



ΓΕΩΠΟΝΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΑΓΡΟΤΙΚΗΣ ΠΑΡΑΓΩΓΗΣ, ΥΠΟΔΟΜΩΝ ΚΑΙ ΠΕΡΙΒΑΛΛΟΝΤΟΣ
ΤΜΗΜΑ ΑΞΙΟΠΟΙΗΣΗΣ ΦΥΣΙΚΩΝ ΠΟΡΩΝ ΚΑΙ ΓΕΩΡΓΙΚΗΣ ΜΗΧΑΝΙΚΗΣ

Π.Μ.Σ. «Φυσικοί Πόροι, Γεωπεριβάλλον και Γεωργική Μηχανική»

Κατεύθυνση «Γεωπληροφορική και Χωρική Ανάλυση»

Μεταπτυχιακή Εργασία

Χωρική ανάλυση ψεκασμού σε δενδρώδεις καλλιέργειες

Νέστωρ Ι. Λούγκος - Πανταζόπουλος

ΑΘΗΝΑ 2018

Επιβλέπων καθηγητής

Σπύρος Φουντάς



ΓΕΩΠΟΝΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΑΓΡΟΤΙΚΗΣ ΠΑΡΑΓΩΓΗΣ, ΥΠΟΔΟΜΩΝ ΚΑΙ ΠΕΡΙΒΑΛΛΟΝΤΟΣ
ΤΜΗΜΑ ΑΞΙΟΠΟΙΗΣΗΣ ΦΥΣΙΚΩΝ ΠΟΡΩΝ ΚΑΙ ΓΕΩΡΓΙΚΗΣ ΜΗΧΑΝΙΚΗΣ
Π.Μ.Σ. «Φυσικοί Πόροι, Γεωπεριβάλλον και Γεωργική Μηχανική»
Κατεύθυνση «Γεωπληροφορική και Χωρική Ανάλυση»

Μεταπτυχιακή Εργασία

Χωρική ανάλυση ψεκασμού σε δενδρώδεις καλλιέργειες

Νέστωρ Ι. Λούγκος - Πανταζόπουλος

ΑΘΗΝΑ 2018

Επιβλέπων καθηγητής

Σπύρος Φουντάς

Μεταπτυχιακή Εργασία

Χωρική ανάλυση ψεκασμού σε δενδρώδεις καλλιέργειες

Νέστωρ Ι. Λούγκος - Πανταζόπουλος

Επιβλέπων Καθηγητής

Σπύρος Φουντάς, Επίκουρος Καθηγητής

Εξεταστική Επιτροπή:

Σπύρος Φουντάς, Επίκουρος Καθηγητής

Διονύσιος Καλύβας, Αναπληρωτής Καθηγητής

Κωνσταντίνος Γιαλούρης, Καθηγητής

Ευχαριστίες

Ευχαριστώ όλους όσους με οποιονδήποτε τρόπο βοήθησαν στην ολοκλήρωση της παρούσας μεταπτυχιακής εργασίας.

Περίληψη

Τα σύγχρονα αγροκτήματα λειτουργούν αρκετά διαφορετικά σε σχέση με παλαιότερα, κυρίως λόγω των τεχνολογικών εξελίξεων των τελευταίων ετών. Ο τομέας που έφερε αυτή την αλλαγή στο χώρο της γεωργίας είναι γνωστός ως Information and Communication Technology in agriculture (ICT in agriculture, γνωστός επίσης και ως e-agriculture). Οι γεωργικές εφαρμογές του ICT βασίζονται σε πολλές τεχνολογίες που έγιναν ευρύτατα διαδεδομένες τις τελευταίες δεκαετίες. Μερικές από τις κύριες τεχνολογίες που χρησιμοποιούνται στην e-agriculture είναι τα Παγκόσμια Δορυφορικά Συστήματα Πλοήγησης (Global Navigation Satellite Systems – GNSS), τα ασύρματα δίκτυα, τα Γεωγραφικά Συστήματα Πληροφοριών (Geographical Information Systems – GIS) και οι εφαρμογές κινητών τηλεφώνων. Χρησιμοποιώντας αυτές τις τεχνολογίες οι γεωργοί παράγουν μια μεγάλη ποσότητα δεδομένων τα οποία πρέπει να τα διαχειριστεί το λογισμικό. Κατά τη διάρκεια των τελευταίων ετών ξεκίνησε η χρήση των εφαρμογών ιστού (web applications) στη γεωργία, η οποία μπορεί να λύσει αποδοτικά το πρόβλημα του χειρισμού και της παρουσίασης των δεδομένων.

