχι με δική του χρέωση αλλά με χρέωση του αποστολέα.
RA-RZ	Συστημένη αλλά όχι ασφαλισμένη επιστολή. Η χρήση του κωδικού RZ απαιτεί διμερή συμφωνία εταιριών.
VA-VZ	Ασφαλισμένες επιστολές. Η χρήση του κωδικού VZ απαιτεί διμερή συμφωνία εταιριών.

UA-UZ	Αφορά επιστολές ή δέματα που δεν εντάσσονται σε προηγούμενους κωδικούς και υπόκεινται σε τελωνιακό έλεγχο. Συνήθως συνοδεύονται από φόρμα δήλωσης προϊόντων.
-------	--

Πίνακας 14 Παρεχόμενες υπηρεσίες, 1^ο και 2^ο ψηφίο προτύπου S10

Οι αριθμοί παρακολούθησης που δεν αρχίζουν από ένα κωδικό από τον παραπάνω πίνακα δεν ακολουθούν το διεθνές πρότυπο. Τέτοιοι αριθμοί είναι της UPS που ξεκινούν με το 1Z, έχουν δική τους γραμμογράφηση και χρησιμοποιούνται μόνο από αυτή (ενδοεταιρικά).

Τα δύο ψηφία ΥΥ στη 12^η και 13^η θέση, υποδεικνύουν τη χώρα του αποστολέα, με διψήφια αναπαράσταση που ορίζεται από το ISO 3166-1 alpha-2. Σύμφωνα λοιπόν με αυτό η Ελλάδα θα περιγράφεται ως GR. Εκτός από τους κωδικούς χωρών υπάρχουν και άλλοι κωδικοί που προκύπτουν από τον πίνακα κωδικών ISO 3166-1 alpha-2 και οι οποίοι έχουν εκχωρηθεί σε εταιρίες ή οργανισμούς όπως EU-Europe, FX-France, UN- United Nations και άλλα.

Αριθμοί 3^{ης} έως και 10^{ης} θέσης θα είναι σειριακοί αριθμοί τέτοιοι που να μην επαναλαμβάνονται τουλάχιστον για τους επόμενους δώδεκα μήνες. Με βάση τους αριθμούς αυτούς θα παραχθεί και το ψηφίο ελέγχου στην 11^η θέση.

Ψηφίο ελέγχου 11^{ης} θέσης παράγεται από τους προηγούμενους οκτώ με τον εξής αλγόριθμο:

Αντιστοιχίζουμε τις θέσεις 3 έως 10 με τα βάρη {8 6 4 2 3 5 9 7}
Πολλαπλασιάζουμε τους αριθμούς των θέσεων με τα αντίστοιχα βάρη
Προσθέτουμε τα γινόμενα όλων των πολλαπλασιασμών
Υπολογίζουμε το υπόλοιπο της διαίρεσης του αθροίσματος με το 11
Αφαιρούμε το υπόλοιπο της διαίρεσης από το 11.
Αν το αποτέλεσμα είναι 10 τότε θέση αποτέλεσμα =0
Αν το αποτέλεσμα είναι 11 τότε θέση αποτέλεσμα =5
Αν το αποτέλεσμα δεν είναι ούτε 10 ούτε 11, το χρησιμοποιούμε ως έχει

Η συνάρτηση σε php που θα υπολογίσει τον 11^ο αριθμό θα είναι με βάση τον παραπάνω αλγόριθμο.

```
<?php
function CheckDigit($nDigit)
{
    $nDigit11=null;
    $aWeights=array(8,6,4,2,3,5,9,7);
    $anums=str_split($nDigit,1);
    $sum=0;
    for ($i=0;$i<8;$i++){ $sum=$sum+$aWeights[$i]*$anums[$i]; }
    $nDigit11=11- ($sum % 11);
    $nDigit11=($nDigit11==10?0:$nDigit11);
    $nDigit11=($nDigit11==11?5:$nDigit11);
    return $nDigit11;
}
?>
```

4.1.2 Ραβδωτός ή γραμμωτός κώδικας

Ο διεθνής αριθμός παρακολούθησης που θα παραχθεί από την παραπάνω διαδικασία θα πρέπει να αποτυπωθεί με την μορφή ραβδωτού κώδικα. Δηλαδή να αναπαρασταθεί από μία σειρά κάθετων γραμμών που ανάλογα με το πάχος τους και τη διάταξή τους ορίζουν και διαφορετικό αλφαριθμητικό. Για την ανάγνωση των κωδικών αυτών, χρησιμοποιούμε συσκευές scanner ή εφαρμογές κινητών τηλεφώνων που έχουν τη δυνατότητα της ανάγνωσής τους, μέσω κάμερας. Αυτές οι δυνατότητες που έχουμε, συνδυαζόμενες με το γεγονός ότι ο κωδικός αυτός είναι μοναδικός επιτρέπει να διαχειριζόμαστε εύκολα τα προς αποστολή πακέτα και να ανιχνεύσουμε την κατάσταση που βρίσκονται.

Σύμφωνα με το UPU S10 μπορούμε να κάνουμε χρήση του code 39 ή code 128. Λόγω απλότητας θα χρησιμοποιηθεί ο code 39 ο οποίος είναι ο πρώτος αλφαριθμητικός κώδικας που κυκλοφόρησε και σημείωσε επιτυχία. Η απλότητά του στηρίζεται στη μέση πυκνότητα των γραμμών και την έλλειψη ψηφίου ελέγχου στο τέλος (checksum). Με αυτόν μπορούμε να αναπαραστήσουμε έως 43 αλφαριθμητικά με τον περιορισμό ο πρώτος και ο τελευταίος χαρακτήρας να είναι αστερίσκος (*).

Η ενσωμάτωση του κώδικα αυτού στην πλατφόρμα μας θα γίνει μέσω της γραμματοσειράς “Libre Barcode 39 Extended Text” που παρέχεται από την google.com προς χρήση δωρεάν με Open Font License (OFL) άδεια χρήσης. Συνεπώς η ενσωμάτωση θα γίνει ως εξής:

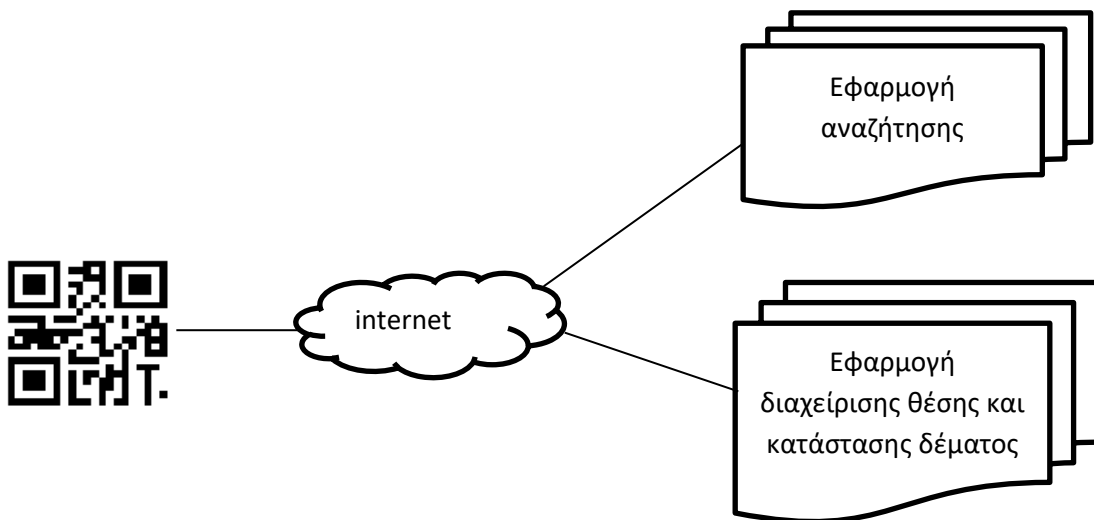
```
<head>
<link
href="https://fonts.googleapis.com/css2?family=Libre+Barcode+39+
Extended+Text&display=swap" rel="stylesheet">
<style type="text/css">
.barcode39{
    font-family: 'Libre Barcode 39 Extended Text', cursive;
    font-size: 40px;
}
</style>
</head>
...
<div class="barcode39">*LA000000014GR*</div>
```

4.1.3 QR κωδικός

Σε αντίθεση με τους ραβδωτούς κώδικες που είδαμε προηγουμένως και οι οποίοι είναι μίας διάστασης (1D), υπάρχουν και οι κώδικες δύο διαστάσεων (2D), οι οποίοι αποτελούν ένα πλέγμα μικρών κουκίδων σε ορθογώνιο ή παραλληλόγραμμο σχήμα. Η πιο γνωστή μορφή δισδιάστατου κώδικα είναι το QR-code, το οποίο χρησιμοποιεί τετράγωνο σχήμα. Για να μπορέσει να έχει το σχήμα αυτό, προσανατολισμό, ώστε να υπάρχει αρχική θέση ανάγνωσης, έχουν προστεθεί στις τρεις ή και στις τέσσερις γωνίες μικρά τετράγωνα. Ο κώδικας αυτός έχει αποκτήσει μεγάλη δημοτικότητα για τους ακόλουθους δύο λόγους:

- Αναπαράσταση μεγάλου όγκου πληροφορίας σε πολύ μικρό χώρο. Υπάρχει η δυνατότητα αποθήκευσης και αναπαραγωγής 4.296 αλφαριθμητικών με τη μορφή κειμένου ή 7.089 αριθμών.
- Δεν χρειάζεται εξειδικευμένη συσκευή ανάγνωσης παρά μόνο η ύπαρξη κάμερας και σχετικού λογισμικού. Η ευελιξία αυτή σε συνδυασμό με τη μικρή επιφάνεια που καταλαμβάνει η αναπαράσταση του κώδικα έχει βρει μεγάλη εφαρμογή τους περισσότερους τομείς δραστηριοτήτων όπου με ένα κινητό μπορούμε να αναγνώσουμε την πληροφορία σε οποιοδήποτε σημείο και να βρεθούμε.

Το UPU S10 δεν προβλέπει την ύπαρξή του αλλά ούτε και απαγορεύει την αναπαράστασή του στο έντυπο. Έτσι, στην εκτύπωση του συνοδευτικού δελτίου θα συμπεριλάβουμε την αναπαράσταση πληροφορίας με τη μορφή qr-code. Η πληροφορία που θα αποτυπώσουμε θα είναι μία μοναδική διαδρομή προς τον ιστόχωρο της πλατφόρμας με ερώτημα για τον συγκεκριμένο αριθμό παρακολούθησης. Για τους χρήστες- επισκέπτες (guest) θα παρέχουμε πληροφορίες σχετικά με τη θέση και την κατάσταση του πακέτου ενώ για τους εργαζόμενους ταχυμεταφορείς θα υπάρχει η δυνατότητα αλλαγής κατάστασης του δέματος (παράδοση , επιστροφή κλπ).



Εικόνα 8 Εφαρμογές, χρήστες και διασυνδέσεις

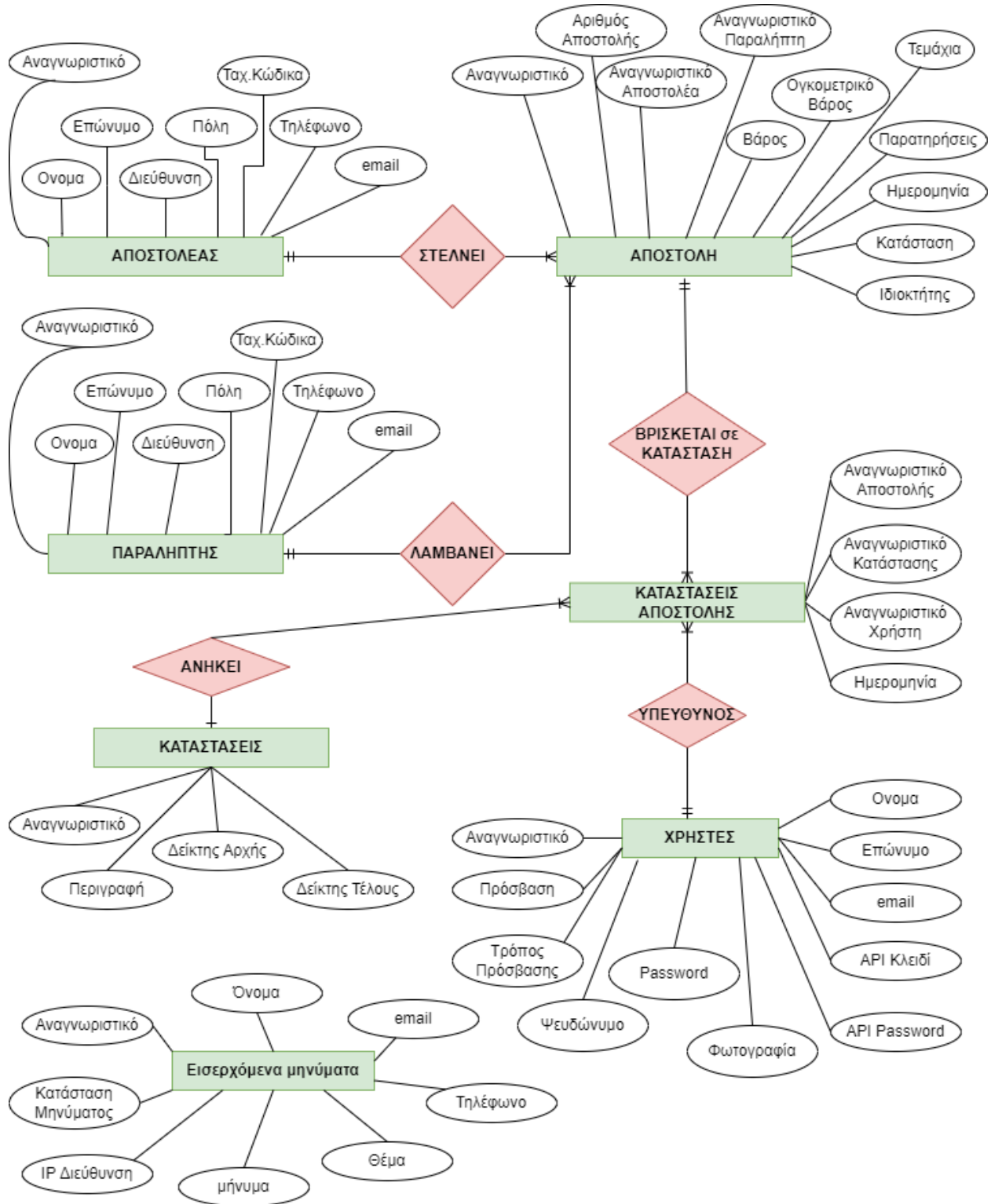
Η δημιουργία και ενσωμάτωση του QR-code στην πλατφόρμα μας θα γίνει μέσω του QR Codes API που παρέχει η Google. Με αυτό έχουμε τη δυνατότητα να δημιουργήσουμε κωδικό QR και να τον αναπαραστήσουμε ως εικόνα σε κώδικα HTML. Σύμφωνα με αυτά, η ενσωμάτωση θα γίνει ως εξής:

```

```

4.2 Δεδομένα αποστολών, σχεδιασμός βάσης

Το διάγραμμα οντοτήτων συσχετίσεων για την εφαρμογή μας με βάση τη σημειογραφία Crow's foot (πρβλ. ενότητα 3.6), είναι όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 9 Διάγραμμα οντοτήτων - συσχετίσεων

Το παραπάνω διάγραμμα απαντά στις απαιτήσεις που έχουμε από το σύστημα. Συγκεκριμένα

- Κάθε φυσικό ή νομικό πρόσωπο μπορεί να είναι αποστολέας δέματος, επιστολής ή οποιουδήποτε άλλου αντικειμένου.
- Κάθε φυσικό ή νομικό πρόσωπο μπορεί να είναι παραλήπτης δέματος, επιστολής ή οποιουδήποτε άλλου αντικειμένου.
- Κάθε αντικείμενο που αποστέλλεται από κάποιον αποστολέα προς κάποιο παραλήπτη αποκτά ένα μοναδικό αριθμό, ο οποίος διατηρείται για κάποιο χρονικό διάστημα στη βάση δεδομένων μας.
- Κάθε απεσταλμένο αντικείμενο μπορεί να εντοπιστεί ως προς το που βρίσκεται κατά τη διάρκεια της μεταφοράς του από τον αποστολέα στον παραλήπτη.
- Σε κάθε αλλαγή θέσης του απεσταλμένου αντικειμένου υπάρχει ένας χρήστης του προγράμματος (μεταφορέας) ο οποίος έχει υπό την ευθύνη του το αντικείμενο.
- Μπορούμε να γνωρίζουμε τη θέση αντικειμένων που έχουν αποσταλεί από άλλες μεταφορικές εταιρίες, αρκεί να ξέρουμε τον αριθμό δέματος.
- Η πρόσβαση στα δεδομένα θα γίνεται από εξουσιοδοτημένους χρήστες.

Με βάση το παραπάνω διάγραμμα οι πίνακες που δημιουργήθηκαν είναι οι εξής.

- Πίνακας **contactus**

Στον πίνακα αυτό καταχωρούνται όλα τα εισερχόμενα μηνύματα που δημιουργούνται από την κεντρική σελίδα στην επιλογή “Επικοινωνία”. Η πρόσβαση στην αποστολή έχουν όλοι οι επισκέπτες της ιστοσελίδας και τα μηνύματα αφορούν τον κεντρικό διαχειριστή

Στήλη	Τύπος	Κενό	Προεπιλογή	Περιγραφή
id	int(11)	Όχι		Μοναδικός αύξων αριθμός (αναγνωριστικό)
username	varchar(100)	Όχι		Όνομα αποστολέα μηνύματος
useremail	varchar(100)	Όχι		Email αποστολέα μηνύματος
userphone	varchar(50)	Όχι		Τηλέφωνο αποστολέα μηνύματος
subject	text	Όχι		Θέμα μηνύματος
message	text	Όχι		Κύριο σώμα μηνύματος
userip	varchar(20)	Όχι		Public IP του αποστολέα μηνύματος
messagestatus	int(1)	Όχι	1	Κατάσταση μηνύματος 0-διαβασμένο και 1-αδιάβαστο (Νέο)

Πίνακας 15 Πίνακας contactus

- Πίνακας **senders**

Στον πίνακα αυτό καταχωρούνται οι αποστολείς δεμάτων.

Στήλη	Τύπος	Κενό	Προεπιλογή	Περιγραφή
id	int(11)	Όχι		Μοναδικός αύξων αριθμός (αναγνωριστικό)
fname	varchar(100)	Όχι		Όνομα αποστολέα
lname	varchar(100)	Όχι		Επώνυμο αποστολέα
address	varchar(200)	Όχι		Διεύθυνση αποστολέα
tk	varchar(10)	Όχι		Ταχυδρομικό κώδικα αποστολέα

town	varchar(50)	Όχι		Πόλη αποστολέα
phone	varchar(15)	Όχι		Τηλέφωνο αποστολέα
email	varchar(100)	Όχι		Email αποστολέα

Πίνακας 16 Πίνακας senders

- Πίνακας **receivers**

Στον πίνακα αυτό καταχωρούνται οι παραλήπτες των δεμάτων.

Στήλη	Τύπος	Κενό	Προεπιλογή	Περιγραφή
id	int(11)	Όχι		Μοναδικός αύξων αριθμός (αναγνωριστικό)
fname	varchar(100)	Όχι		Όνομα παραλήπτη
lname	varchar(100)	Όχι		Επώνυμο παραλήπτη
address	varchar(200)	Όχι		Διεύθυνση παραλήπτη
tk	varchar(10)	Όχι		Ταχυδρομικό κώδικα παραλήπτη
town	varchar(50)	Όχι		Πόλη παραλήπτη
phone	varchar(15)	Όχι		Τηλέφωνο παραλήπτη
email	varchar(100)	Όχι		Email παραλήπτη

Πίνακας 17 Πίνακας receivers

- Πίνακας **sydeta**

Στον πίνακα αυτό καταχωρούνται οι αποστολές έχοντας ως αναγνωριστικό τον μοναδικό αριθμό ΣΥ.ΔΕ.ΤΑ. το οποίο παράγεται από το πρόγραμμα.