Το αντικείμενο αυτής της μεταπτυχιακής εργασίας είναι η ανάπτυξη μιας εφαρμογής ιστού με σκοπό την παρουσίαση και το χειρισμό των χωρικών δεδομένων που προκύπτουν από τις επεμβάσεις ψεκασμού στις δενδρώδεις καλλιέργειες, χρησιμοποιώντας κατά κύριο λόγο ελεύθερο και ανοιχτού κώδικα λογισμικό.

Ο πυρήνας της εφαρμογής στην πλευρά του διακομιστή (server-side) αναπτύχθηκε στην Python, χρησιμοποιώντας το Flask Web Framework. Το Σύστημα Διαχείρισης Σχεσιακής Βάσης Δεδομένων που χρησιμοποιήθηκε ήταν η PostgreSQL. Η υλοποίηση στην πλευρά του πελάτη (client-side) δημιουργήθηκε με το συνδυασμό των τεχνολογιών JavaScript (ES6), HTML5 και CSS3. Το Linux είναι το λειτουργικό σύστημα του διακομιστή στο οποίο μεταφέρθηκε η εφαρμογή.

Πριν από την υλοποίηση της εφαρμογής πραγματοποιήθηκε μια ανάλυση των απαιτήσεων της. Η ανάλυση απαιτήσεων είναι σημαντική, επειδή μπορεί να κάνει τη διαφορά ανάμεσα στην επιτυχία και την αποτυχία ενός project λογισμικού. Κατά τη διάρκεια της ανάλυσης απαιτήσεων καταγράφηκαν οι λειτουργικές και μη λειτουργικές απαιτήσεις, οι χειριστές και οι περιπτώσεις χρήσης του συστήματος. Ο σχεδιασμός της εφαρμογής ακολούθησε την ανάλυση απαιτήσεων. Κατά το στάδιο αυτό ορίστηκε το σχήμα της βάσης δεδομένων και η αρχιτεκτονική και δομή της εφαρμογής.

Ένας πιθανός χρήστης της εφαρμογής θα πρέπει πρώτα να δημιουργήσει ένα λογαριασμό, στη συνέχεια ο εγγεγραμμένος χρήστης θα πρέπει να εισέλθει στο λογαριασμό που δημιούργησε. Για να είναι εφικτή η χρησιμοποίηση των λειτουργιών της εφαρμογής θα πρέπει ο γεωργός να ψηφιοποιήσει ένα αγρόκτημα και να καταγράψει μια επέμβαση ψεκασμού. Επίσης, ο χρήστης της εφαρμογής μπορεί να χρησιμοποιήσει τα χωρικά εργαλεία που αναπτύχθηκαν με σκοπό την

πραγματοποίηση βασικών χωρικών υπολογισμών. Η κύρια λειτουργικότητα της εφαρμογής αποτελείται από: 1) μια προβολή χάρτη, όπου οι χρήστες μπορούν να δουν τη διαδρομή της επέμβασης ψεκασμού, 2) μια προβολή πίνακα με τα δεδομένα, όπου παρουσιάζονται τα δεδομένα μιας επέμβασης ψεκασμού και 3) μια προβολή γραφήματος, όπου οι χρήστες μπορούν να δουν το ποσοστό των εγγραφών ταχύτητας σε σχέση με τη βέλτιστη ταχύτητα.

Μετά τη μεταφορά της εφαρμογής στο διακομιστή ακολούθησε ένας εκτεταμένος έλεγχος σε πέντε κατηγορίες, οι οποίες είναι: λειτουργικότητα, ευχρηστία, συμβατότητα, απόδοση και ασφάλεια.

Τέλος, τα κύρια συμπεράσματα που προέκυψαν, μεταξύ άλλων, μετά την ανάπτυξη της εφαρμογής, είναι ότι το λογισμικό ανοικτού κώδικα προσφέρει πολλές δυνατότητες. Επίσης, η δυνατότητα που έχουν οι εφαρμογές ιστού να λειτουργούν σε διαφορετικά λειτουργικά συστήματα και συσκευές αποτελεί μεγάλη διευκόλυνση για το χρήστη τους. Ενώ, η ασφάλεια μιας εφαρμογής ιστού είναι κάτι που θα πρέπει να λαμβάνεται σοβαρά υπόψιν κατά τη διάρκεια της ανάπτυξής της. Ως μελλοντικές επεκτάσεις της εφαρμογής, αυτές θα μπορούσαν να είναι η εισαγωγή τεχνικών από το πεδίο της επιστήμης των δεδομένων και η αύξηση των χωρικών λειτουργιών.