Στήλη	Τύπος	Κενό	Προεπιλογή	Περιγραφή
id	int(11)	Όχι		Μοναδικός αύξων αριθμός (αναγνωριστικό)
trackingnumber	varchar(20)	Όχι		Μοναδικός αριθμός που παράγεται σύμφωνα με το κεφάλαιο 4.1.1
owner	varchar(10)	Όχι	SELF	SELF για αποστολές της ίδιας ταχυμεταφορικής εταιρίας και OTHER για αποστολές άλλων ταχυμεταφορικών εταιριών
senderid	int(11)	Όχι		Μοναδικός Α/Α αποστολέα
receiverid	int(11)	Όχι		Μοναδικός Α/Α παραλήπτη
weight	decimal(10,0)	Όχι		Βάρος αντικειμένου
dimweight	decimal(10,0)	Όχι		Ογκομετρικό βάρος αντικειμένου
pieces	int(11)	Όχι		Τεμάχια προς αποστολή
note	text	Όχι		Συνοδευτικές σημειώσεις
createtime	timestamp	Όχι	CURRENT_TIMESTAMP	Ημέρα και ώρα καταχώρισης.

status	int(1)	Όχι	1	Δείκτης κατάστασης δέματος 0 - Σε εξέλιξη και 1 - ολοκληρωμένη αποστολή.
---------------	--------	-----	---	--

Πίνακας 18 Πίνακας sydetata

- Πίνακας **checkpoints**

Στον πίνακα αυτό καταχωρούνται οι θέσεις-καταστάσεις που μπορούν να βρεθούν τα αποστέλλόμενα αντικείμενα

Στήλη	Τύπος	Κενό	Προεπιλογή	Περιγραφή
id	int(11)	Όχι		Μοναδικός αύξων αριθμός (αναγνωριστικό)
title	varchar(100)	Όχι		Περιγραφή(ονομασία) κατάστασης
initial	tinyint(1)	Όχι	0	Δείκτης χαρακτηρισμού κατάστασης ως «αρχική θέση» (0-Όχι, 1-Ναι)
ended	tinyint(1)	Όχι	0	Δείκτης χαρακτηρισμού κατάστασης ως «τελική θέση» (0-Όχι, 1-Ναι)

Πίνακας 19 Πίνακας checkpoints

- Πίνακας **_users**

Στον πίνακα αυτό καταχωρούμε τους πιστοποιημένους χρήστες του προγράμματος. Η καταχώριση είναι ανεξάρτητη από τον τρόπο εισόδου τους στο πρόγραμμα.

Στήλη	Τύπος	Κενό	Προεπιλογή	Περιγραφή
id	int(11)	Όχι		Μοναδικός αύξων αριθμός (αναγνωριστικό)
userlevel	int(11)	Όχι	0	Επίπεδα χρηστών. 0-επισκέπτης, 9-suppper user, 8-διαχειριστής.
createdfrom	varchar(20)	Όχι	SELF	Τρόπος εισαγωγής τους χρήστη SELF, GOOGLE, FACEBOOK
username	varchar(100)	Όχι		Ψευδώνυμο χρήστη
userpassword	varchar(200)	Όχι		Password χρήστη
useremail	varchar(200)	Όχι		Email χρήστη
fname	varchar(100)	Όχι		Όνομα χρήστη
lname	varchar(100)	Όχι		Επώνυμο χρήστη
userpicture	varchar(200)	Όχι		Εικονίδιο avatar χρήστη
releasecode	varchar(30)	Όχι		Κωδικός επιβεβαίωσης χρήστη
api_key	varchar(200)	Όχι		Κλειδί API για χρήση υπηρεσιών
api_password	varchar(200)	Όχι		Password API

Πίνακας 20 Πίνακας users

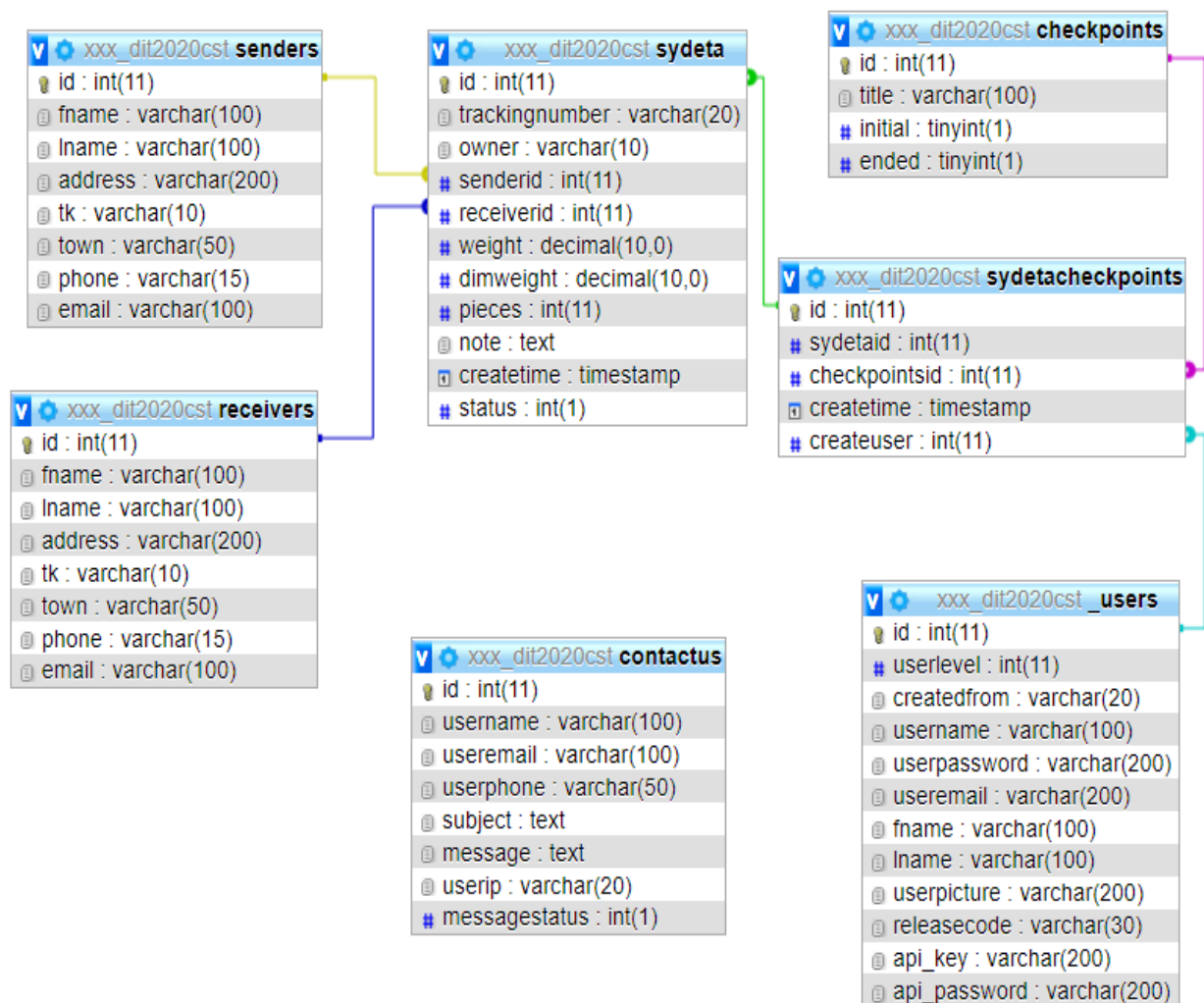
- Πίνακας **sydetacheckpoints**

Ο πίνακας αυτός αποτελεί τον συνδυαστικό κρίκο ανάμεσα στον πίνακα sydetata (πίνακας αποστολών), στον πίνακα checkpoints (πίνακας θέσεων) και στον πίνακα _users (πίνακας πιστοποιημένων χρηστών).

Στήλη	Τύπος	Κενό	Προεπιλογή	Περιγραφή
id	int(11)	Όχι		Μοναδικός αύξων αριθμός (αναγνωριστικό)
sydetaid	int(11)	Όχι		Μοναδικός Α/Α αποστολής
checkpointsid	int(11)	Όχι		Μοναδικός Α/Α κατάστασης
createtime	timestamp	Όχι	CURRENT_TIMESTAMP	Ημέρα και ώρα δημιουργίας της παρούσας κατάστασης
createuser	int(11)	Όχι		Χρήστης που την δημιούργησε.

Πίνακας 21 Πίνακας sydetacheckpoints

Οι παραπάνω πίνακες μπορούν να παρασταθούν στο πιο κάτω σχεσιακό μοντέλο.



Εικόνα 10 Σχεσιακό μοντέλο

Οι συσχετίσεις που προκύπτουν τόσο από το διάγραμμα ER όσο και από το σχεσιακό μοντέλο και συνδέουν τους πίνακες μεταξύ τους μπορούν να συνοψιστούν στον παρακάτω πίνακα

Πίνακες συσχετίσεων		Σχέση
senders (id)	sydeta (sendersid)	Είναι μία σχέση «ένα προς πολλά» καθώς ένας αποστολέας μπορεί να έχει πολλές αποστολές.
receivers(id)	sydeta(receiverid)	Είναι μία σχέση «ένα προς πολλά» καθώς ένας παραλήπτης μπορεί να έχει πολλές παραλαβές αποστολών.
sydeta(id)	sydetacheckpoints(sydetaid)	Μία σχέση «ένα προς πολλά» αφού μία αποστολή μπορεί να έχει πολλές ίδιες ή και διαφορετικές καταστάσεις.
checkpoints(id)	sydetacheckpoints(checkpointsid)	Σχέση «ένα προς ένα» καθώς κάθε checkpoint έχει ένα και μόνο σύνολο περιγραφικών παραμέτρων.
_users(id)	sydetacheckpoints(createuser)	Σχέση «ένα προς ένα» αφού κάθε checkpoint έχει δημιουργηθεί από ένα και μόνο χρήστη.

Πίνακας 22 Συσχετίσεις πινάκων εφαρμογής

Οι εντολές που πρέπει να δοθούν στην MySQL για να δημιουργηθούν οι πίνακες είναι οι εξής.

```
-- Δομή πίνακα για τον πίνακα `checkpoints`
CREATE TABLE `checkpoints` (
  `id` int(11) NOT NULL,
  `title` varchar(100) NOT NULL,
  `initial` tinyint(1) NOT NULL DEFAULT '0',
  `ended` tinyint(1) NOT NULL DEFAULT '0'
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

-- Ευρετήρια για πίνακα `checkpoints`
ALTER TABLE `checkpoints` ADD PRIMARY KEY (`id`);

-- AUTO_INCREMENT για πίνακα `checkpoints`
ALTER TABLE `checkpoints` MODIFY `id` int(11) NOT NULL AUTO INCREMENT;
```

```
-- Δομή πίνακα για τον πίνακα `contactus`
CREATE TABLE `contactus` (
  `id` int(11) NOT NULL,
  `username` varchar(100) NOT NULL,
  `useremail` varchar(100) NOT NULL,
  `userphone` varchar(50) NOT NULL,
  `subject` text NOT NULL,
  `message` text NOT NULL,
  `userip` varchar(20) NOT NULL,
  `messagestatus` int(1) NOT NULL DEFAULT '1' COMMENT '0-read, 1-unread'
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

-- Ευρετήρια για πίνακα `contactus`
ALTER TABLE `contactus` ADD PRIMARY KEY (`id`);

-- AUTO_INCREMENT για πίνακα `contactus`
ALTER TABLE `contactus` MODIFY `id` int(11) NOT NULL AUTO INCREMENT;
```

```

-- Δομή πίνακα για τον πίνακα `receivers`
CREATE TABLE `receivers` (
  `id` int(11) NOT NULL,
  `fname` varchar(100) NOT NULL,
  `lname` varchar(100) NOT NULL,
  `address` varchar(200) NOT NULL,
  `tk` varchar(10) NOT NULL,
  `town` varchar(50) NOT NULL,
  `phone` varchar(15) NOT NULL,
  `email` varchar(100) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

-- Ευρετήριο για πίνακα `receivers`
ALTER TABLE `receivers` ADD PRIMARY KEY (`id`);

-- AUTO_INCREMENT για πίνακα `receivers`
ALTER TABLE `receivers` MODIFY `id` int(11) NOT NULL AUTO INCREMENT;

```

```

-- Δομή πίνακα για τον πίνακα `senders`
CREATE TABLE `senders` (
  `id` int(11) NOT NULL,
  `fname` varchar(100) NOT NULL,
  `lname` varchar(100) NOT NULL,
  `address` varchar(200) NOT NULL,
  `tk` varchar(10) NOT NULL,
  `town` varchar(50) NOT NULL,
  `phone` varchar(15) NOT NULL,
  `email` varchar(100) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

-- Ευρετήριο για πίνακα `senders`
ALTER TABLE `senders` ADD PRIMARY KEY (`id`);

-- AUTO_INCREMENT για πίνακα `senders`
ALTER TABLE `senders` MODIFY `id` int(11) NOT NULL AUTO INCREMENT;

```

```

-- Δομή πίνακα για τον πίνακα `sydeta`
CREATE TABLE `sydeta` (
  `id` int(11) NOT NULL,
  `trackingnumber` varchar(20) NOT NULL,
  `owner` varchar(10) NOT NULL DEFAULT 'SELF',
  `senderid` int(11) NOT NULL,
  `receiverid` int(11) NOT NULL,
  `weight` decimal(10,0) NOT NULL,
  `dimweight` decimal(10,0) NOT NULL,
  `pieces` int(11) NOT NULL,
  `note` text NOT NULL,
  `createtime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  `status` int(1) NOT NULL DEFAULT '1' COMMENT '0-open, 1-close'
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

```

```

-- Ευρετήρια για πίνακα `sydeta`
ALTER TABLE `sydeta` ADD PRIMARY KEY (`id`);

-- AUTO_INCREMENT για πίνακα `sydeta`
ALTER TABLE `sydeta` MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;

```

```

CREATE TABLE `sydetacheckpoints` (
  `id` int(11) NOT NULL,
  `sydetaid` int(11) NOT NULL,
  `checkpointsid` int(11) NOT NULL,
  `createtime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `createuser` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Ευρετήρια για πίνακα `sydetacheckpoints`
ALTER TABLE `sydetacheckpoints` ADD PRIMARY KEY (`id`);

-- AUTO_INCREMENT για πίνακα `sydetacheckpoints`
ALTER TABLE `sydetacheckpoints`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;

```

```

-- Δομή πίνακα για τον πίνακα `_users`
CREATE TABLE `_users` (
  `id` int(11) NOT NULL,
  `userlevel` int(11) NOT NULL DEFAULT '0'
    COMMENT '0-usersite, 9-supper_admin 8-adminofsite',
  `createdfrom` varchar(20) NOT NULL DEFAULT 'SELF'
    COMMENT 'SELF, GOOGLE, FACEBOOK',
  `username` varchar(100) NOT NULL,
  `userpassword` varchar(200) NOT NULL,
  `useremail` varchar(200) NOT NULL,
  `fname` varchar(100) NOT NULL,
  `lname` varchar(100) NOT NULL,
  `userpicture` varchar(200) NOT NULL,
  `releasecode` varchar(30) NOT NULL,
  `api_key` varchar(200) NOT NULL,
  `api_password` varchar(200) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

-- Ευρετήρια για πίνακα `_users`
ALTER TABLE `_users` ADD PRIMARY KEY (`id`);

-- AUTO_INCREMENT για πίνακα `_users`
ALTER TABLE `_users` MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;

```

Με βάση τις σχέσεις που ορίσαμε πιο πάνω βασικές αναζητήσεις στην βάση μας περιγράφονται ως εξής.

```

-- Εμφάνιση όλων των καταστάσεων μίας αποστολής και τους χρήστες που
είχαν σχετική ευθύνη
select *,sydetacheckpoints.createtime as pointtime
  from sydetacheckpoints

```

```

left join sydetata on sydetata.id=sydetatacheckpoints.sydetataid
left join checkpoints on
            checkpoints.id=sydetatacheckpoints.checkpointsid
left join _users on _users.id=sydetatacheckpoints.createuser
where sydetata.trackingnumber='a tracking number'
order by sydetatacheckpoints.createtime DESC

```

```

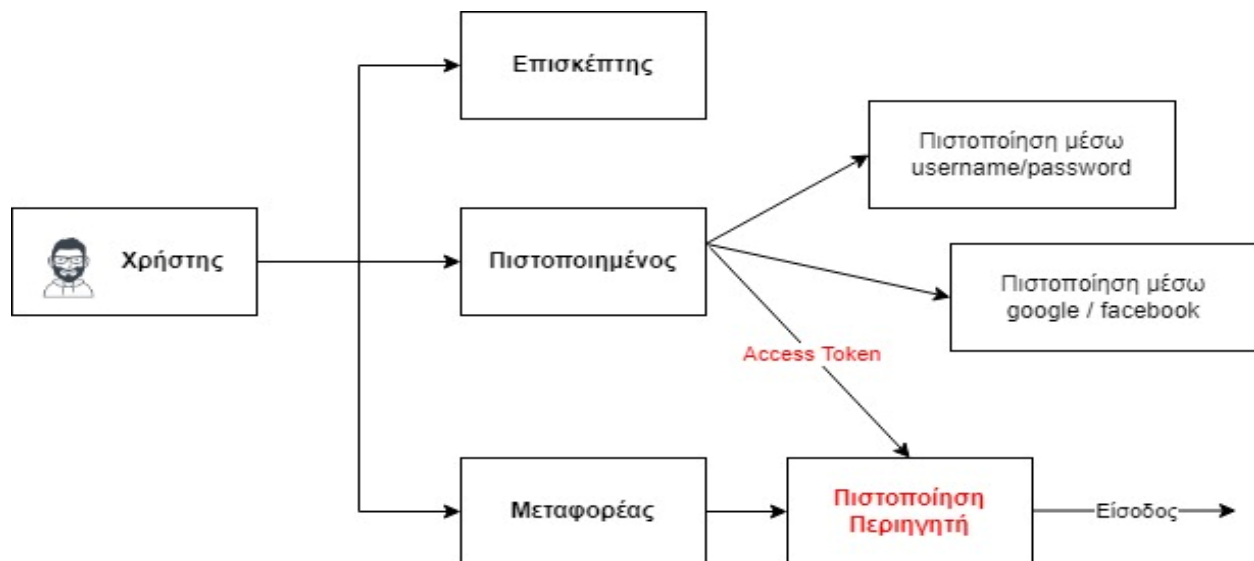
-- Εμφάνιση όλων των αποστολών με στοιχεία αποστολέα παραλήπτη.
SELECT sydetata.*,
       senders.lname as slname,
       senders.fname as sfname,
       receivers.fname as rfname,
       receivers.lname as rlname
FROM sydetata left join senders on senders.id=sydetata.senderid
              left join receivers on receivers.id=sydetata.receiverid
where sydetata.owner='OWNERTYPE'
order by sydetata.createtime DESC

```

4.3 Πρόσβαση χρηστών

Η εφαρμογή υποστηρίζει την πρόσβαση για τις κατηγορίες χρηστών (α) επισκέπτη, (β) εγγεγραμμένο χρήστη και (γ) τον μεταφορέα που αναλαμβάνει τη διεκπεραίωση της αποστολής στον χώρο του παραλήπτη. Ανάλογα με την ιδιότητα κάθε χρήστη υπάρχουν και οι αντίστοιχες ενέργειες που μπορεί να κάνει. Η πρόσβαση στην εφαρμογή με κωδικούς χρήσης μπορεί να γίνει με πιστοποίηση της ίδιας της εφαρμογής ή μέσω άλλων ιστοτόπων, όπως είναι το Google και το Facebook.

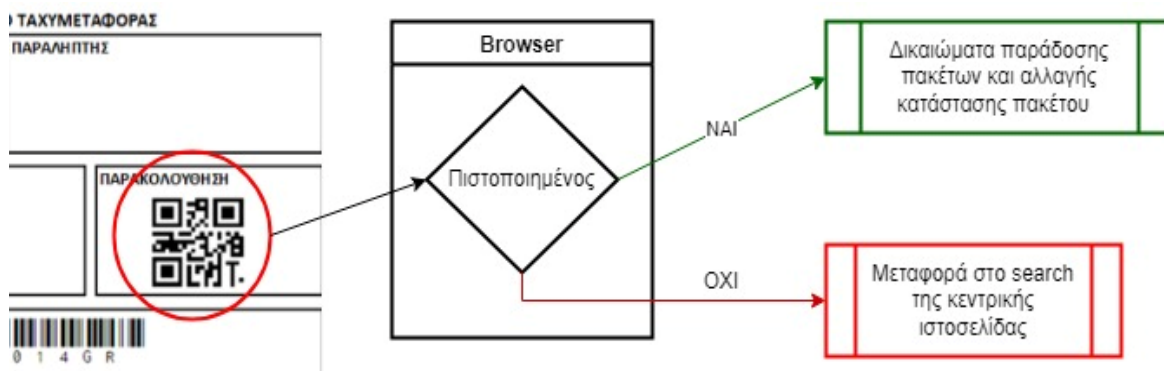
Ο πρώτος πιστοποιημένος χρήστης που θα δημιουργηθεί κατά την εγκατάσταση της εφαρμογής θα χαρακτηριστεί ως super user από το σύστημα και θα του παρέχονται δυνατότητες καθοριστικές για την λειτουργία της εφαρμογής, όπως η αρχικοποίηση μεταβλητών του συστήματος.