Λέξεις κλειδιά: Python, Flask, PostgreSQL, JavaScript, Google Maps API, Geolocation API, Turf.js, Εφαρμογή ιστού.

Abstract

Modern farms operate far differently than those a few decades ago, mainly because of advancements in technology. The sector responsible for that shift is known as Information and Communication Technology in agriculture (ICT in agriculture, also known as e-agriculture). Agricultural applications of ICT rely on many technologies that became mainstream over the last decades. Some of the main technologies used in e-agriculture are the Global Navigation Satellite Systems (GNSS), the wireless networks, the Geographic Information Systems (GIS), and the mobile apps. By using these technologies farmers generate a vast amount of data which has to be managed by software. During the last couple of years has started the adoption of web applications in agriculture, that kind of applications can resolve data related issues.

The objective of this thesis is the development of a web application in order to record, visualize and manipulate spatial data of orchard spraying operations, relying foremost on free and open source software.

Server-side development was done in Python, using the Flask Web Framework. The Relational Database Management System used is the PostgreSQL. The implementation of the client-side was done with a combination of JavaScript (ES6), HTML5, and CSS3 technologies. The operating system of the virtual private server is Linux.

Requirements analysis was the first step of application's design procedure. Requirement analysis is critical to the success or failure of a software project. Functional and non-functional requirements, actors, and use cases were specified during the requirements analysis. The Design of the application came up after the requirement analysis. At this stage the database schema and the application's architecture and structure were defined.

Anonymous users have to create an account in order to use the application. Firstly, a registered user should digitize a farm field and then record a spraying operation. Also, the farmer can make use of the spatial tools which are developed for basic spatial calculations. The main web application's functionality, consists of: 1) a map view, where users can see the full path of the spraying operation, 2) a data table view, where the operation data being displayed, and 3) a chart view, where users can watch the percentage of speed records relative to the optimal speed.

An extended testing was taken after the deployment process. The five domains of the testing was: functionality, usability, compatibility, performance, and security.

The main conclusion that can be drawn is that open source software provides a lot of capabilities. Also, web applications can serve as true cross-platform software solutions. However, web applications security is an issue that should be taken into consideration by developers. Looking forward, the application can benefit from the addition of data science techniques and advanced spatial data algorithms.

Keywords: Python, Flask, PostgreSQL, JavaScript, Google Maps API, Geolocation API, Turf.js, Web Application.

Περιεχόμενα

| | |
|---|----|
| Περίληψη..... | 4 |
| Abstract | 6 |
| 1 Εισαγωγή | 15 |
| 1.1 Διαδίκτυο και Παγκόσμιος Ιστός..... | 15 |
| 1.1.1 Διαδίκτυο (Internet) | 15 |
| 1.1.2 Παγκόσμιος Ιστός (World Wide Web)..... | 16 |
| 1.1.2.1 Γενικά..... | 16 |
| 1.1.2.2 Επισκόπηση της Αρχιτεκτονικής..... | 16 |
| 1.1.2.3 Η πλευρά του πελάτη..... | 18 |
| 1.1.2.4 Η πλευρά του διακομιστή | 18 |
| 1.2 Βάσεις Δεδομένων | 19 |
| 1.2.1 Εισαγωγή | 19 |
| 1.2.2 Το Σχεσιακό Μοντέλο..... | 19 |
| 1.3 Γεωγραφικά Πληροφοριακά Συστήματα | 20 |
| 1.3.1 Γενικά..... | 20 |
| 1.3.2 Χαρτογραφικές Προβολές | 20 |
| 1.3.3 Γεωγραφικές Συντεταγμένες..... | 20 |
| 1.3.4 Παγκόσμια Δορυφορικά Συστήματα Πλοήγησης | 21 |
| 1.3.4.1 Global Positional System (GPS) | 21 |
| 1.3.5 Χωρικά Καθορισμένη Ψηφιακή Γεωργία | 22 |
| 1.4 Πληροφοριακά Συστήματα στη Γεωργία | 23 |
| 1.4.1 Εισαγωγή | 23 |
| 1.4.2 Πληροφοριακά Συστήματα Διαχείρισης Αγροκτήματος (ΠΣΔΑ)..... | 24 |
| 1.4.3 Υλοποιήσεις..... | 26 |
| 1.5 Εφαρμογή υγρών φυτοπροστατευτικών | 32 |
| 1.5.1 Γενικά..... | 32 |
| 1.5.2 Μηχανοκίνητα ψεκαστικά μηχανήματα | 32 |
| 1.5.2.1 Περιγραφή..... | 32 |
| 1.5.2.2 Ψεκαστικά για δενδρώδεις καλλιέργειες | 33 |
| 1.5.2.3 Ρύθμιση ψεκαστικών | 34 |
| 1.6 Σκοπός | 34 |
| 2 Υλικά και Μέθοδοι | 36 |
| 2.1. Η γλώσσα προγραμματισμού Python | 36 |

| | |
|---|----|
| 2.1.1 Εισαγωγή | 36 |
| 2.1.2 Συντακτικό | 36 |
| 2.1.2.1 Βασική δομή ενός σεναρίου (script) | 36 |
| 2.1.2.2 Μεταβλητές | 37 |
| 2.1.2.3 Εκφράσεις | 38 |
| 2.1.2.4 Τύποι Δεδομένων | 38 |
| 2.1.2.5 Έλεγχος ροής | 45 |
| 2.1.2.6 Συναρτήσεις | 48 |
| 2.1.2.7 Κλάσεις | 48 |
| 2.1.2.8 Εξαιρέσεις | 49 |
| 2.1.2.9 Αρχεία | 51 |
| 2.2 Flask Web Framework | 52 |
| 2.3 SQL | 53 |
| 2.3.1 Εισαγωγή στην SQL | 53 |
| 2.3.2 Το συντακτικό της SQL | 54 |
| 2.4 PostgreSQL | 59 |
| 2.5 Η Object-Relational Mapping (ORM) εργαλειοθήκη SQLAlchemy | 60 |
| 2.6 Nginx | 61 |
| 2.7 Gunicorn | 61 |
| 2.8 Η γλώσσα προγραμματισμού JavaScript | 62 |
| 2.8.1 Εισαγωγή | 62 |
| 2.8.2 Συντακτικό | 62 |
| 2.8.2.1 Μεταβλητές | 62 |
| 2.8.2.2 Τύποι Δεδομένων | 64 |
| 2.8.2.3 Συμβολοσειρές | 65 |
| 2.8.2.4 Αντικείμενα | 66 |
| 2.8.2.5 Πίνακες | 67 |
| 2.8.2.6 Έλεγχος ροής | 68 |
| 2.8.2.7 Συναρτήσεις | 69 |
| 2.8.2.8 Κλάσεις | 70 |
| 2.9 HyperText Markup Language (HTML) | 71 |
| 2.10 Cascading Style Sheets (CSS) | 71 |
| 2.11 Bootstrap | 71 |
| 2.12 jQuery | 72 |
| 2.13 Asynchronous JavaScript και XML (AJAX) | 72 |
| 2.14 JavaScript Object Notation (JSON) | 73 |