Εικόνα 11 Είσοδος χρηστών

Μία ιδιαίτερη περίπτωση χρήστη είναι ο διεκπεραιωτής της αποστολής ο οποίος θα έχει πρόσβαση στο τμήμα της εφαρμογής που αφορά την παράδοση του πακέτου στον παραλήπτη και η πρόσβαση αυτή δεν θα γίνεται με κωδικό (username/password) όπως στους υπόλοιπους χρήστες, αλλά μέσω μοναδικού αλφαριθμητικού (access token) με το οποίο ο χρήστης θα πιστοποιεί τον περιηγητή του για την διεκπεραίωση της εργασίας παράδοσης. Το access token θα παράγεται από την εφαρμογή ως ζευγάρι token-password και θα αποδίδεται στον χρήστη. Όταν ο χρήστης σαρώσει το QR code του δελτίου αποστολής του πακέτου θα μεταφέρεται αυτόματα σε σελίδα της εφαρμογής, στην οποία θα μπορεί να αλλαχθεί η κατάσταση του δέματος (π.χ. από «προς παράδοση» σε «παραδόθηκε») εφόσον έχει γίνει πιστοποίηση του περιηγητή με χρήση του ζευγαριού token-password. Ο περιηγητής θα διατηρεί τα στοιχεία πιστοποίησης μέσω cookies στην πλευρά του χρήστη ώστε κάθε φορά να μπορεί να συνδέεται αυτόματα. Τα στοιχεία θα μπορούν να διαγραφούν είτε από τον ίδιο τον browser με καθαρισμό ή από επιλογή που θα έχει ο χρήστης στο μενού των επιλογών του. Σε περίπτωση που δεν υλοποιηθεί η πιστοποίηση του χρήστη τότε αυτόματα η εφαρμογή θα ανακατευθύνεται στην κεντρική σελίδα στο τμήμα αναζήτησης πακέτων.

Με αυτό τον τρόπο θα μπορούσαν να δημιουργηθούν χρήστες με περιορισμένα δικαιώματα ώστε να διεκπεραιώνουν συγκεκριμένες εργασίες.



Εικόνα 12 Πρόσβαση μεταφορέα με πιστοποίηση περιηγητή

Συγκεντρωτικά οι εργασίες που θα μπορεί να κάνει κάθε χρήστης φαίνονται στον παρακάτω πίνακα:

Χρήστης	Δικαιώματα / εργασίες
Super User	<ul style="list-style-type: none"> • Έχει όλα τα δικαιώματα του administrator • Ορισμός κλειδιών google και facebook • Καθορισμός παραμέτρων πρόσβασης σε email server ώστε να μπορεί να λειτουργήσει η αποστολή μηνυμάτων από την ιστοσελίδα. • Ανακατασκευή πληροφοριακών στοιχείων προερχόμενων από το 17Track.net

Administrator	<ul style="list-style-type: none"> • Διαχείριση καταστάσεων δεμάτων. • Διαχείριση μηνυμάτων ιστοσελίδας • Διαχείριση αποστολών • Διαχείριση παραληπτών • Δημιουργία , καταχώριση , εκτύπωση και επανεκτύπωση ΣΥΔΕΤΑ. • Καταχώριση και παρακολούθηση αποστολής από τρίτη ταχυμεταφορική. • Αλλαγή κατάστασης δέματος, λειτουργία πανομοιότυπη με αυτή του απλού μεταφορέα.
Μεταφορέας	<ul style="list-style-type: none"> • Αλλαγή κατάστασης μίας αποστολής μέσω εξειδικευμένης εφαρμογής και πιστοποίησης του browser με API κλειδιά.
Επισκέπτης	<ul style="list-style-type: none"> • Περιήγηση στον ιστότοπο. • Αναζήτηση δεμάτων που έχουν ανατεθεί αποκλειστικά για μεταφορά από τον παρόν ιστότοπο • Αποστολή μηνύματος επικοινωνίας με τον διαχειριστή • Εγγραφή και δημιουργία λογαριασμού ως πιστοποιημένο μέλος • Ανάκτηση κωδικών πρόσβασης στον λογαριασμό του ως πιστοποιημένο μέλος

Πίνακας 23 Δικαιώματα χρηστών ανά περίπτωση

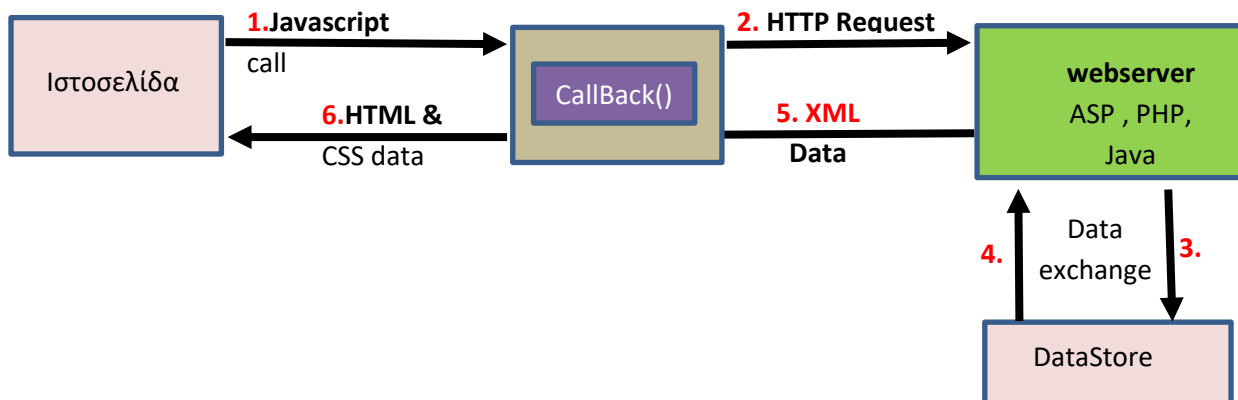
5 Τεχνικές και τεχνολογίες προγραμματισμού

Σε αυτό το κεφάλαιο θα ακολουθήσει περιγραφή των βασικών τεχνικών και τεχνολογιών που εφαρμόστηκαν στην κατασκευή της εφαρμογής. Θα εξετάσουμε τον τρόπο επικοινωνίας της ιστοσελίδας με τον server, την επικοινωνία της εφαρμογής με τρίτους διακομιστές για την αποστολή και παραλαβή πληροφοριών, και τη δυνατότητα πρόσβασης πιστοποιημένου χρήστη μέσω λογαριασμών που διατηρούν σε τρίτες ιστοσελίδες όπως Google και Facebook.

5.1 Asynchronous JavaScript and XML (AJAX)

Όπως έχουμε δει προηγουμένως ο περιηγητής αποτιμά σειριακά τη Γλώσσα Μορφοποίησης Υπερκειμένου (HTML) και το αποτέλεσμα αυτής της αποτίμησης διαμορφώνεται και παρουσιάζεται στους χρήστες. Μέχρι την αρχή της δεκαετίας του 2000, προκειμένου να ζητηθούν εκ νέου δεδομένα από τον server και αυτά να αναπαρασταθούν στη σελίδα, έπρεπε να γίνει επαναφόρτωση της ιστοσελίδας από την αρχή. Έτσι όταν θέλαμε να ανανεώσουμε το περιεχόμενο ενός τμήματος της ιστοσελίδας και μόνο αυτό ή όταν έπρεπε να στείλουμε δεδομένα στον server έπρεπε ή να καλέσουμε πάλι την ιστοσελίδα ή να κάνουμε χρήση του `iframe tag`. Αυτό άλλαξε από το 2000 μέσω του αντικειμένου XMLHttpRequest, η χρήση του οποίου έγινε ευρύτερα γνωστή το 2005. Για να μπορέσουμε λοιπόν να κάνουμε χρήση του αντικειμένου XMLHttpRequest υπάρχει συνεργασία ανάμεσα σε υπάρχουσες τεχνολογίες το σύνολο των οποίων αποτέλεσε την τεχνολογία AJAX - Asynchronous JavaScript and XML.

Η τεχνολογία αυτή στηρίζεται στο αντικείμενο XMLHttpRequest, μέσω του οποίου έχουμε την δυνατότητα να κάνουμε αποστολή αιτήματος μέσω διαδικτύου σε κάποιο απομακρυσμένο server. Αυτό υλοποιείται με τη βοήθεια της JavaScript, χωρίς να απαιτείται επαναφόρτωση της ιστοσελίδας. Όταν ο server απαντήσει στο παραπάνω ερώτημα, τότε ενεργοποιείται εκ νέου το αντικείμενο XMLHttpRequest για την υποδοχή των δεδομένων, τα οποία με την βοήθεια της JavaScript και του μοντέλου αντικειμένων DOM, τοποθετούνται στις θέσεις που πρέπει, χωρίς να χρειαστεί να γίνει επαναφόρτωση. Η παραπάνω λειτουργία απεικονίζεται σε σχεδιάγραμμα ως φαίνεται πιο κάτω.



Εικόνα 13 Αλληλουχία γεγονότων στην υλοποίηση με Ajax

Οι βασικές μέθοδοι και ιδιότητες του αντικειμένου XMLHttpRequest καταγράφονται στον παρακάτω πίνακα.

Μέθοδοι αντικειμένου	Περιγραφή-σχόλια
<i>new XMLHttpRequest()</i>	Δημιουργείται ένα νέο αντικείμενο
<i>abort()</i>	Ακύρωση αιτήματος
<i>setRequestHeader(header,value)</i>	Προσθέτει κεφαλίδες HTTP (ετικέτας / τιμή) στην κεφαλίδα της αίτησης που θα αποσταλεί
<i>getResponseHeader(headerName)</i>	Επιστρέφει συγκεκριμένη κεφαλίδα από την απάντηση που απέστειλε ο server
<i>getAllResponseHeaders()</i>	Επιστρέφει όλες τις κεφαλίδες που απέστειλε στην απάντησή του ο server
<i>open (Method, Url, Async, User, Psw)</i>	Καθορίζει και διαμορφώνει το αίτημα που θα σταλεί. Method: Είναι η μέθοδος αποστολής που μπορεί να είναι <i>GET</i> ή <i>POST</i> . Παρ' όλο που το <i>GET</i> είναι απλούστερη μέθοδος, η <i>POST</i> είναι πιο ασφαλής και επιτρέπει την αποστολή μεγάλου όγκου δεδομένων προς τον server. Ο συνδυασμός των δύο παραπάνω ενδείκνυται για την μεγιστοποίηση ασφάλειας, ταχύτητας και ευχρηστίας. url: Ο καθολικός εντοπιστής πόρων που υποδεικνύει το πρωτόκολλο (π.χ. http/https) τον server, ενδεχομένως τη θύρα TCP/IP, τη διαδρομή της αίτησης και ενδεχομένως παραμέτρους. Async: Με <i>true</i> έχουμε ασύγχρονη επικοινωνία με τον server και με <i>false</i> έχουμε σύγχρονη δηλαδή αναμονή για την απάντηση από τον server χωρίς να μπορούμε να κάνουμε κάτι άλλο. User , Psw: Προαιρετικά όνομα χρήστη και κωδικός πρόσβασης.
<i>send(<stringData>)</i>	Η <i>send()</i> αποστέλλει τη διαμορφωμένη αίτηση στον server. Αν θέλουμε να στείλουμε δεδομένα με την μέθοδο <i>POST</i> τότε μπορούμε να χρησιμοποιήσουμε την προαιρετική παράμετρο <i>stringData</i> η οποία θα συμπεριλάβει τα δεδομένα προς αποστολή.

Πίνακας 24 Βασικές μέθοδοι του αντικειμένου XMLHttpRequest

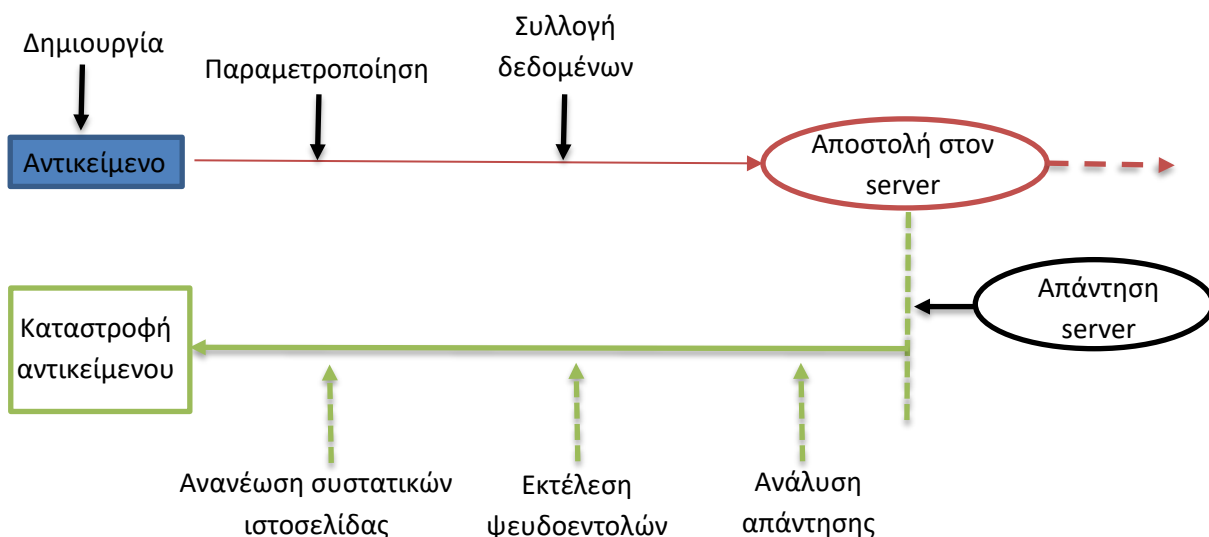
Ιδιότητες αντικειμένου	
<i>readyState</i>	Επιστρέφει την κατάσταση του αιτήματος. Οι τιμές που μπορεί να πάρει είναι: <ul style="list-style-type: none"> • 0: το αίτημα δεν έχει αρχικοποιηθεί • 1: έχει δημιουργηθεί σύνδεση προς τον server • 2: το αίτημα ελήφθη • 3: το αίτημα βρίσκεται σε επεξεργασία • 4: το αίτημα ολοκληρώθηκε
<i>onreadystatechange</i>	Ορίζουμε μία συνάρτηση η οποία θα εκτελείται κάθε φορά που η ιδιότητα <i>readyState</i> θα αλλάζει κατάσταση.

status	Επιστρέφει κωδικό αποτελέσματος της επεξεργασίας της αίτησης (μόνο τον κωδικό και όχι την περιγραφή). Οι συνηθέστερες τιμές είναι <ul style="list-style-type: none"> • 200: "OK" • 403: "Forbidden" • 404: "Page not found"
statusText	Επιστρέφει την περιγραφή που αντιστοιχεί στον κωδικό αποτελέσματος.
responseXML	Επιστρέφει τα δεδομένα που έχουν σταλεί από τον server ως απάντηση, με την μορφή XML.
responseText	Επιστρέφει τα δεδομένα που έχουν σταλεί από τον server ως απάντηση, με την μορφή συμβολοσειράς.

Πίνακας 25 Βασικές ιδιότητες του αντικειμένου XMLHttpRequest

Η παραπάνω τεχνολογία έχει χρησιμοποιηθεί στην εφαρμογή για την κωδικοποίηση των σχετικών λειτουργιών, η οποία περιλαμβάνεται στο αρχείο dit-ajax.js. Η υλοποίηση έχει κινηθεί στα εξής στάδια:

- Δημιουργία αντικειμένου με βασικές ιδιότητες
- Συλλογή δεδομένων που πρέπει να αποσταλούν στον server.
Θέλοντας η υλοποίηση να μπορεί να προσαρμοστεί κάθε φορά στα στοιχεία της προβαλλόμενης σελίδας έχουν φτιαχτεί μία σειρά από συναρτήσεις που αλληλοεπιδρούν με το μοντέλο DOM ώστε να συλλέγουν τις πληροφορίες πριν την τελική τους αποστολή.
- Αποστολή διαμορφωμένης αίτησης στον server και υλοποίηση και υπόδειξη της συνάρτησης υποδοχής των δεδομένων από τον server.
- Επεξεργασία των δεδομένων, όταν αυτά ληφθούν με ασύγχρονο τρόπο από τον server.
Προκειμένου να γενικευτεί ο τρόπος αλληλεπίδρασης server και client έχουν υλοποιηθεί μέσω μίας συνάρτησης «ψευδοεντολές» που μπορούν να σταλούν από τον server και να αποτιμηθούν στον client.
- Αφού ολοκληρωθεί η ασύγχρονη επικοινωνία θα πρέπει να καταστραφεί το αντικείμενο.



Εικόνα 14 Ασύγχρονη επικοινωνία client-server

Οι συναρτήσεις που έχουν υλοποιηθεί μπορούν να ομαδοποιηθούν στα εξής:

Ομάδες συναρτήσεων υλοποίησης ασύγχρονης επικοινωνίας	
Δημιουργία και καταστροφή αντικειμένου	<ul style="list-style-type: none"> • <i>ajax_new()</i>: Δημιουργία αντικειμένου • <i>ajax_destroy()</i>: Καταστροφή αντικειμένου
Συλλογή και διαμόρφωση δεδομένων	<ul style="list-style-type: none"> • <i>ajax_form()</i>: Συλλογή όλων των στοιχείων μίας φόρμας form tag. • <i>ajax_field()</i>: Συλλογή δεδομένων από συγκεκριμένο tag. • <i>ajax_post_as_field()</i>: Διαμόρφωση τιμής ως τιμή tag (κλειδί=τιμή). • <i>ajax_dataset_post_as_field()</i>: Διαμόρφωση τιμών από το dataset ως τιμή tag (κλειδί=τιμή).
Αποστολή αιτήματος	<ul style="list-style-type: none"> • <i>ajax_post()</i>: αποστολή αιτήματος στον server, ενημέρωση χρήστη ότι πραγματοποιείται επικοινωνία, και καθορισμός συνάρτησης η οποία θα εκτελεστεί όταν υπάρξει απάντηση (onreadystatechange).
Ανάλυση απάντησης	<p><i>ajax_post_display_split()</i>: Εκτελείται μέσα από την οριζόμενη στο onreadystatechange συνάρτηση και έχει σαν σκοπό να αναλύσει την απάντηση από τον server, να εντοπίσει ψευδοεντολές που έχουν αποσταλεί και να τις εκτελέσει. Οι ψευδοεντολές αυτές είναι:</p> <ul style="list-style-type: none"> • <<changetarget=divname>> Αλλαγή του προκαθορισμένου τμήματος (div) εξόδου κειμένου που είναι το none. • <<inittarget =divname>> Καθαρισμός του τμήματος. • <<javascript=jscod>> Εκτελεί κώδικα JavaScript που έχει ζητηθεί από τον server. • <<RunAfterjavascript =jscod>> Εκτελεί κώδικα JavaScript που έχει ζητηθεί από τον server στο τέλος της ανάλυσης ως τελευταία εντολή πριν την καταστροφή του αντικειμένου. • <<set tagid=value>> Τοποθετεί την τιμή value στο tagid.value της ιστοσελίδας.

Πίνακας 26 Ομάδες συναρτήσεων υλοποίησης ασύγχρονης επικοινωνίας

5.2 JavaScript Object Notation (JSON), αποστολή και λήψη μέσω cURL

Το JSON³ είναι ένα πρότυπο ανταλλαγής δεδομένων μεταξύ διαφορετικών πλατφορμών. Χρησιμοποιείται στην ανταλλαγή δεδομένων μεταξύ περιηγητή και διακομιστή μέσω της Ajax και στην επικοινωνία μεταξύ διακομιστών για την ανάπτυξη εφαρμογών, στο πλαίσιο της υλοποίησης των RESTful API. Οι παραπάνω χρήσεις καθιστούν το JSON ως ένα ευρέως αποδεκτό πρότυπο κειμένου. Το πρότυπο JSON είναι ανεξάρτητο από γλώσσες προγραμματισμού, παρόλο που χρησιμοποιεί πρακτικές που συναντώνται σε γλώσσες προγραμματισμού (όπως η C). Τόσο η JavaScript όσο και η PHP έχουν την δυνατότητα να μετατρέπουν αντικείμενα JSON σε συμβολοσειρές με μεγάλη ευκολία. Ένα αντικείμενο θα έχει τις εξής μορφές:

³ <https://www.json.org/>

- {"όνομα κλειδιού": "τιμή" ,}είναι ζευγάρια κλειδιού και τιμής τα οποία διαχωρίζονται μεταξύ τους από το κόμμα. Η τιμή μπορεί να είναι:
 - Απλή τιμή, που θα μπορούσε να αναπαριστά συμβολοσειρά, αριθμό, λογική τιμή (true ή false), ή την «κενή τιμή» null.
 - Αντικείμενο με τη μορφή {"όνομα κλειδιού": "τιμή" ,}
 - πίνακας, που έχει τη μορφή ["τιμή1" , "τιμή2".....]. Κάθε τιμή μπορεί να είναι με τη σειρά της απλή τιμή, αντικείμενο ή πίνακας.
- Ένας πίνακας με την μορφή JavaScript πίνακα και ο οποίος μετατρέπεται σε συμβολοσειρά μέσω της συνάρτησης stringify().