| | |
|---|-----|
| 2.15 GeoJSON | 73 |
| 2.16 Geolocation API | 74 |
| 2.16.1 Γενικά..... | 74 |
| 2.16.2 Χρήση του Geolocation API..... | 75 |
| 2.17 Google Maps API | 77 |
| 2.17.1 Γενικά..... | 77 |
| 2.17.2 Χρήση του Google Maps API | 77 |
| 2.18 Turf.js..... | 79 |
| 2.18.1 Γενικά..... | 79 |
| 2.18.2 Χρήση της βιβλιοθήκης Turf.js | 79 |
| 2.19 Google Charts | 81 |
| 2.20 Chart.js..... | 81 |
| 2.21 Linux | 81 |
| 3 Ανάλυση Απαιτήσεων και Σχεδιασμός της εφαρμογής | 82 |
| 3.1 Απαιτήσεις..... | 82 |
| 3.2 Χαρακτήρες (Actors)..... | 83 |
| 3.3 Περιπτώσεις χρήσης (Use Cases) | 83 |
| 3.3.1 Δημιουργία λογαριασμού | 85 |
| 3.3.2 Είσοδος στο λογαριασμό..... | 87 |
| 3.3.3 Ψηφιοποίηση αγροτεμαχίου | 89 |
| 3.3.4 Προβολή στοιχείων αγροτεμαχίου | 90 |
| 3.3.5 Καταγραφή επέμβασης ψεκασμού..... | 92 |
| 3.3.6 Μεταφορά αρχείου στο διακομιστή..... | 94 |
| 3.3.7 Προβολή των στοιχείων ψεκασμού | 96 |
| 3.3.8 Εύρεση συντεταγμένων σημείου | 99 |
| 3.3.9 Υπολογισμός απόστασης μεταξύ δύο σημείων | 100 |
| 3.3.10 Εύρεση συντεταγμένων του μέσου δύο σημείων | 101 |
| 3.3.11 Εμφάνιση buffer zone | 102 |
| 3.3.12 Υπολογισμός μήκους γραμμής..... | 103 |
| 3.3.13 Υπολογισμός εμβαδού πολυγώνου | 104 |
| 3.3.14 Εύρεση κεντροειδούς πολυγώνου | 105 |
| 3.4 Σχεδιασμός της Βάσης Δεδομένων | 107 |
| 3.4.1 Πίνακας User | 107 |
| 3.4.2 Πίνακας Field | 108 |
| 3.4.3 Πίνακας Operation | 110 |
| 3.4.4 Πίνακας OperationData..... | 111 |

| | |
|---|-----|
| 3.5 Αρχιτεκτονική και Δομή της εφαρμογής..... | 112 |
| 4 Η εφαρμογή..... | 118 |
| 4.1 Εγκατάσταση και ρύθμιση | 118 |
| 4.2 Περιγραφή..... | 119 |
| 4.2 Έλεγχος της εφαρμογής | 136 |
| 4.2.1 Λειτουργικότητα..... | 136 |
| 4.2.2 Ευχρηστία | 150 |
| 4.2.3 Συμβατότητα | 162 |
| 4.2.4 Απόδοση..... | 165 |
| 4.2.5 Ασφάλεια..... | 168 |
| 5 Συμπεράσματα | 171 |
| Βιβλιογραφία | 173 |

Κατάλογος Εικόνων

| | |
|---|----|
| Εικόνα 1.1: Ποσοστό του πληθυσμού κάθε χώρας που κάνει χρήση του Διαδικτύου | 15 |
| Εικόνα 1.2: Αναπαράσταση του Διαδικτύου από το Internet Mapping Project των Bell Labs | 15 |
| Εικόνα 1.3: Η αρχιτεκτονική του Ιστού | 17 |
| Εικόνα 1.4: Αναπαράσταση της αλληλεπίδρασης μεταξύ πελάτη (client) και διακομιστή (web server)..... | 17 |
| Εικόνα 1.5: Το Καρτεσιανό και το σφαιροειδές σύστημα συντεταγμένων. | 21 |
| Εικόνα 1.6: Η κατανομή των ΠΣΔΑ ανά πλατφόρμα λειτουργίας σύμφωνα με τη μελέτη των Fountas et al. (2015)..... | 25 |
| Εικόνα 1.7: Η αρχιτεκτονική ενός ΠΣΔΑ από την οπτική ενός απλού χρήστη | 26 |
| Εικόνα 1.8: Η αρχιτεκτονική ενός ΠΣΔΑ από την οπτική του υπεύθυνου ανάπτυξης..... | 26 |
| Εικόνα 1.9: Η ροή των διεργασιών του ΠΣΔΑ για τα οπωροφόρα δέντρα | 27 |
| Εικόνα 1.10: Εισαγωγή των δεδομένων των δένδρων | 27 |
| Εικόνα 1.11: Ο χάρτης της θερμοκρασίας της κόμης όπως παρουσιάζεται στην εφαρμογή ιστού | 28 |
| Εικόνα 1.12: Ένα χάρτης απόδοσης της εφαρμογής AGRI-AG..... | 29 |
| Εικόνα 1.13: Η αρχιτεκτονική της πλατφόρμας του διακομιστή ελέγχου του συστήματος .. | 29 |
| Εικόνα 1.14: Το γραφικό περιβάλλον του συστήματος τηλεμετρίας με την καταγραφή της διαδρομής | 30 |
| Εικόνα 1.15: Η επιχειρησιακή αρχιτεκτονική του συστήματος Cropinfra | 31 |
| Εικόνα 1.16: Η αρχιτεκτονική του συστήματος για τον ψεκασμό ακριβείας | 31 |
| Εικόνα 2.1: Οι ειδικές μέθοδοι που καλούνται από τη Python | 41 |
| Εικόνα 2.2: Το διάγραμμα των πινάκων της βάσης δεδομένων των παραδειγμάτων | 54 |
| Εικόνα 3.1: Διάγραμμα περιπτώσεων χρήσης για τον ανώνυμο χρήστη | 83 |
| Εικόνα 3.2: Διάγραμμα περιπτώσεων χρήσης για τον εγγεγραμμένο χρήστη | 84 |
| Εικόνα 3.3: Το διάγραμμα δραστηριότητας για τη δημιουργία λογαριασμού | 86 |
| Εικόνα 3.4: Το διάγραμμα δραστηριότητας για την είσοδο στο λογαριασμό..... | 88 |