Η απλή δομή στη γραφή αλλά και τη χρήση, ενός αντικειμένου JSON, έχει ως συνέπεια τον μικρό όγκο δεδομένων και την αυξημένη ταχύτητα μεταφοράς και επεξεργασίας. Παράλληλα έχουμε τη δυνατότητα να χρησιμοποιούμε πίνακες αλλά και άλλους τύπους δεδομένων, όπως ημερομηνίες χωρίς να χρειάζεται να υπάρχει κάποιος αναλυτής (parser) για την μετατροπή του κειμένου σε αντικείμενο, αφού αυτή την λειτουργία την έχει ενσωματωμένη η JavaScript.

Στην εφαρμογή που αναπτύχθηκε, η αναπαράσταση JSON χρησιμοποιήθηκε στην κατασκευή της διεπαφής μέσω της οποίας θα γίνεται η διαβίβαση και λήψη δεδομένων από την ίδια την εφαρμογή αλλά και από τρίτους χρήστες. Η διεπαφή αυτή ακολουθεί το πρότυπο REST API. επίσης, η αναπαράσταση JSON χρησιμοποιείται για τη μεταφορά και λήψη δεδομένων από τρίτους διακομιστές οι οποίοι μας παρέχουν πληροφορίες που χρειαζόμαστε. Συγκεκριμένα:

- Στη προγραμματιστική διεπαφή (API) του προγράμματος, το αρχείο .php έχει τα εξής απαραίτητα συστατικά:


```
header("Content-Type: application/json; ");
```

 Με αυτή την κεφαλίδα ενημερώνουμε τον παραλήπτη, ότι το κείμενο απάντησης θα είναι σε μορφή JSON.

Επιστροφή πίνακα: Κατά τη δόμηση της απάντησης τα στοιχεία που θα σταλούν στον παραλήπτη καταχωρούνται σε πίνακα, ο οποίος λόγω της κεφαλίδας του αρχείου θα επιστραφεί με την μορφή JSON.

```
$returnjson=Array();
$returnjson['RETURN']=$cresult;
$returnjson['MESSAGE']=$cmessgae;
return $returnjson;
```
- Στην επικοινωνία της εφαρμογής με την API διεπαφή τρίτου ιστότοπου. Η αποστολή και η λήψη δεδομένων γίνεται μέσω JSON χρησιμοποιώντας την βιβλιοθήκη cURL της PHP.

Όπως είδαμε πιο πάνω, η εφαρμογή που αναπτύχθηκε ανταλλάσσει δεδομένα με τρίτους διακομιστές. Αυτό επιτυγχάνεται με χρήση της βιβλιοθήκης libcurl, που είναι υλοποίηση του cURL (συντομογραφία του *Client for URLs*). Η μεγάλη δημοφιλία που έχει ανάμεσα στους προγραμματιστές οφείλεται στην ευκολία της (α) να ανταλλάσσει δεδομένα μεταξύ διακομιστών υποστηρίζοντας πρωτόκολλα FILE, FTP, GOPHER, HTTP, HTTPS, IMAP, LDAP, MQTT, POP3, RTMP, SMTP, SMTPS, TELNET και άλλα, (β) η επικοινωνία να βασιστεί σε μεθόδους GET, POST, PUT, HEAD, (γ) υποστηρίζοντας proxies,

συνδέσεις πάνω από SSL, cookies και τον έλεγχο ταυτότητας με όνομα χρήστη. Η cURL συνεργάζεται με την PHP ως επέκτασή της με τρόπο λειτουργίας που περιγράφεται από τα εξής βήματα:

1. *Αρχικοποίηση της cURL:* Αυτό γίνεται με χρήση της `curl_init()` για την δημιουργία αιτήματος επικοινωνίας (session).
2. *Συλλογή ιδιοτήτων και χαρακτηριστικών:* Ο καθορισμός των παραμέτρων μπορεί να γίνει μέσω της `curl_setopt()` με την οποία ρυθμίζουμε μία παράμετρο ανά κλήση ή μέσω της `curl_setopt_array()` με την οποία ρυθμίζουμε πολλές παραμέτρους ταυτόχρονα με χρήση πίνακα. Ο κατάλογος των παραμέτρων που μπορούμε να ρυθμίσουμε είναι εκτενής και εδώ θα αναφέρουμε αυτές που εμείς χρησιμοποιήθηκαν στην εφαρμογή. Για πλήρη κατάλογο των παραμέτρων και περισσότερες πληροφορίες, ο ενδιαφερόμενος αναγνώστης παραπέμπεται στο <https://www.php.net/manual/en/book.curl.php>.

Βασικές παράμετροι <code>curl_setopt()</code>	
CURLOPT_URL	Το URL στο οποίο απευθύνεται το αίτημα.
CURLOPT_RETURNTRANSFER	Ρυθμίζουμε αν η απάντηση του αιτήματος θα είναι με την μορφή συμβολοσειράς (true, σε λειτουργία σίγασης) ή θα εμφανιστεί στο πρόγραμμα περιήγησης.
CURLOPT_ENCODING	Δείχνει την κωδικοποίηση της κεφαλίδας Accept-Encoding. Οι πιθανές κωδικοποιήσεις είναι identity, deflate και gzip. Όταν είναι κενό τότε τις δέχεται όλες τις κωδικοποιήσεις.
CURLOPT_MAXREDIRS	Ο μέγιστος αριθμός ανακατευθύνσεων HTTP που επιτρέπεται να συμβούν κατά την εκτέλεση του αιτήματος στον απομακρυσμένο διακομιστή.
CURLOPT_TIMEOUT	Ο μέγιστος αριθμός δευτερολέπτων που επιτρέπει η PHP στη <code>curl_exec()</code> να εκτελεστεί.
CURLOPT_FOLLOWLOCATION	Επιτρέπει ή όχι, την ανακατεύθυνση μέσω της κεφαλίδας Location.
CURLOPT_HTTP_VERSION	Ορίζεται η έκδοση του πρωτοκόλλου HTTP που μπορεί να χρησιμοποιήσει η curl.
CURLOPT_POST	Το αίτημα θα εκτελεστεί με τη μέθοδο POST.
CURLOPT_POSTFIELDS	Τα δεδομένα προς αποστολή όταν χρησιμοποιείται η μέθοδος POST.
CURLOPT_HTTPHEADER	Αποστέλλουμε έναν πίνακα, τα στοιχεία του οποίου είναι κεφαλίδες που θέλουμε να ενεργοποιήσουμε και να ορίσουμε τις τιμές τους.

Πίνακας 27 Βασικές παράμετροι `curl_setopt()`

3. *Εκτέλεση αιτήματος:* Το αίτημα που έχουμε δημιουργήσει και παραμετροποιήσει, εκτελείται μέσω της `curl_exec()`, η οποία όταν ολοκληρωθεί θα μας επιστρέψει την απάντηση από τον τρίτο διακομιστή (που προσδιορίζεται εντός του URL) με την μορφή συμβολοσειράς JSON. Αυτή η συμβολοσειρά θα μετατραπεί εντός της PHP σε αντικείμενο JSON ή πίνακα τιμών μέσω της `json_decode()`.
4. *Κλείσιμο και καταστροφή του αιτήματος:* Ο τερματισμός του αιτήματος επιτυγχάνεται μέσω της `curl_close()`.

Η PHP χειρίζεται τα δεδομένα που στέλνει στο αίτημα και τα αποτελέσματα της απάντησης μέσω κωδικοποίησης και αποκωδικοποίησης JSON. Αυτό επιτυγχάνεται με τις εξής εντολές:

- `json_decode (string $json , bool $assoc)`
Μετατρέπει ένα JSON string (\$json) σε αναπαράσταση αντικειμένων της PHP. Όταν η \$assoc είναι false τότε επιστρέφει την τιμή ως αντικείμενο, ενώ όταν είναι true επιστρέφει την τιμή με τη μορφή συσχετιστικού πίνακα.
- `json_encode ($value)`
Επιστρέφει ένα string που περιέχει την τιμή της παραμέτρου \$value σε μορφή JSON.

Στην εφαρμογή μας, η υλοποίηση της επικοινωνίας με τον διακομιστή 17track.net για την αναζήτηση πληροφοριών σχετικά με κατάσταση συγκεκριμένης αποστολής υλοποιείται μέσω του παρακάτω κώδικα.

```
<?php
$curl = curl_init();
$data = array(0 => array("number" => "3123751433",
                        "carrier" => "100070",
                        "auto_detection" => false
                    )
            );
$data = json_encode($data);
curl_setopt_array($curl, array(
    CURLOPT_URL => 'https://api.17track.net/track/v1/GetTrackInfo',
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => '',
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 0,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_POST => true,
    CURLOPT_POSTFIELDS => $data,
    CURLOPT_HTTPHEADER => array(
        '17token: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx',
        'Content-Type: application/json'
    )
));
$response = curl_exec($curl);
curl_close($curl);
echo $response;
?>
```

Η κλήση μέσω API, με χρήση της curl από έναν τρίτο server σε μέθοδο που έχουμε υλοποιήσει (στο παράδειγμα seek) γίνεται ως εξής:

```
<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
$curl = curl_init();
```

```

curl_setopt($curl, CURLOPT_URL,
            'https://api.dit2020cst-thesis.ga/v1/seek/');
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS,
            array('parcelnumber' => $_POST["parcelnumber"])
            );
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$output = curl_exec($curl);
curl_close($curl);
$json = json_decode($output, true);
?>

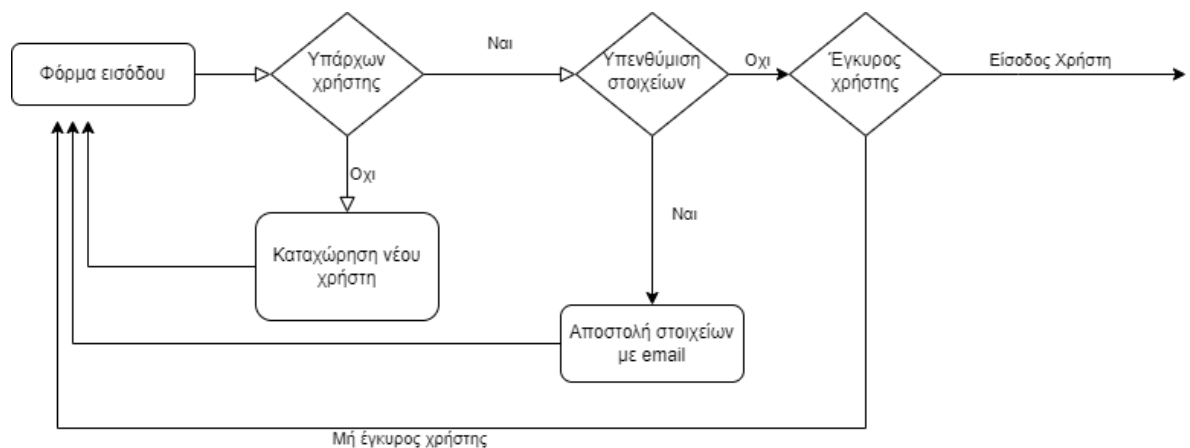
```

5.3 Διαδικασία πιστοποίησης (Authentication) χρήστη

Στην εφαρμογή υπάρχει πρόσβαση για τον μη πιστοποιημένο χρήστη, επισκέπτη, ο οποίος έχει μόνο συγκεκριμένες δυνατότητες αναζήτησης μίας αποστολής. Όταν όμως πρέπει ο χρήστης να προχωρήσει σε περισσότερες ενέργειες και να ενεργοποιήσει διαδικασίες τότε θα πρέπει να κάνει είσοδο ως εγγεγραμμένος χρήστης. Για την είσοδο του χρήστη έχουν προβλεφθεί τρεις τρόποι:

- *Είσοδος με εγγραφή χρήστη και διαχείριση των στοιχείων του από την ίδια την εφαρμογή.*

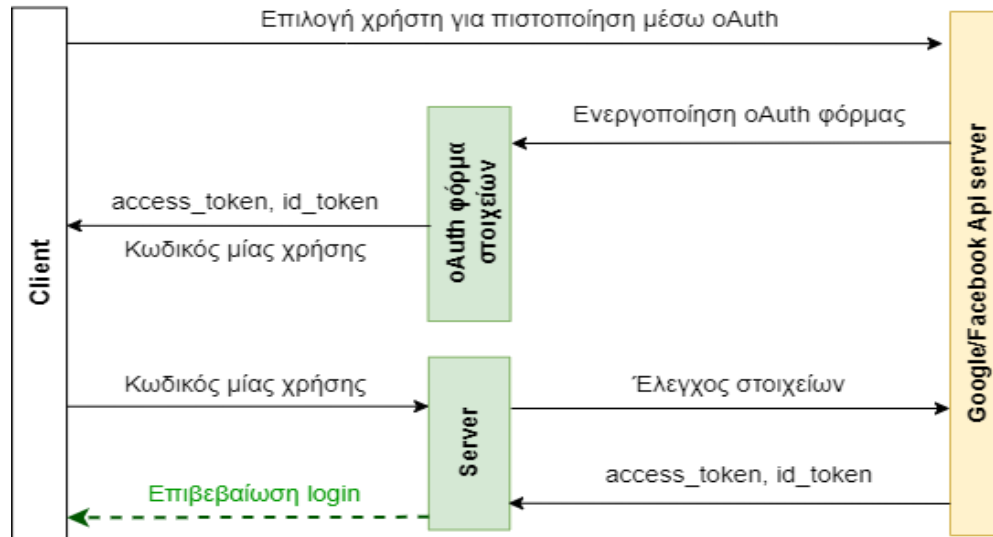
Σε αυτή την περίπτωση θα πρέπει η μεταφορά των δεδομένων από τον client στον server να γίνεται μέσω πρωτοκόλλου HTTPS (δηλαδή με κρυπτογράφηση), η αποθήκευση των κωδικών πρόσβασης στην βάση δεδομένων να γίνεται με κρυπτογράφηση (AES-256-CBC) και τέλος οι κωδικοί πρόσβασης δεν καταχωρούνται σε καμία μορφή (cookies ή οτιδήποτε άλλο) στην πλευρά του client. Το διάγραμμα χειρισμού του χρήστη είναι:



Εικόνα 15 Έλεγχος και είσοδος χρήστη ως μέλος.

- *Είσοδο με χρήση OAuth2 credentials.*

Η δεύτερη επιλογή εισόδου ενός χρήστη στο σύστημα είναι μέσω πιστοποίησης OAuth^[14] ενός τρίτου παρόχου όπως είναι το Google και το Facebook και οι οποίοι θεωρούνται ασφαλείς για την αποθήκευση ευαίσθητων πληροφοριών εισόδου. Ο τρόπος λειτουργίας και για τους δύο παρόχους είναι παρόμοιος και φαίνεται στο παρακάτω διάγραμμα.

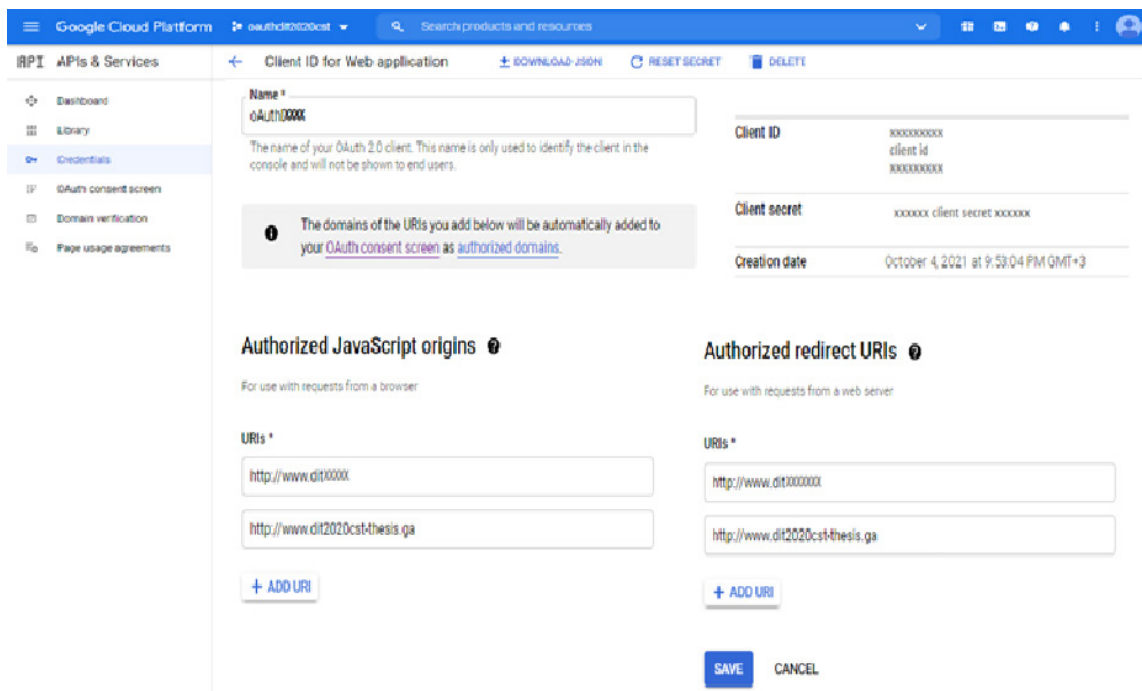
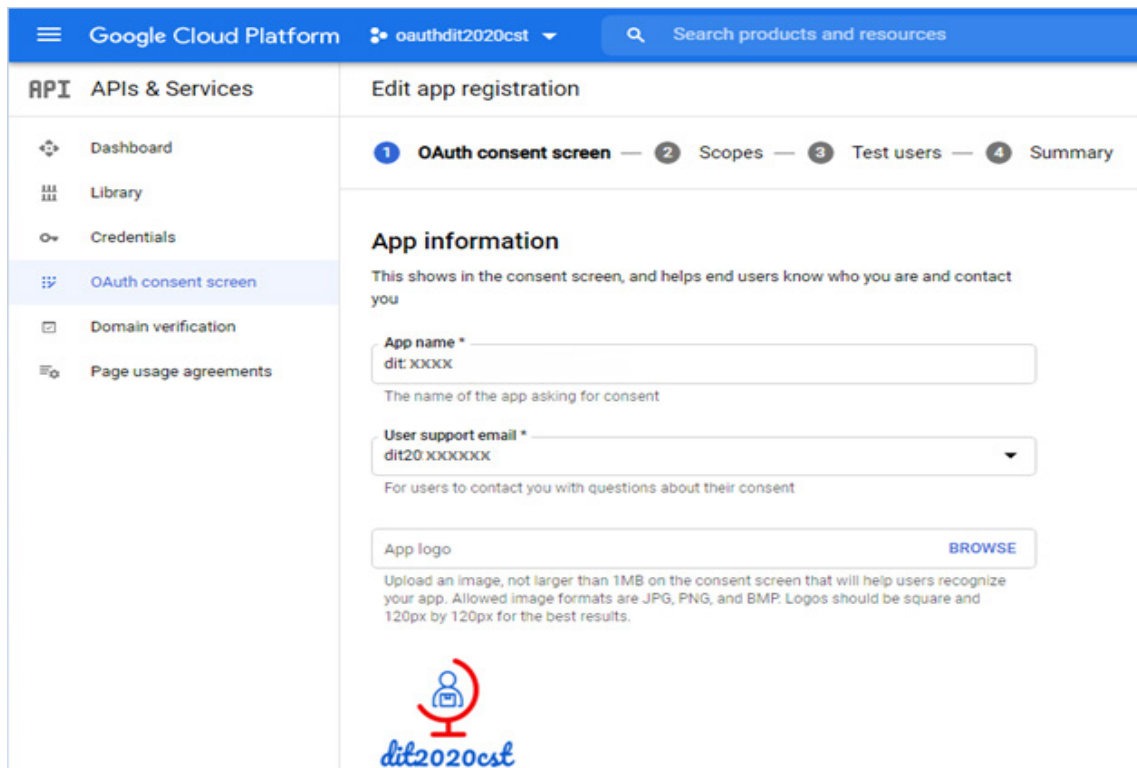


Εικόνα 16 Είσοδο με χρήση OAuth2 credentials.

Ο τρόπος αυτός εισόδου είναι αρκετά φιλικός προς τον χρήστη αφού του παρέχει την δυνατότητα να διατηρεί έναν κωδικό σε κάποιον πάροχο και με αυτόν να εισέρχεται στην περιοχή μελών σε πολλές ιστοσελίδες και εφαρμογές. Σε κάθε μία περίπτωση, θα χρησιμοποιηθεί η αντίστοιχη βιβλιοθήκη σε PHP ή JavaScript που προτείνει και τυπικά διαθέτει ο πάροχος, αλλά πριν από αυτό θα πρέπει σε κάθε έναν από τους παρόχους (Facebook και Google) να ορίσουμε μία εφαρμογή πρόσβασης στον μηχανισμό πιστοποίησης και να μας αποδοθεί το αντίστοιχο κλειδί χρήσης.

- Στο Google, η διαδικασία περιγράφεται στο <https://developers.google.com/adwords/api/docs/guides/authentication> και είναι η εξής:
 1. Άνοιγμα της σελίδας Google API Console Credentials
 2. Επιλογή υπάρχοντος project ή δημιουργία ενός νέου.
 3. Επιλογή Web application στην περιοχή Application type.
 4. Επιλογή στη σελίδα Credentials → Create credentials → OAuth client ID. Τα αναγνωριστικά αυτά τα σημειώνουμε, καθώς είναι απαραίτητα για την κλήση της υπηρεσίας από την PHP, προκειμένου να διασφαλιστεί ότι η εφαρμογή καλείται από εξουσιοδοτημένο χρήστη.





Εικόνα 17 Δημιουργία διαπιστευτηρίων OAuth2 στην πλατφόρμα της Google

Η προσαρμογή στην εφαρμογή θα γίνει μέσω της Google APIs Client v. 2.4.0 βιβλιοθήκης για PHP.

oAuth_Google.php

```
<?php
//Βασική παραμετροποίηση
//Include Google Client Library for PHP autoload file
require_once '/oAuth_Google/google-api-php-client-2.4.0/vendor/
    autoload.php';
//Make object of Google API Client for call Google API
$google_client = new Google_Client();
//Set the OAuth 2.0 Client ID
$google_client->setClientId("999999-xxxx.apps.googleusercontent.com");
//Set the OAuth 2.0 Client Secret key
$google_client->setClientSecret"Xyz-Xyz-Xyz");
//Set the OAuth 2.0 Redirect URI
$google_client->setRedirectUri('http://www.dit2020cst-thesis.ga');
//
$google_client->addScope('email');
$google_client->addScope('profile');
//start session on web page
if (session_status() == PHP_SESSION_NONE){session_start();}
$GOOGLE_login_link = '';
$GOOGLE_login_link = $google_client->createAuthUrl();
echo "<<javascript=window.location.href='". $GOOGLE_login_link. "'>>";
?>
```

index.php

```
<?php
if(isset($_GET['code']))
{
    require_once('./oAuth_Google.php');
    $GOOGLE_login_link = '';
    $token = $google_client-
>fetchAccessTokenWithAuthCode($_GET["code"]);
    if(!isset($token['error']))
    {
        $_SESSION['access_token'] = $token['access_token'];
    }
    if(isset($_SESSION['access_token']))
    {
        $google_client->setAccessToken($_SESSION['access_token']);
        $google_service = new Google_Service_Oauth2($google_client);
        $data = $google_service->userinfo->get();
    }
}
?>
```

- Κατά αντίστοιχο τρόπο πραγματοποιούμε και την παραμετροποίηση στο Facebook δημιουργώντας μια εφαρμογή “Facebook Login” εντός του Meta for Developers, εκτελώντας τις

απαραίτητες παραμετροποιήσεις ώστε να ανταποκρίνεται στην εφαρμογή, όπως φαίνεται και από τις παρακάτω εικόνες.

The screenshot shows the 'Required actions' section of the application dashboard. At the top, it displays the application name 'oAuth-dit2020cst-the...', App ID '402477554595944', App type 'Consumer', and App Mode 'Development' with a 'Live' toggle. The left sidebar contains navigation options: Dashboard, Settings, Roles, Alerts, App Review, Products (with 'Add Product' link), Facebook Login, Activity Log, and Activity Log. The main content area is titled 'Required actions' and contains a message: 'You don't have any required action items to display. If any of your apps need immediate attention in the future, an item will show here.' Below this, there are two panels: 'Application Rate Limit' and 'User Rate Limit'. The 'Application Rate Limit' panel shows the application icon and name, '0% of limit used', and a progress bar indicating '100% Remaining'. The 'User Rate Limit' panel shows '0 Users throttled'.

The screenshot shows the 'Client OAuth settings' configuration page. The top navigation bar is identical to the previous screenshot. The left sidebar is also identical, but 'Facebook Login' is expanded to show 'Settings', 'Quickstart', and 'Activity Log'. The main content area is titled 'Client OAuth settings' and contains several configuration options: 'Client OAuth login' (Yes), 'Web OAuth login' (Yes), 'Enforce HTTPS' (Yes), 'Force Web OAuth reauthentication' (No), 'Embedded Browser OAuth Login' (No), 'Use Strict Mode for redirect URIs' (Yes), 'Valid OAuth Redirect URIs' (with a text input field and a 'Copy to clipboard' button), 'Login from Devices' (No), and 'Login with the JavaScript SDK' (Yes). At the bottom, there is a section for 'Allowed Domains for the JavaScript SDK' with a text input field containing 'https://www.dit2020cst-thesis.ga/' and 'https://dit2020cst-thesis.ga/'.

oAuth-dit2020cst-the... App ID: 402477554595944 App type: Consumer App Mode: Development Live

- Dashboard
- Settings
 - Basic
 - Advanced
- Roles
- Alerts
- App Review
- Products [Add Product](#)
- Facebook Login
- Activity Log
- Activity Log

App ID
402477554595944

App secret
***** Show

Display name
oAuth-dit2020cst-thesis


Namespace

App domains
dit2020cst-thesis.ga x www.dit2020cst-thesis.ga x
www.dit2020cst-thesis.ga x

Privacy Policy URL
https://www.freeprivacypolicy.com/live/1bcaa1e3-78fa-4b1d-8fe...

Terms of Service URL
https://www.dit2020cst-thesis.ga/TermsConditions.html

Contact email
Used for important communication about your app

App icon (1024 x 1024)


App purpose
This app's primary purpose is to access and use data from Facebook's Platform on behalf of:
 Yourself or your own business
 Clients
Select this option if the primary purpose of this app is to manage data or assets on behalf of an individual client or multiple clients.

If you are developing an app that accesses and uses data from Facebook's Platform on behalf of clients, you are subject to Section 5b of the Platform Terms.

Discard Save changes

Εικόνα 18 Δημιουργία διαπιστευτηρίων OAuth2 στην πλατφόρμα Facebook

Στην περίπτωση του Facebook για να κάνουμε login θα χρησιμοποιήσουμε JavaScript, όπου αρκεί να ορίσουμε στην κεντρική σελίδα την τιμή της παραμέτρου fbAsyncInit_fbappid στο σωστό "facebook client id", έτσι ώστε όταν θα καλέσουμε την fbLogin() μέσω JavaScript να εμφανιστεί το παράθυρο εισαγωγής στοιχείων για το Facebook login. Η παράμετρος fbAsyncInit_fbappid αποτελεί τμήμα της διεπαφής της βιβλιοθήκης API του Facebook για τη γλώσσα Javascript.

```
index.php
<script src="js/scripts.js"></script>
<script >
    fbAsyncInit_fbappid=
        '<?php echo $aconfig["FACEBOOK_OAUTH"] ["CLIENTID"] ?>';
</script>
<script src="./oAuth_Facebook/oAuth_Facebook.js"></script>
```

```
system.php
// Έλεγχος επιστροφής δεδομένων από login σε facebook
if ($_POST["LoginFrom"]=="FACEBOOK")
{
    __MakeUser(array('USERNAME'=>$_POST["name"]));
    __MakeUser(array('CREATEDFROM'=>'FACEBOOK'));
    __MakeUser(array('USERID'=>'0'));
    __MakeUser(array('connectedID'=>$_POST["id"]));
    __MakeUser(array('USERPICTURE'=>$_POST["picture"]));
    __MakeUser(array('EMAIL'=>$_POST["email"]));
    $redirect=__CheckUser($_SESSION["oauth"]);
}
}
```

5.4 Τεκμηρίωση API συστήματος

Το σύστημα σχεδιάστηκε με σκοπό να μπορεί να παρέχει τις υπηρεσίες του σε τρίτους ιστότοπους παρέχοντας ένα σύνολο λειτουργιών, οι οποίες διατίθενται με τη μορφή API. Το API ακολουθεί τη λογική του REST, και για τη χρήση των παρεχόμενων λειτουργιών απαιτείται η αποστολή κατάλληλα διαμορφωμένων αιτημάτων HTTP. Στη συνέχεια του κειμένου, όπου γίνεται αναφορά σε REST resource, για να περιγράψουμε το αίτημα πρόσβασης σε έναν πόρο, θα παραλείψουμε το πρόθεμα, δηλ. το σταθερό τμήμα των διευθύνσεων στις οποίες θα αποστέλλονται τα αιτήματα. Το σταθερό αυτό τμήμα εξαρτάται από τη διεύθυνση στην οποία έχει εγκατασταθεί το σύστημα. Για την έκδοση v.1.0 στο πλαίσιο της ανάπτυξης της παρούσας πτυχιακής εργασίας, το πρόθεμα θα είναι το <https://api.dit2020cst-thesis.ga/> το οποίο αυτόματα θα κάνει αναφορά στο <https://api.dit2020cst-thesis.ga/v1/> κάνοντας χρήση του .htaccess

```
RewriteEngine on
RewriteCond %{REQUEST_URI} !^/v1
RewriteRule ^(.*)$ /v1/$1 [NC,L]
```

Το κύριο format που υποστηρίζεται για την υποβολή κλήσεων προς το API είναι το JSON. Παρακάτω δίνονται παραδείγματα της δομής του JSON που επιστρέφεται κατά την ανάκτηση των πληροφοριών ανά πόρο.

```
GET api.dit2020cst-thesis.ga/seek?XA000000385GR
{
  "OWNER": "SELF",
  "TRACKINGNUMBER": "XA000000385GR",
  "ERROR": "NO",
  "DATA": [
    {
      "createdate": "20/04/2022",
      "createtime": "7:50 PM",
      "title": "Παράδοση Παραλήπτη",
      "initial": "0",
      "ended": "1"
    },
    {
      "createdate": "20/04/2022",
      "createtime": "7:34 PM",
      "title": "Παραλαβή από πελάτη",
      "initial": "1",
      "ended": "0"
    }
  ]
}
```


6 Σενάρια Χρήσης

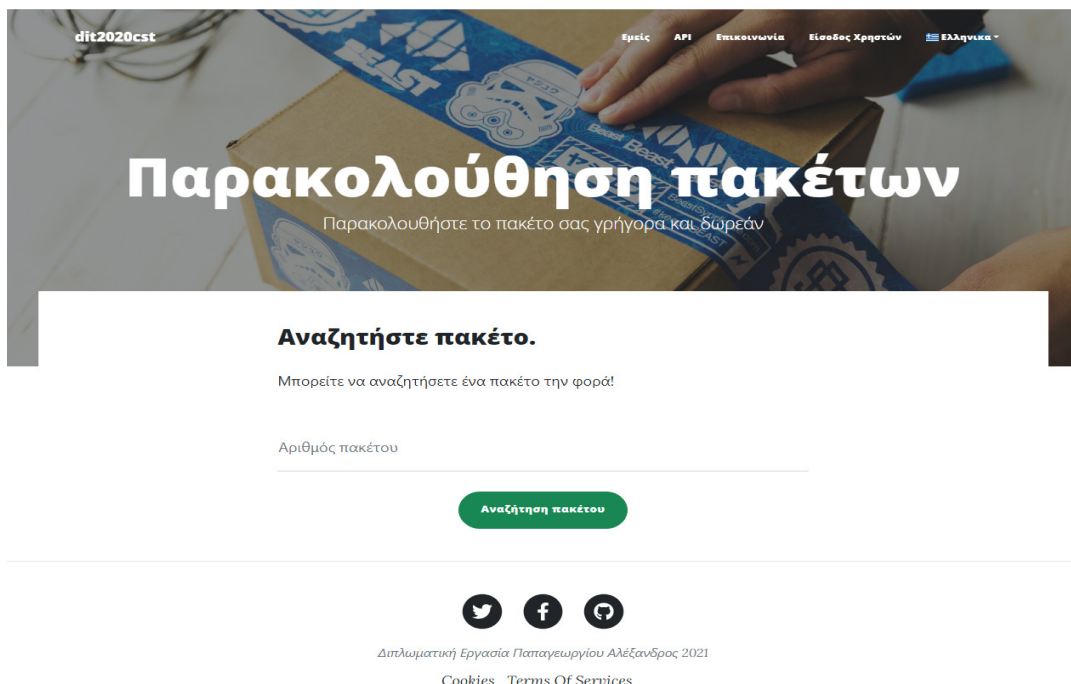
Στο κεφάλαιο αυτό θα παρουσιάσουμε αναλυτικά τα σενάρια χρήσης της ιστοσελίδας. Η ιστοσελίδα χρησιμοποιείται στο πλαίσιο τριών κυρίων σεναρίων χρήσης:

1. την περιήγηση χρήστη στην ιστοσελίδα με περιορισμένες δυνατότητες
2. την είσοδο χρήστη στον χώρο χρηστών όπου ανάλογα με την διαβάθμισή του (administrator, πιστοποιημένος χρήστης), μπορεί να εκτελέσει και διαφορετικές εργασίες και τέλος
3. τον χειρισμό της εφαρμογής από τον χρήστη που αναλαμβάνει να παραδώσει ένα δέμα και πρέπει να αλλάξει την κατάσταση του δέματος.

6.1 Περιήγηση στην ιστοσελίδα

Ο χρήστης επισκέπτης της ιστοσελίδας μπορεί να ενημερωθεί για τη θέση κάποιας αποστολής μέσω της αναζήτησης πακέτου, να επικοινωνήσει με τον διαχειριστή, να επιλέξει την γλώσσα εμφάνισης του ιστοτόπου και φυσικά να κάνει είσοδο ως πιστοποιημένος χρήστης.

- **Αρχική σελίδα:** Στην αρχική σελίδα του ιστοτόπου μπορεί να αναζητήσει κάποια αποστολή η οποία έχει διεκπεραιωθεί από τον ίδιο τον ιστότοπο. Αν ζητηθεί η παρακολούθηση πακέτου που ανήκει σε τρίτη εταιρία ταχυμεταφορικής τότε θα πρέπει να έχει καταχωρίσει τον αριθμό αποστολής ένας πιστοποιημένος χρήστης ώστε να είναι διαθέσιμο στους χρήστες επισκέπτες.



Εικόνα 19 Αρχική σελίδα περιήγησης ιστοχώρου

Παρακολούθηση πακέτων

Παρακολουθήστε το πακέτο σας γρήγορα και δωρεάν

Αναζητήστε πακέτο.

Μπορείτε να αναζητήσετε ένα πακέτο την φορά!

Αριθμός πακέτου

bc864

Αναζήτηση πακέτου

Η διαδρομή του πακέτου είναι η εξής:

Ταχυμεταφορική: ELTA

Χώρα Αφίξης: GR - Greece



Εικόνα 20 Αναζήτηση πακέτου τρίτης εταιρίας.

Παρακολούθηση πακέτων

Παρακολουθήστε το πακέτο σας γρήγορα και δωρεάν

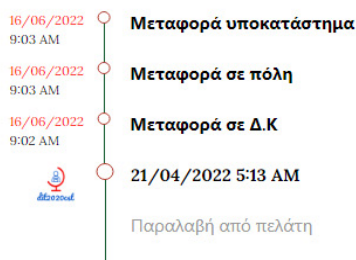
Αναζητήστε πακέτο.

Μπορείτε να αναζητήσετε ένα πακέτο την φορά!

Αριθμός πακέτου
XA000000399gr

Αναζήτηση πακέτου

Η διαδρομή του πακέτου είναι η εξής:



Εικόνα 21 Αναζήτηση πακέτου που διεκπεραιώνεται από τον ίδιο τον ιστότοπο.

Παρακολούθηση πακέτων

Παρακολουθήστε το πακέτο σας γρήγορα και δωρεάν

Αναζητήστε πακέτο.

Μπορείτε να αναζητήσετε ένα πακέτο την φορά!

Αριθμός πακέτου
XA000000399

Αναζήτηση πακέτου

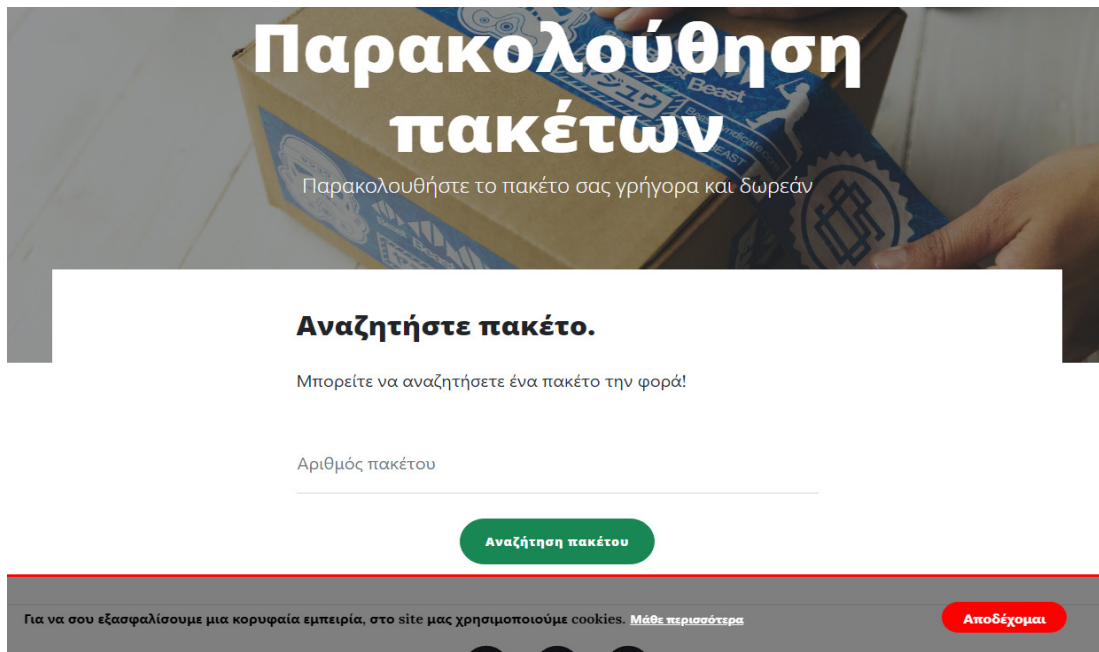
Η διαδρομή του πακέτου είναι η εξής:

Βρέθηκε λάθος συστήματος. Η διαδικασία δεν μπορεί να συνεχιστεί.

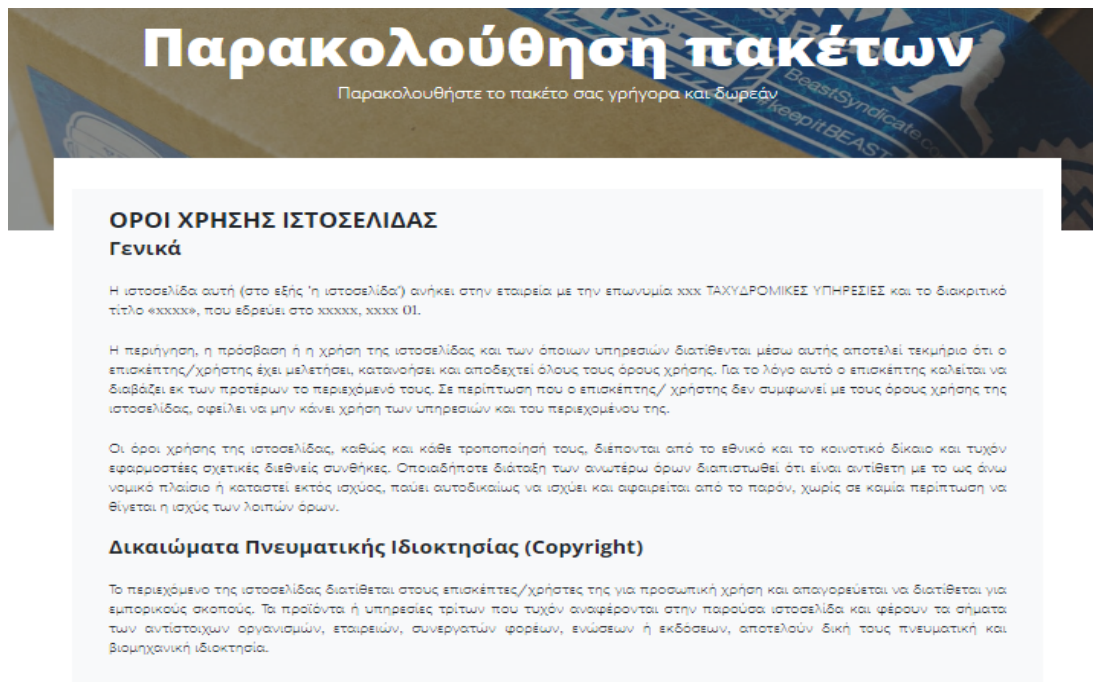


Εικόνα 22 Το πακέτο δεν βρέθηκε, εμφάνιση σχετικού λάθους.

- Cookies και Terms Of Services:** Πληροφοριακές σελίδες ο υπερσύνδεσμος των οποίων εμφανίζεται στο υποσέλιδο της σελίδας και σε αυτά μπορούμε να πάρουμε πληροφορίες σχετικά με τα cookies που χρησιμοποιούνται από τον ιστόχωρο αλλά και τους όρους χρήσης της ιστοσελίδας. Η πληροφορία σχετικά με τα cookies, είναι διαθέσιμη από την πρώτη φορά που θα επισκεφτούμε την ιστοσελίδα και μέχρι να αποδεχτούμε το κείμενο αυτό, το οποίο αποτελεί ευρωπαϊκή οδηγία σχετικά με τον σεβασμό της ιδιωτικότητας και της επεξεργασίας των προσωπικών δεδομένων, General Data Protection Regulation (GDPR).



Εικόνα 23 Ενημέρωση και αποδοχή των cookies, όπως προβλέπεται από τον General Data Protection Regulation (GDPR)



Εικόνα 24 Οροι χρήσης ιστοσελίδας, Terms Of Services.

Παρακολούθηση πακέτων

Παρακολουθήστε το πακέτο σας γρήγορα και δωρεάν

Τι είναι Cookies

Τα cookies είναι μικρά αρχεία κειμένου τα οποία αποθηκεύονται στον φυλλομετρητή μας κατά την πλοήγησή μας στο διαδίκτυο. Σκοπός τους είναι να ειδοποιούν τον ιστότοπο που επισκέπτεται ο χρήστης, για την προηγούμενη δραστηριότητά του. Συνήθως περιγράφουν στοιχεία μας όπως όνομα χρήστη (user name) και συνθηματικό πρόσβασης (password) με σκοπό κατά την επόμενη μας στον ίδιο ιστότοπο αργότερα, να μας "θυμάται" και να μην χρειάζεται να κάνουμε login.

Τα cookies μπορεί να προέρχονται από τον ιστότοπο τον οποίο έχουμε επισκεφθεί ή από κάποιον άλλον (third-party cookies), για παράδειγμα μέσω διαφημίσεων. Συνήθως είναι άκακα, έχει όμως αποδειχθεί ότι τα third-party cookies συλλέγουν πληροφορίες για τη συμπεριφορά του κάθε χρήστη στο διαδίκτυο, κάτι που εγείρει σημαντικά ερωτήματα για την ιδιωτικότητα. Αυτό ώθησε την Ε.Ε. και τις Η.Π.Α. να εκδώσουν οδηγίες για τη χρήση τους και την ενημέρωση του χρήστη, για κάθε ιστότοπο που τα χρησιμοποιεί. Υπάρχουν προγράμματα που καθαρίζουν τα κακόβουλα cookies, ενώ αν ο χρήστης επιθυμεί να τα διαγράψει δίνεται αυτή η δυνατότητα μέσα από το φυλλομετρητή ιστοσελίδων.

wikipedia.org

Η παρούσα ιστοσελίδα δεν θα μπορούσε να αποφύγει την χρήση των cookies ώστε να είναι πιο φιλική προς τον χρήστη. Τα Cookies που χρησιμοποιούνται φαίνονται στον παρακάτω πίνακα.

Πίνακας cookies

Πίνακας με τα ενεργά cookies στο παρόν site:

Όνομα Cookie	Χρήση cookie
PHPSESSID	Αυτό τα cookies χρειάζεται για να λειτουργεί σωστά ο ιστότοπος. Είναι εσωτερικό στοιχείο της PHP και επιτρέπει στις ιστοσελίδες να αποθηκεύουν δεδομένα για την κατάσταση σε σειριακή μορφή. Αυτά τα cookies λήγουν στο τέλος της περιήγησής σας στον ιστότοπό μας.

Απενεργοποίηση Cookies

Μπορείτε να ρυθμίσετε τον περιηγητή σας να μπλοκάρει ή να σας προειδοποιεί για αυτά τα cookies, αλλά κάποια μέρη της ιστοσελίδας δεν θα δουλεύουν σωστά. Ο παρακάτω πίνακας παρουσιάζει τον τρόπο απενεργοποίησης για τους βασικότερους περιηγητές. Τα βήματα ενδέχεται να παρουσιάζουν απόκλιση από την περιγραφή ανάλογα την έκδοση του περιηγητή του χρήστη.

Περιηγητής	Διαδικασία απενεργοποίησης
Internet Explorer	Επιλέξτε από το μενού 'Εργαλεία' και στη συνέχεια 'Επιλογές Internet'.

Εικόνα 25 Πληροφορίες για τα Cookies

- **Επιλογή «εμείς»:** Πληροφορίες για τα εργαλεία που χρησιμοποιήθηκαν και τους συντελεστές που εργάστηκαν για την κατασκευή του ιστοτόπου.

Παρακολούθηση πακέτων

Παρακολουθήστε το πακέτο σας γρήγορα και δωρεάν



About me

Η παρόν πλατφόρμα έγινε στα πλαίσια του Π.Μ.Σ. «Επιστήμη Υπολογιστών» του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Πελοποννήσου. Υπό την επίβλεψη του καθηγητή Κωνσταντίνου Βασιλάκη

Έτοιμα Template

Για την κατασκευή του παρόντος site χρησιμοποιήθηκαν template adminkit από την τοποθεσία github.com και το Clean Blog θέμα από το startbootstrap.com.

AdminKit [adminkit](#)

Clean Blog [Clean Blog](#)

Στοιχεία προγραμματισμού

Για την λειτουργικότητα έγινε χρήση PHP, Javascript, MySQL.

Εικόνα 26 Πληροφορίες εργαλείων ανάπτυξης και συντελεστών.

- **Επιλογή «API»:** Πληροφορίες σχετικά με τη χρήση του api για χρήση από άλλους ιστοχώρους.

RESTful API v.1
<https://api.dit2020cst-thesis.ga>

API Endpoint
 /seek

api.dit2020cst-thesis.ga/seek

Περιγραφή

Η αναζήτηση πακέτων είναι η μόνη API κλήση που δεν χρειάζεται αυθεντικοποίηση όταν πρόκειται για πακέτο που εξυπηρετείται από τον ίδιο τον ιστότοπο. Εάν το πακέτο που αναζητούμε ανήκει σε τρίτη εταιρία τότε θα πρέπει να γίνει αυθεντικοποίηση μέσω του API key.

`https://api.dit2020cst-thesis.ga/seek?{sydetalId}`

Δέχεται :
 {sydetalId}: Ο αριθμός πακέτου ΣΥ.ΔΕ.ΤΑ. Αν το πακέτο ανήκει στον ίδιο τον ιστόχωρο ή όχι θα το αποφασίσει το πρόγραμμα ανάλογα την μορφή που θα έχει ο γενικός αριθμός.
 (Η μορφή έχει οριστεί να είναι "#^ΧΑ[0-9]{9}GR\$#")

Επιστρέφει:

```
{
  "OWNER": "type Of Owner",
  "TRACKINGNUMBER": "Tracing Number",
  "ERROR": "isError-YES/NO",
  "DATA": ["Array of Data"]
  "MISCOATA": ["Miscellaneous Information"]
}
```

Εικόνα 27 Πληροφορίες για το προσφερόμενο API

- **Επιλογή «Επικοινωνία»:** Με αυτή την επιλογή δίνουμε τη δυνατότητα σε οποιονδήποτε χρήστη να έρθει σε επαφή με τον διαχειριστή της ιστοσελίδας. Τα μηνύματα που στέλνονται με αυτό τον τρόπο καταχωρούνται στη βάση δεδομένων και είναι διαθέσιμα στον διαχειριστή όταν αυτός θα εισέλθει ως πιστοποιημένος χρήστης.

Παρακολουθήση πακέτων
 Παρακολουθήστε το πακέτο σας γρήγορα και δωρεάν

Επικοινωνήστε μαζί μας.

Αν θελήσετε να επικοινωνήσετε μαζί μας παρακαλώ συμπληρώστε τα στοιχεία σας και το συντομότερο θα έρθουμε σε επαφή μαζί σας!

Όνομα

Διεύθυνση Email

Τηλέφωνο

Θέμα Μηνύματος

Μήνυμα

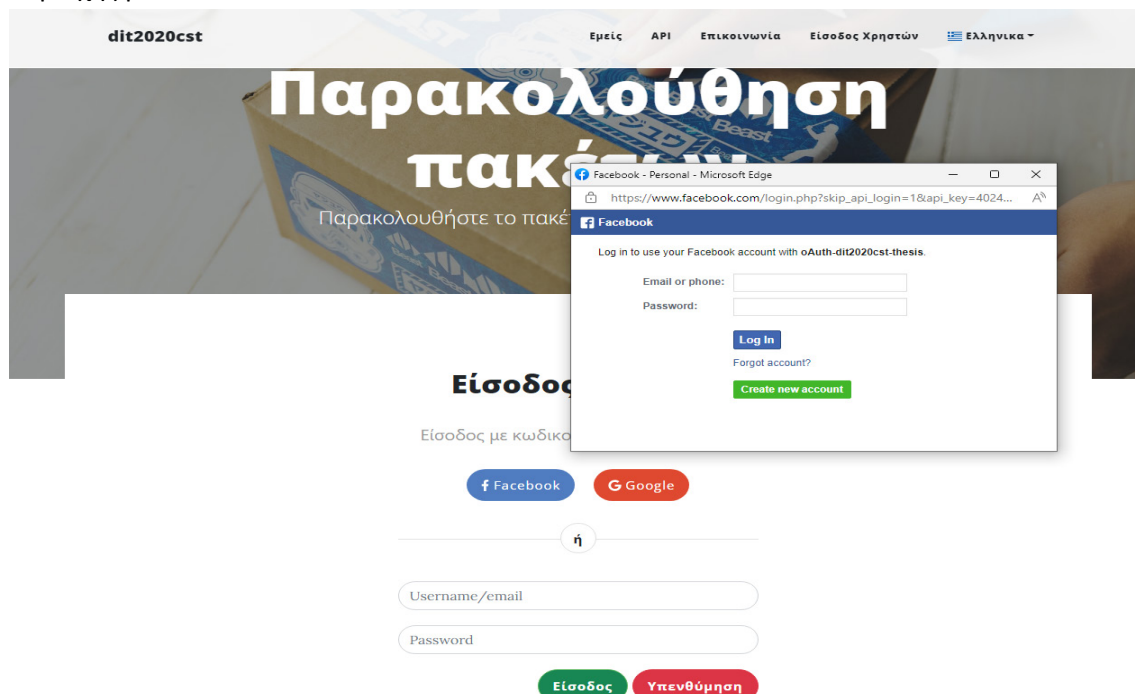
Αποστολή μηνύματος

Εικόνα 28 Επικοινωνία με διαχειριστές του ιστοχώρου

6.2 Εγγραφή και είσοδος χρηστών

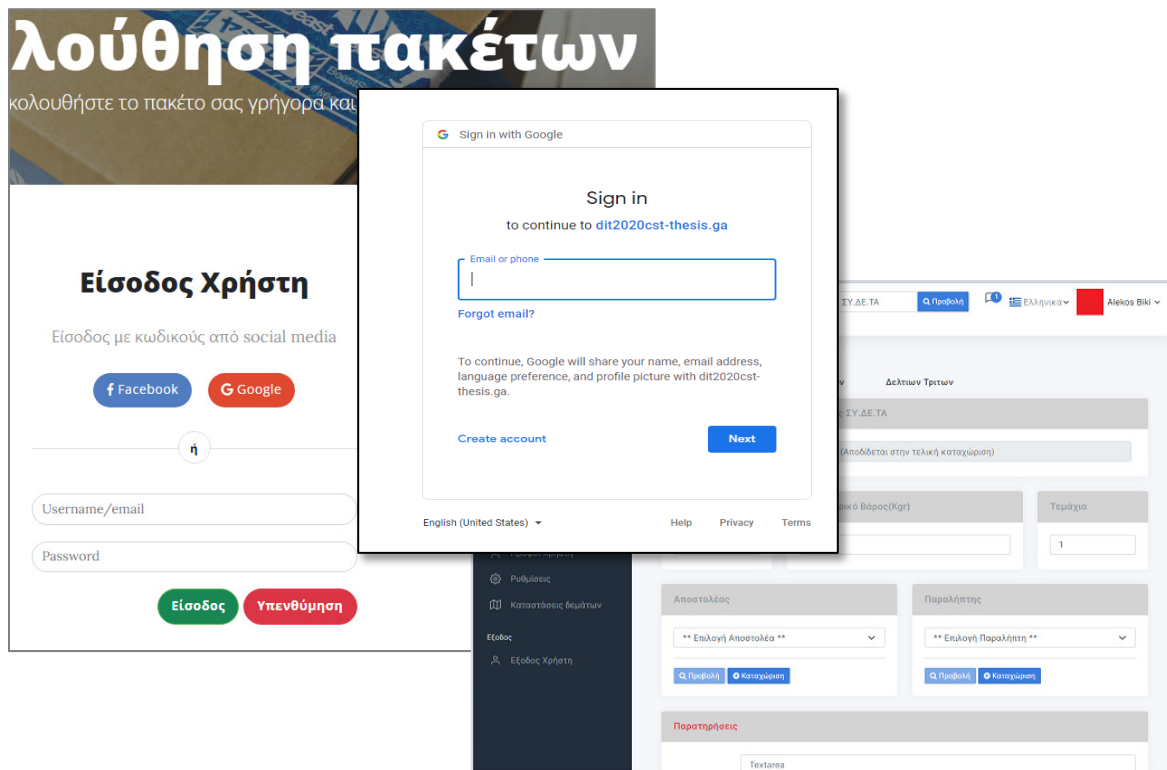
Η εφαρμογή έχει κατασκευαστεί έτσι ώστε να υπάρχει διαβάθμιση στα δικαιώματα των χρηστών. Κατά την πρώτη εκκίνηση της εφαρμογής και αφού οι βάσεις δεδομένων είναι κενές, ο πρώτος χρήστης που θα δημιουργηθεί θα αποκτήσει ταυτόχρονα και δικαιώματα υπερχρήστη («Super User»). Κάθε άλλος χρήστης που θα εγγραφεί θα αποκτά δικαιώματα απλού χρήστη εφαρμογής. Η διαφορά ανάμεσα στους χρήστες είναι ότι ο Super User έχει τη δυνατότητα να ρυθμίζει και να παραμετροποιεί το σύστημα, σε αντίθεση με τον απλό χρήστη που έχει μόνο δικαιώματα επί των εργασιών της ταχυμεταφορικής. Η είσοδος μπορεί να γίνει είτε μέσω Google, είτε μέσω Facebook, είτε με εγγραφή στο site με επικύρωση από τον ίδιο τον ιστότοπο. Για να μπορέσει η είσοδος μέσω Facebook και Google να λειτουργήσει θα πρέπει ο Super User να έχει ορίσει τα κλειδιά που έχουν παραχθεί από τις αντίστοιχες εφαρμογές, στην επιλογή «ρυθμίσεις».

- **Είσοδος μέσω Facebook:** Ανοίγει νέο παράθυρο προκειμένου ο χρήστης να εισάγει τα στοιχεία του στο facebook.com. Μετά από την επιτυχημένη είσοδο, ο χρήστης μεταφέρεται στη σελίδα της περιοχής μελών.



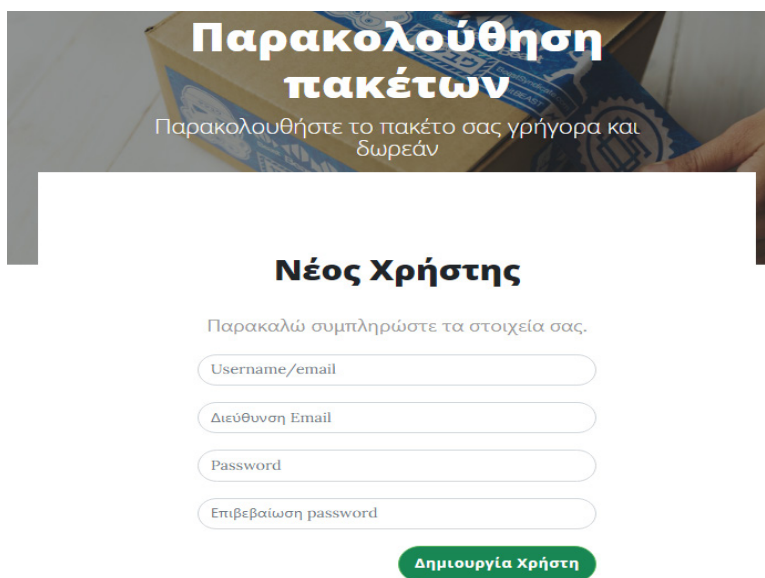
Εικόνα 29 Είσοδος χρήστη με χρήση Κωδικών Facebook

- **Είσοδος μέσω Google:** Ανοίγει νέο παράθυρο προκειμένου ο χρήστης να εισάγει τα στοιχεία του στο google.com. Μετά από την επιτυχημένη είσοδο, ο χρήστης μεταφέρεται στη σελίδα της περιοχής μελών.



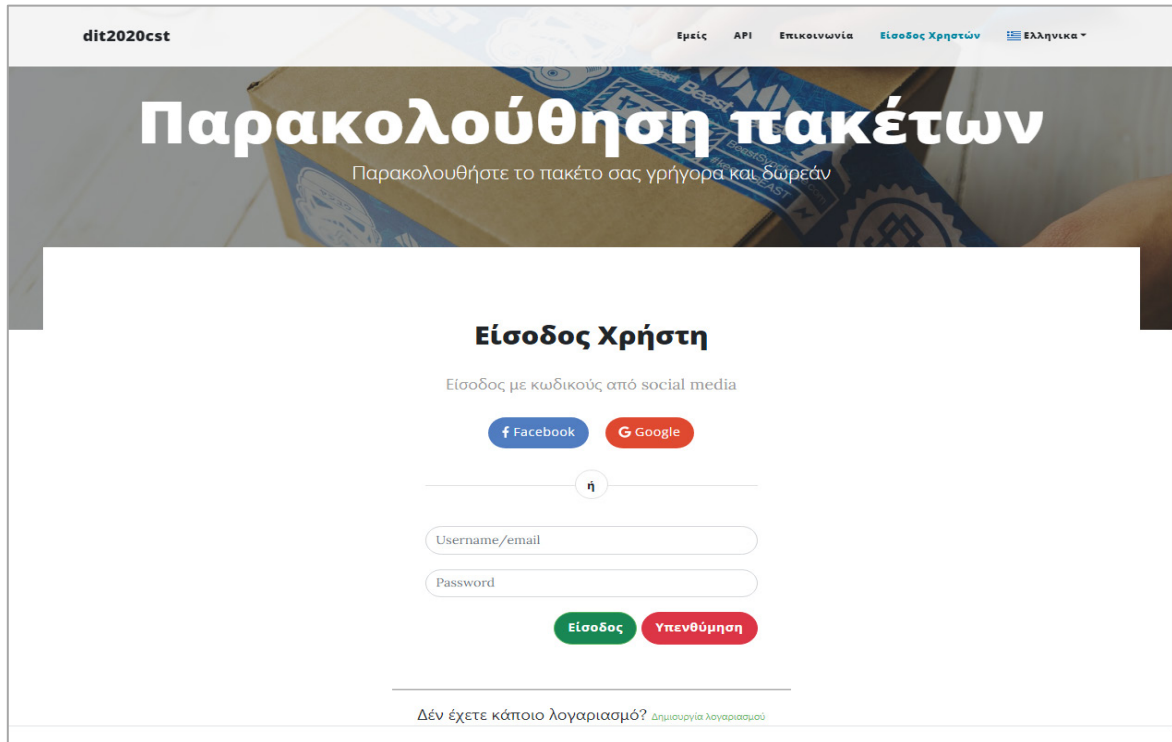
Εικόνα 30 Είσοδος χρηστών μέσω Google

- **Δημιουργία νέου χρήστη:** Κατά την είσοδο χρηστών, η εφαρμογή μας προτρέπει να δημιουργήσουμε έναν λογαριασμό για χρήση στον ιστόχωρο, αν δεν έχουμε ήδη έναν. Επιλέγοντας λοιπόν «Δημιουργία Λογαριασμού» μας μεταφέρει στην οθόνη δημιουργίας νέου χρήστη.

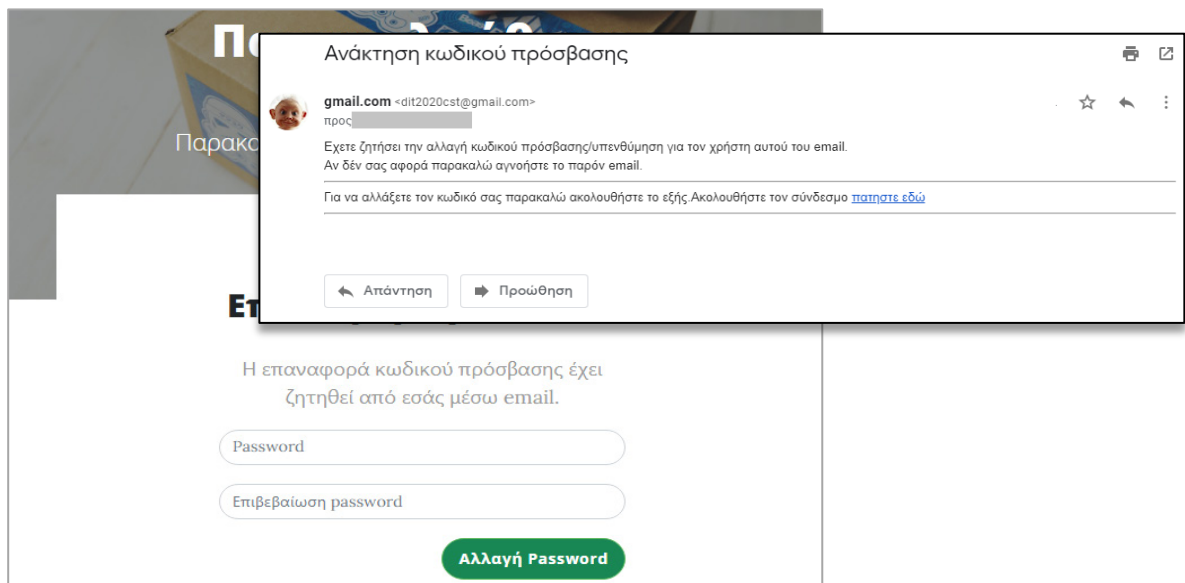


Εικόνα 31 Δημιουργία νέου χρήστη.

- **Υπενθύμιση στοιχείων χρήστη:** Αν ο χρήστης ξεχάσει τα στοιχεία του, τότε μπορεί να επιλέξει τη λειτουργία υπενθύμισης στοιχείων και να εισάγει το όνομα χρήστη ή το email του, και τότε το σύστημα θα του στείλει ένα ηλεκτρονικό μήνυμα με σύνδεσμο, μέσω του οποίου μπορεί να εκτελέσει αρχικοποίηση του κρυφού κωδικού. Ο χρήστης μέσω του συνδέσμου οδηγείται σε μία φόρμα καταχώρησης νέου κωδικού, και μέσω αυτής μπορεί να θέσει τον νέο κωδικό πρόσβασης, σε αντικατάσταση του παλαιού που έχει ξεχάσει.



Εικόνα 32 Υπενθύμιση κωδικού πρόσβασης



Εικόνα 33 Ανάκτηση κωδικού πρόσβασης

6.3 Χώρος διαχειριστή-πιστοποιημένων χρηστών

Όταν ο χρήστης ολοκληρώσει επιτυχημένα την είσοδο στο χώρο εργασίας, η αρχική σελίδα είναι η δημιουργία ενός νέου ΣΥ.ΔΕ.ΤΑ. Ανάλογα με το επίπεδο πρόσβασης που διαθέτει (Super User ή απλός χρήστης), πιθανόν να εμφανίζονται ή να αποκρύπτονται επιλογές και ενέργειες που μπορούν να εκτελεσθούν.

- **Κεντρική σελίδα, Εργασίες Μεταφοράς:: έκδοση ΣΥ.ΔΕ.ΤΑ.:** Αποτελεί την κεντρική καρτέλα εργασίας, μέσω της οποίας καταχωρούμε ένα νέο δέμα προς αποστολή. Ο Αριθμός ΣΥ.ΔΕ.ΤΑ. αποδίδεται αυτόματα στο τέλος της καταχώρισης, και τότε δημιουργούνται τα αντίστοιχα QR-code και Barcode που συνοδεύουν το δέμα. Στοιχεία όπως Αποστολέας ή Παραλήπτης μπορούν να καταχωρηθούν και να προβληθούν είτε από την εν λόγω οθόνη, είτε από τις αντίστοιχες επιλογές. Τέλος μετά την καταχώρηση μπορεί να γίνει και η αντίστοιχη εκτύπωση της ετικέτας. Όλα τα δέματα που καταχωρούνται, αρχικοποιούν την κατάσταση τους σύμφωνα με την ρύθμιση «Κατάσταση Δεμάτων» που υπάρχει στις ρυθμίσεις.

The screenshot displays the 'dit2020cst' web application interface for creating a 'ΣΥ.ΔΕ.ΤΑ' (SYDET). The left sidebar contains navigation options such as 'Κεντρικός Ιστότοπος', 'Εργασίες Μεταφοράς', 'ΣΥ.ΔΕ.ΤΑ', 'Αποστολείς', 'Παραλήπτες', 'Χρησιμες Εργασίες', 'Μηνύματα', 'Ρυθμίσεις', 'Προφίλ Χρήστη', 'Ρυθμίσεις', 'Καταστάσεις δεμάτων', 'Εξοδος', and 'Εξοδος Χρήστη'. The main content area features a search bar for 'ΣΥ.ΔΕ.ΤΑ' and a user profile 'anemos'. Below this, there are tabs for 'Καταχωρίσεις', 'Λίστα Δελτιων', and 'Δελτιων Τριτων'. The 'Καταχωρίσεις' tab is active, showing a form for 'Συνοδευτικό Δελτίου Ταχυμεταφοράς ΣΥ.ΔΕ.ΤΑ'. The form includes a field for 'Αριθμός ΣΥ.ΔΕ.ΤΑ' (auto-filled with 'Αριθμός ΣΥ.ΔΕ.ΤΑ (Αποδίδεται στην τελική καταχώριση)'), three input fields for 'Βάρος (Kgr)', 'Ογκομετρικό Βάρος (Kgr)', and 'Τεμάχια' (with values 0, 0, and 1 respectively). There are also dropdown menus for 'Αποστολέας' and 'Παραλήπτης', each with a search and 'Καταχώριση' button. At the bottom, there is a 'Παρατηρήσεις' section with a text area and buttons for 'Καταχώριση' and 'Εκτύπωση'. The footer contains 'Cookies' and 'Terms' links.

Εικόνα 34 Έκδοση ΣΥ.ΔΕ.ΤΑ.

dit2020cst

Κεντρικός Ιστότοπος

Εργασίες Μεταφοράς

ΣΥ.ΔΕ.ΤΑ

Αποστολές

Παραλήπτες

Χρήσιμες Εργασίες

Μηνύματα

Ρυθμίσεις

Προφίλ Χρήστη

Ρυθμίσεις

Καταστάσεις δεμάτων

Εξόδος

Εξόδος Χρήστη

ΣΥ.ΔΕ.ΤΑ Προβολή

Ελληνικά

anemos

Καταχωρίσεις Λίστα Δελτιών Δελτιών Τρίτων

Συνοδευτικό Δελτίου Ταχυμεταφοράς ΣΥ.ΔΕ.ΤΑ

Στοιχεία Αποστολέα

Επώνυμο Επώνυμο

Όνομα Όνομα

Διεύθυνση Διεύθυνση

Πόλη Πόλη Τ.Κ.

Τηλέφωνο Τηλέφωνο

Email Email

Ακύρωση Καταχώριση

Στοιχεία Παραλήπτη

Επώνυμο ΚΑΡΒΟΥΝΟΠΟΥΛΟΥ

Όνομα ΓΕΩΡΓΙΑ

Διεύθυνση ΖΑΚΥΝΘΟΥ 53

Πόλη ΚΕΡΚΥΡΑ Τ.Κ. 56897

Τηλέφωνο 295869458

Email kar.geo@hotmail.com

Επιστροφή

QR Code


Barcode

X A 0 0 0 0 0 4 1 1 G R

Νέο ΣΥ.ΔΕ.ΤΑ Καταχώριση Εκτύπωση

01/10/22, 11:48 AM Αποστολέας/Υπολήπτης

ΣΥΝΟΔΕΥΤΙΚΟ ΔΕΛΤΙΟ ΤΑΧΥΜΕΤΑΦΟΡΑΣ

<p>ΑΠΟΣΤΟΛΕΑΣ</p> <p>ΑΝΑΣΙΟΠΟΥΛΟΣ ΑΛΕΞΑΚΗΣ</p> <p>ΚΥΘΝΟΥ 55</p> <p>65489 - ΠΑΤΡΑ</p> <p>al.bl@gmail.com</p> <p>699225566</p>	<p>ΠΑΡΑΛΗΠΤΗΣ</p> <p>ΚΑΡΒΟΥΝΟΠΟΥΛΟΥ ΓΕΩΡΓΙΑ</p> <p>ΖΑΚΥΝΘΟΥ 53</p> <p>56897 - ΚΕΡΚΥΡΑ</p> <p>kar.geo@hotmail.com</p> <p>295869458</p>
<p>ΠΑΡΑΤΗΡΗΣΕΙΣ</p> <p>Ώρες παράδοσης 10-13</p>	<p>QR Code</p> 

Barcode

X A 0 0 0 0 0 4 1 1 G R

Print 1 page

Destination Save as PDF

Pages All



Layout Portrait

More settings

Save Cancel

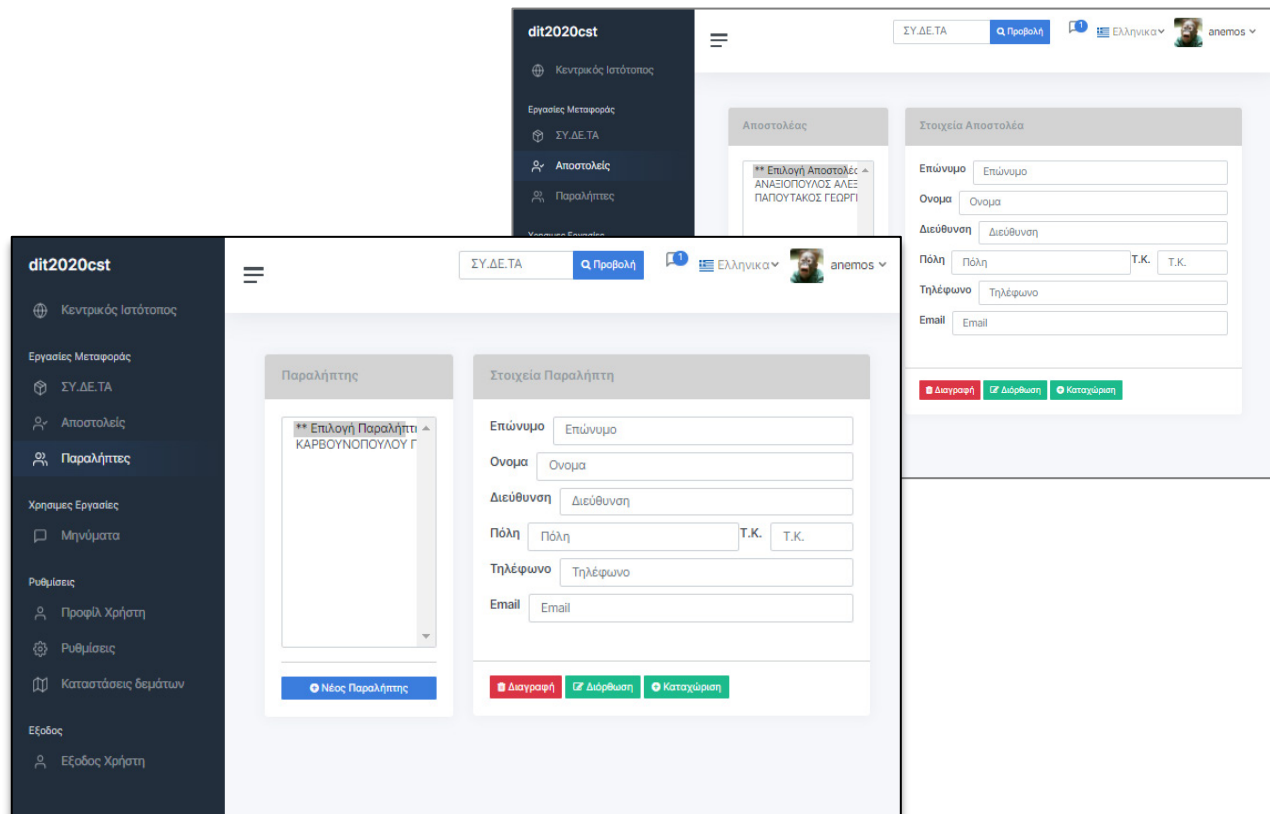
https://www.dit2020cst.thess.gub.gr/en/vr 1/1

Εικόνα 35 Επιλογή εκτύπωσης Δελτίου

ΣΥΝΟΔΕΥΤΙΚΟ ΔΕΛΤΙΟ ΤΑΧΥΜΕΤΑΦΟΡΑΣ	
ΑΠΟΣΤΟΛΕΑΣ ΠΑΠΟΥΤΣΑΝΗΣ ΓΕΩΡΓΙΟΥ ΜΑΚΑΡΙΟΥ 52 - ΙΩΑΝΝΙΝΑ pap.geo@yahoo.com 6897889859	ΠΑΡΑΛΗΠΤΗΣ ΚΑΡΒΟΥΝΙΑΡΗ ΓΕΩΡΓΙΑ ΖΑΚΥΝΘΟΥ 53 56897 - ΚΕΡΚΥΡΑ kar.geo@hotmail.com 295869458
ΠΑΡΑΤΗΡΗΣΕΙΣ Παράδοση πρωινές ώρες	QRcode 
Barcode  X A 0 0 0 0 0 3 9 9 G R	

Εικόνα 36 Εκτύπωση ΣΥ.ΔΕ.ΤΑ.

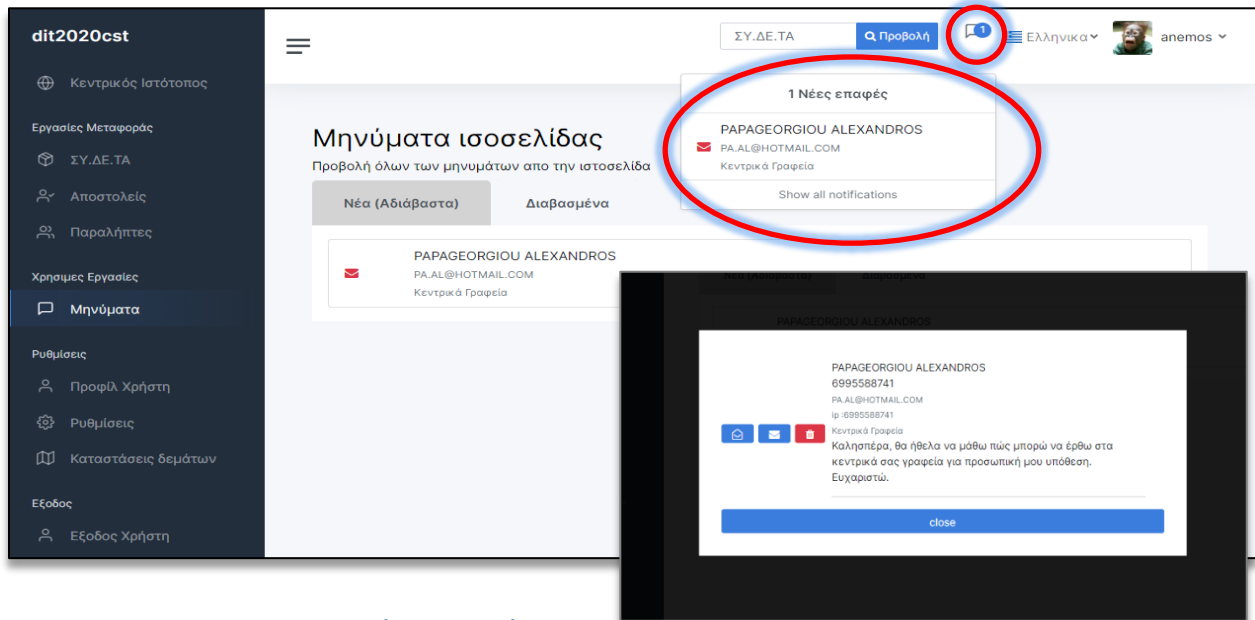
- **Εργασίες Μεταφοράς:: Αποστολείς - Παραλήπτες:** Παρ' όλο που η καταχώριση και η προβολή των αποστολών και παραληπτών μπορεί να γίνει από την οθόνη έκδοσης δελτίου, σε αυτές τις επιλογές μπορούμε να κάνουμε πλήρη διαχείριση αυτών (εισαγωγή, διόρθωση, διαγραφή και προβολή).



The screenshot displays the 'dit2020cst' web application interface. The top navigation bar includes the application name, a search bar for 'ΣΥ.ΔΕ.ΤΑ', a language selector set to 'Ελληνικά', and a user profile for 'anemos'. The main content area is divided into two sections: 'Αποστολέας' (Sender) and 'Παραλήπτης' (Recipient). The 'Αποστολέας' section shows a dropdown menu for selecting a sender, currently displaying 'ΑΝΑΣΙΟΠΟΥΛΟΣ ΑΛΕΞ ΠΑΠΟΥΤΑΚΟΣ ΓΕΩΡΓΙ'. The 'Παραλήπτης' section shows a dropdown menu for selecting a recipient, currently displaying 'ΚΑΡΒΟΥΝΟΠΟΥΛΟΥ Γ'. Both sections include a form for entering details such as 'Επώνυμο', 'Όνομα', 'Διεύθυνση', 'Πόλη', 'Τ.Κ.', 'Τηλέφωνο', and 'Email'. At the bottom of each section are buttons for 'Διαγραφή' (Delete), 'Διόρθωση' (Edit), and 'Καταχώριση' (Save). The left sidebar contains a menu with options like 'Κεντρικός Ιστότοπος', 'Εργασίες Μεταφοράς', 'ΣΥ.ΔΕ.ΤΑ', 'Αποστολείς', 'Παραλήπτες', 'Χρησιμες Εργασίες', 'Μηνύματα', 'Ρυθμίσεις', 'Προφίλ Χρήστη', 'Ρυθμίσεις', 'Καταστάσεις δεμάτων', and 'Εξόδος'.

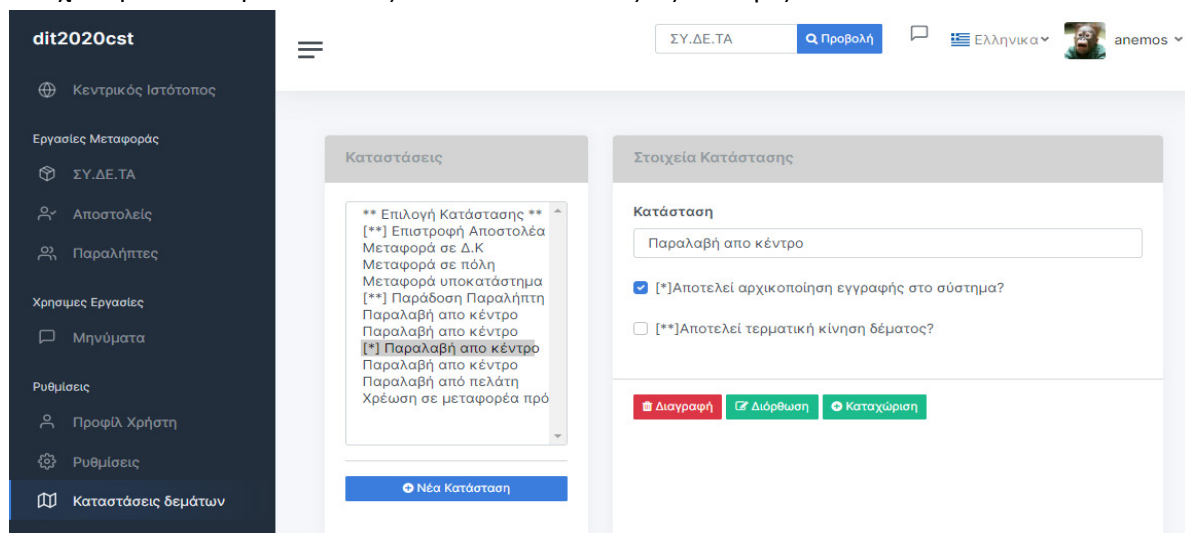
Εικόνα 37 Διαχείριση Αποστολέων και Παραληπτών

- Χρήσιμες Εργασίες:: Μηνύματα:** Τα μηνύματα που αποστέλλονται προς τον διαχειριστή μέσω της σελίδας «Επικοινωνήστε μαζί μας» καταχωρούνται στη βάση δεδομένων και είναι διαθέσιμα στους διαχειριστές. Στο κυρίως μενού εμφανίζεται διακριτικό σημάδι που μας πληροφορεί για την ύπαρξη νέων μηνυμάτων και από το κάθετο μενού επιλογών μπορούμε να οδηγηθούμε στην οθόνη διαχείρισής τους. Ένα μήνυμα έχουμε δικαίωμα να το σημειώσουμε ως αναγνωσμένο ή μη αναγνωσμένο, ή και να το διαγράψουμε.



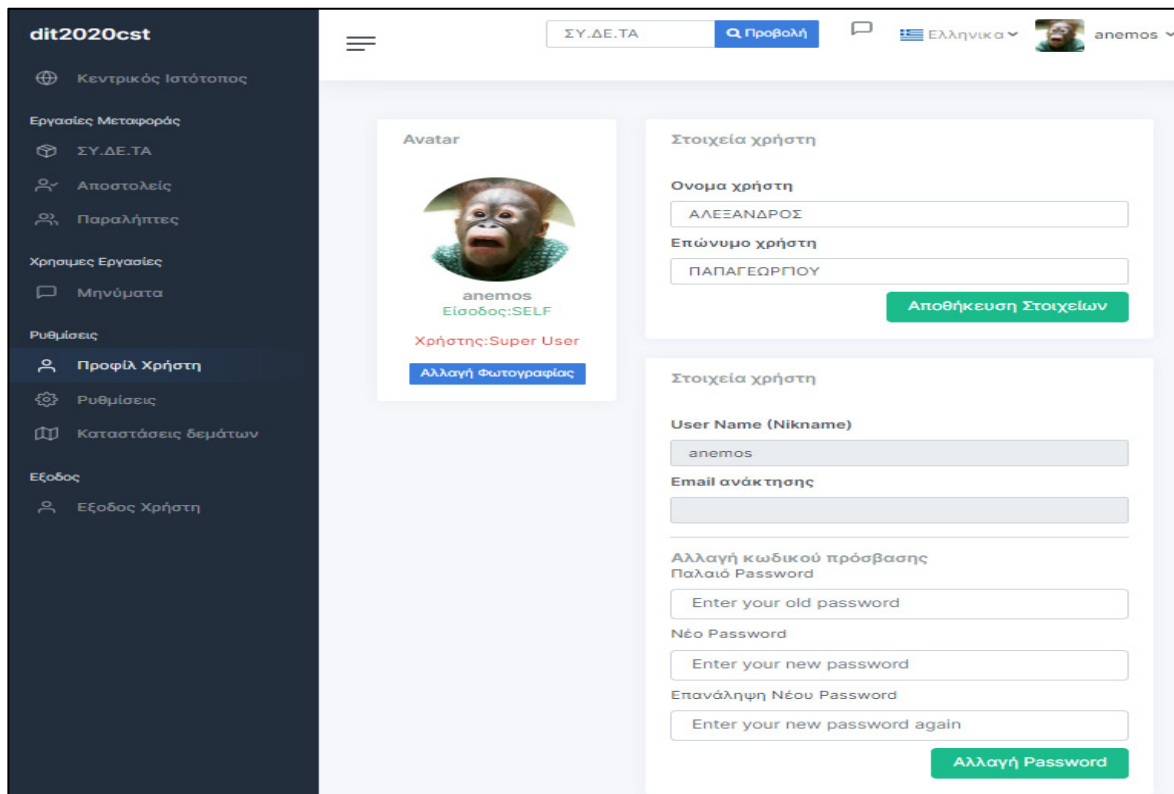
Εικόνα 38 Διαχείριση Μηνυμάτων επισκεπτών

- Ρυθμίσεις:: Καταστάσεις δερμάτων:** Μέσω της λειτουργικότητας της διαχείρισης των καταστάσεων των δερμάτων δημιουργούμε την αρχική, την τελική και τις ενδιάμεσες καταστάσεις που μπορεί να έχει ένα δέμα. Μόνο μία κατάσταση μπορεί να είναι αρχική κατάσταση δέματος ενώ πολλές μπορεί να αποτελούν τερματισμό της αποστολής. Αν δεν επιλέξουμε καμία από τις παραπάνω ιδιότητες τότε αυτή θα αποτελεί ενδιάμεση κατάσταση. Ο μεταφορέας που θα αναλάβει να παραδώσει ένα δέμα θα έχει τη δυνατότητα να επιλέξει κάποια από αυτές τις επιλογές.



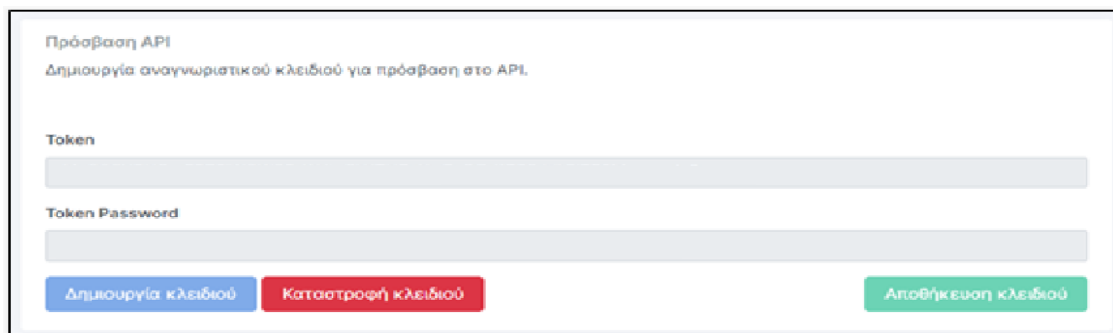
Εικόνα 39 Διαχείριση καταστάσεων δερμάτων

- **Ρυθμίσεις:: Προφίλ Χρήστη:** Μέσω της σελίδας του προφίλ του χρήστη έχουμε τη δυνατότητα να αλλάξουμε ιδιότητες του λογαριασμού όπως η εικόνα avatar , ονοματεπώνυμο χρήστη και κωδικούς πρόσβασης.



Εικόνα 40 Διαχείριση προφίλ χρήστη.

- **Ρυθμίσεις:: Ρυθμίσεις:** Για την επιλογή αυτή υπάρχει διαβαθμισμένη πρόσβαση και μόνο ο υπερχρήστης μπορεί να την ενεργοποιήσει. Ρυθμίζονται θέματα πρόσβασης του χρήστη στο API του προγράμματος, αλλά ταυτόχρονα καθορίζεται και ο τρόπος διασύνδεσης του δικού μας ιστότοπου με άλλους ιστότοπους από τους οποίους αντλούμε πληροφορίες. Οι ιδιότητες που μπορούμε να ρυθμίσουμε είναι:
 - **Πρόσβαση API:** Κάθε χρήστης έχει το δικό του κλειδί. Οι μεταφορείς με βάση αυτό το κλειδί μπορούν να πιστοποιήσουν τον περιηγητή τους ώστε να κάνουν αλλαγή στην κατάσταση δερμάτων.



Εικόνα 41 Κλειδί API

- Facebook και Google API: Ο υπερχρήστης καταχωρεί τα κλειδιά τα οποία έχει προμηθευτεί από τους αντίστοιχους ιστότοπους ώστε να είναι δυνατή η πρόσβαση χρηστών με τα στοιχεία που αυτοί οι ιστότοποι κατέχουν.

The image shows two side-by-side configuration panels. The left panel is titled 'Google API' with the subtitle 'Πρόσβαση στο API της google.' It contains two input fields: 'Google API client ID' and 'Google API client secret'. Below these are two buttons: a red one labeled 'Καταστροφή κλειδιού' and a green one labeled 'Αποθήκευση κλειδιού'. The right panel is titled 'Facebook API' with the subtitle 'Πρόσβαση στο API της facebook.' It also contains two input fields: 'Facebook API client ID' and 'Facebook API client secret'. Below these are two buttons: a red one labeled 'Καταστροφή κλειδιού' and a green one labeled 'Αποθήκευση κλειδιού'.

Εικόνα 42 Παράμετροι Facebook API και Google API

- Email provider: Ο υπερχρήστης καθορίζει τα στοιχεία σύνδεσης που θα έχει ο ιστότοπος με τον email server προκειμένου να μπορεί να χρησιμοποιηθεί η αποστολή email προς τους χρήστες.

The image shows a configuration form titled 'Email provider' with the subtitle 'Καθορισμός email για χρήση από την ιστοσελίδα'. It contains several input fields: 'Email Host' (smtp.gmail.com), 'Email Port' (465), 'Email SMTPAuth' (ON), 'Email SMTPSecure' (tls), 'Email From', 'Email From Name' (gmail.com), 'Email UserName', and 'Email Password'. At the bottom are two buttons: a red one labeled 'Καθαρισμός ρυθμίσεων' and a green one labeled 'Αποθήκευση ρυθμίσεων'.

Εικόνα 43 Παράμετροι αποστολής μηνυμάτων ηλεκτρονικού ταχυδρομείου

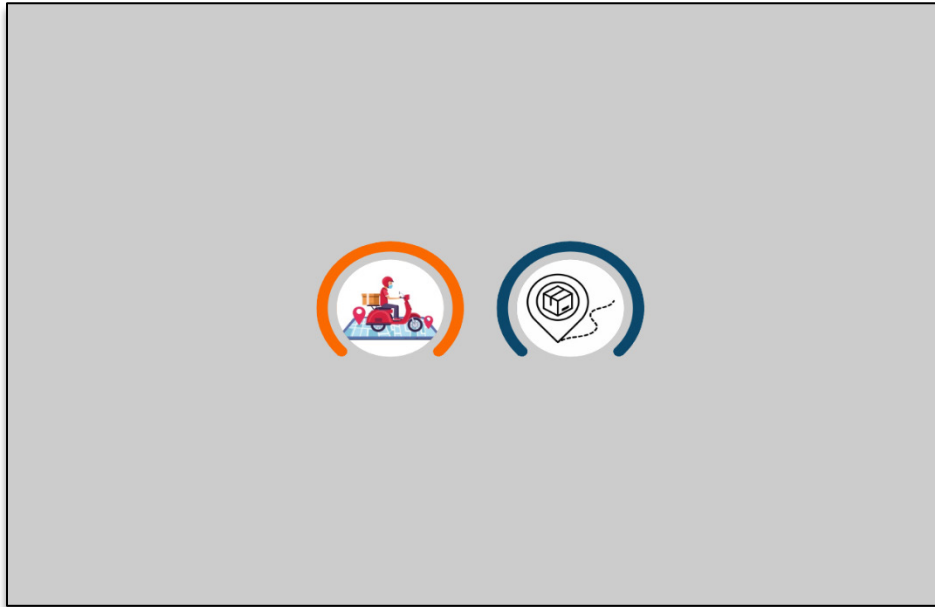
- 17track.com: Ο υπερχρήστης συγχρονίζει τα δεδομένα με αυτά του 17Track ώστε να μην χρειάζεται κάθε φορά να αντλούμε όλη την πληροφορία από αυτό, αλλά μόνο τα στοιχεία που μας ενδιαφέρουν. Με αυτό τον τρόπο έχουμε μικρότερο χρόνο υλοποίησης ερωτημάτων.

The image shows a configuration form titled '17Track.com' with the subtitle 'Συγχρονισμος api server me 17Track'. It contains two text labels: 'Αρχαιο Χωρων :' and 'Αρχαιο Ταχυμεταφορικών'. At the bottom right is a green button labeled 'Συγχρονισμος'.

Εικόνα 44 Συγχρονισμός API server με 17Track

6.4 Χώρος Μεταφορά

Ο κάθε μεταφορέας έχει τη δυνατότητα με το κινητό του τηλέφωνο να σαρώσει το QR-code που είναι εκτυπωμένο πάνω στο δέμα και να μεταφερθεί μέσω του περιηγητή του στην περιοχή της εφαρμογής όπου μπορεί να τροποποιηθεί η κατάσταση του δέματος. Για να λειτουργήσει η ανωτέρω διαδικασία, θα πρέπει ο περιηγητής να είναι πιστοποιημένος καταχωρώντας μία φορά το κλειδί API που του έχει αποδοθεί από την εφαρμογή. Σε περίπτωση λανθασμένης πρόσβασης μεταφέρεται στην αρχική σελίδα του ιστοτόπου.



Εικόνα 45 Πρόσβαση μη πιστοποιημένου browser στον χώρο μεταφορέων

The image displays a login form titled "Πιστοποίηση μεταφορέα" (Transporter Authentication). Below the title is the subtitle "Αποθήκευση και αρχικοποίηση κλειδιών χρήστη" (Storage and initialization of user keys). The form contains three input fields: "Username/email", "Token" (with the label "Κλειδί (Apikey)" below it), and "Token Password". At the bottom of the form are two buttons: a red button labeled "Ακύρωση" (Cancel) and a green button labeled "Πιστοποίηση Browser" (Browser Authentication).

Εικόνα 46 Πιστοποίηση browser μεταφορέα

Ο μεταφορέας έχει τη δυνατότητα να δει τη διαδρομή του πακέτου και να τροποποιήσει την κατάσταση του. Η τροποποίηση της κατάστασης οδηγεί σε προσθήκη μιας νέας καταχώρησης στη λίστα καταστάσεων του πακέτου.

The image displays two side-by-side screenshots of a tracking application interface for 'dit2020cst'. Both screens show a vertical timeline of events for a package. The left screenshot shows the current status as 'Παράδοση Παραλήπτη' (Delivered to Recipient) at 7:50 PM on 20/04/2022. The right screenshot shows the same timeline but with a dropdown menu open, allowing the user to change the status to 'Μεταφορά σε Δ.Κ.' (Transfer to D.K.).

Left Screenshot (Current Status):

- 20/04/2022 7:50 PM: **Παράδοση Παραλήπτη**
- 20/04/2022 7:49 PM: **Χρέωση σε μεταφορέα προς παράδοση**
- 20/04/2022 7:49 PM: **Μεταφορά σε Δ.Κ.**
- 20/04/2022 7:49 PM: **Μεταφορά σε πόλη**
- 20/04/2022 7:49 PM: **Μεταφορά υποκατάστημα**
- 20/04/2022 7:47 PM: **Παραλαβή απο κέντρο**
- 20/04/2022 7:34 PM: Παραλαβή από πελάτη

Right Screenshot (Status Change):

- 20/04/2022 7:50 PM: **Παράδοση Παραλήπτη**
- 20/04/2022 7:49 PM: **Χρέωση σε μεταφορέα προς παράδοση**
- 20/04/2022 7:49 PM: **Μεταφορά σε Δ.Κ.**
- 20/04/2022 7:49 PM: **Μεταφορά σε πόλη**
- 20/04/2022 7:49 PM: **Παραλαβή απο κέντρο**
- 20/04/2022 7:49 PM: **Μεταφορά υποκατάστημα**
- 20/04/2022 7:49 PM: **Χρέωση σε μεταφορέα προς παράδοση**
- 20/04/2022 7:49 PM: **Παράδοση Παραλήπτη**
- 20/04/2022 7:49 PM: **Επιστροφή Αποστολέα**
- 20/04/2022 7:49 PM: **Μεταφορά υποκατάστημα**
- 20/04/2022 7:47 PM: **Παραλαβή απο κέντρο**
- 20/04/2022 7:34 PM: Παραλαβή από πελάτη

Εικόνα 47 Tracking και αλλαγή κατάστασης δέματος από μεταφορέα

7 Συμπεράσματα

Στις μέρες μας οι εργασίες ταχυμεταφορών και παρακολούθησης των πακέτων από τους εμπλεκόμενους στην αποστολή είναι μία σύνθετη διαδικασία, η οποία όμως μπορεί να απλοποιηθεί με χρήση των κατάλληλων εφαρμογών και εργαλείων. Οι υπολογιστές, το διαδίκτυο αλλά και οι τεχνολογίες προγραμματισμού αποτελούν εργαλείο πρόσβασης του χρήστη στην πληροφορία που παράγεται μέσα από την μεταφορά πακέτων.

Στο πλαίσιο της παρούσας εργασίας αναπτύχθηκε ένα σύστημα που σεβόμενο τους κανόνες που ορίζονται από την ρυθμιστική αρχή για τον τρόπο λειτουργίας των ταχυμεταφορών (Εθνική Επιτροπή Τηλεπικοινωνιών και Ταχυδρομείων – ΕΕΤΤ) μπορεί να υλοποιήσει τις λειτουργικότητες που σχετίζονται με την παρακολούθηση των αποστολών και να αποτελέσει το εργαλείο πληροφόρησης χρηστών. Ταυτόχρονα μπορεί να καλύψει όλη την αλυσίδα ενεργειών που χρειάζονται ώστε ένα δέμα να παραδοθεί από τον αποστολέα στον παραλήπτη γνωρίζοντας πάντα τη θέση του. Το σύστημα είναι ανοικτό, διαθέτοντας ένα REST API, και νέα συστήματα μπορούν να αναπτυχθούν κάνοντας χρήση αυτού του API ώστε να εκτελέσουν βασικές ενέργειες στην εφαρμογή μας (λήψη δεδομένων/τροποποίηση καταστάσεων) και αξιοποιώντας τα αποτελέσματα για δόμηση υπηρεσιών προστιθέμενης αξίας.

Ως εργαλεία ανάπτυξης χρησιμοποιήθηκαν η HTML5, η CSS και η JAVASCRIPT προκειμένου η σχεδίαση της διεπαφής του χρήστη να είναι φιλική προς αυτόν. Ταυτόχρονα η JAVASCRIPT αποτέλεσε το εργαλείο σύνδεσης του client με τον server, αναλαμβάνοντας να στέλνει τα αιτήματα από την εφαρμογή χρήστη προς επεξεργασία στον server, αλλά και να εμφανίζει την απάντηση του server στον χρήστη. Στην πλευρά του server χρησιμοποιήθηκε η PHP για την υλοποίηση της διασύνδεσης με τον client και την υλοποίηση της επιχειρηματικής λογικής και η MYSQL για την διαχείριση και αποθήκευση των δεδομένων.

Μελλοντικές επεκτάσεις που θα μπορούσαν να γίνουν είναι να προστεθούν οικονομικά στοιχεία εντός της εφαρμογής σχετικά με την αποστολή των πακέτων, ώστε να είναι εφικτή η τιμολόγηση. Για πακέτα τρίτων ταχυμεταφορικών εταιρειών, αυτή τη στιγμή γίνεται χρήση της 17TRACK.COM ώστε να αντληθούν πληροφορίες σχετικές με τη θέση των πακέτων διεθνώς. Μελλοντικά μπορεί να υλοποιηθεί διασύνδεση και με άλλες εφαρμογές ώστε να υπάρχει δικαίωμα επιλογής από τον πιστοποιημένο χρήστη ή τον διαχειριστή. Τέλος θα μπορούσαν να υλοποιηθούν και άλλες εργασίες που θα διατίθενται μέσω API, ώστε να είναι εφικτή η ενσωμάτωσή τους σε διαφορετικό user interface ή/και σε άλλους ιστοτόπους.

8 Πηγές – Βιβλιογραφία

- [1] Εθνική Επιτροπή Τηλεπικοινωνιών και Ταχυδρομείων (ΕΕΤΤ) - Υποχρεώσεις αδειοδοτημένων επιχειρήσεων. Ανάκτηση από:
https://www.eett.gr/opencms/opencms/EETT/Postal_Services_n/Register_of_Postal_Services/GeneralLicense/Obligations/
- [2] Identification of postal items – 13-character identifier. Ανάκτηση από:
<https://www.upu.int/UPU/media/upu/files/postalSolutions/programmesAndServices/standards/S10-12.pdf>
- [3] S10 (UPU standard). Ανάκτηση από:
[https://en.wikipedia.org/wiki/S10_\(UPU_standard\)#:~:text=4%20External%20links-Format,issuing%20postal%20administration's%20country.](https://en.wikipedia.org/wiki/S10_(UPU_standard)#:~:text=4%20External%20links-Format,issuing%20postal%20administration's%20country.)
- [4] ISO 3166-1 alpha-2. Ανάκτηση από: https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2
- [5] Προγραμματισμός Διαδικτύου. Ανάκτηση από:
<http://arch.ict.e.uowm.gr/courses/webprogramming/>
- [6] AJAX XMLHttpRequest. Ανάκτηση από: <https://tutorial.smartcom.gr/02-ajax-xmlhttp/>
- [7] QR Codes. Ανάκτηση από:
https://developers.google.com/chart/infographics/docs/qr_codes?csw=1
- [8] Δι@ύγεια - API Ανοιχτών Δεδομένων , Τεκμηρίωση API Δι@ύγειας. Ανάκτηση από:
<https://diavgeia.gov.gr/api/help>
- [9] A Guide to the Entity Relationship Diagram (ERD). Ανάκτηση από:
<https://www.databasestar.com/entity-relationship-diagram/>
- [10] MIME types (IANA media types). Ανάκτηση από:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types
- [11] Statelessness in REST APIs). Ανάκτηση από:
<https://restfulapi.net/statelessness/>
- [12] HTTP Methods. Ανάκτηση από:
<https://restfulapi.net/http-methods/>
- [13] REST Resource Naming Guide. Ανάκτηση από:
<https://restfulapi.net/resource-naming/>
- [14] OAuth 2.0. Ανάκτηση από:
<https://oauth.net/2/>
- [15] What is SQL Injection (SQLi) and How to Prevent It. Ανάκτηση από:
<https://www.acunetix.com/websitesecurity/sql-injection/>
- [16] Document Object Model (DOM). Ανάκτηση από:
https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model