| | |
|---|-----|
| Εικόνα 3.5: Το διάγραμμα δραστηριότητας για τη ψηφιοποίηση ενός αγροτεμαχίου | 90 |
| Εικόνα 3.6: Το διάγραμμα δραστηριότητας για την προβολή των στοιχείων ενός αγροτεμαχίου | 91 |
| Εικόνα 3.7: Το διάγραμμα δραστηριότητας για την καταγραφή μιας επέμβασης ψεκασμού | 93 |
| Εικόνα 3.8: Το διάγραμμα δραστηριότητας για τη μεταφορά του αρχείου | 95 |
| Εικόνα 3.9: Το διάγραμμα δραστηριότητας για την προβολή των στοιχείων ενός ψεκασμού | 98 |
| Εικόνα 3.10: Το διάγραμμα δραστηριότητας για την εύρεση των συντεταγμένων ενός σημείου | 99 |
| Εικόνα 3.11: Το διάγραμμα δραστηριότητας για τον υπολογισμό της απόστασης μεταξύ δύο σημείων | 100 |
| Εικόνα 3.12: Το διάγραμμα δραστηριότητας για την εύρεση των συντεταγμένων του μέσου δύο σημείων..... | 101 |
| Εικόνα 3.13: Το διάγραμμα δραστηριότητας για την εμφάνιση buffer zone γύρω από ένα σημείο | 103 |
| Εικόνα 3.14: Το διάγραμμα δραστηριότητας για τον υπολογισμό του μήκους μιας γραμμής | 104 |
| Εικόνα 3.15: Το διάγραμμα δραστηριότητας για τον υπολογισμό του εμβαδού | 105 |
| Εικόνα 3.16: Το διάγραμμα δραστηριότητας για την εμφάνιση του κεντροειδούς ενός πολυγώνου | 106 |
| Εικόνα 3.17: Το σχήμα της Βάσης Δεδομένων..... | 107 |
| Εικόνα 3.18: Μια γενική εικόνα του συστήματος..... | 113 |
| Εικόνα 3.19: Η αρχιτεκτονική της εφαρμογής | 113 |
| Εικόνα 3.20: Η δομή της εφαρμογής στην πλευρά του διακομιστή..... | 114 |
| Εικόνα 3.21: Οι τεχνολογίες που χρησιμοποιούνται στην πλευρά του πελάτη (client-side)..... | 115 |
| Εικόνα 3.22: Η δομή του πηγαίου κώδικα στην πλευρά του διακομιστή (server-side) | 117 |
| Εικόνα 4.1: Το logo της εφαρμογής | 119 |
| Εικόνα 4.2: Η αρχική σελίδα | 119 |
| Εικόνα 4.3: Η φόρμα δημιουργίας λογαριασμού | 120 |
| Εικόνα 4.4: Η φόρμα δημιουργίας λογαριασμού συμπληρωμένη | 120 |
| Εικόνα 4.5: Η φόρμα εισόδου στο λογαριασμό..... | 121 |
| Εικόνα 4.6: Η αρχική εικόνα που βλέπει ο χρήστης όταν μπαίνει στο λογαριασμό του | 122 |
| Εικόνα 4.7: Όταν ο χρήστης επιλέγει ένα αγρόκτημα (“Select field”) ενεργοποιείται η δυνατότητα επιλογής επέμβασης ψεκασμού (“Select operation”) | 122 |
| Εικόνα 4.8: Πάνω δεξιά εμφανίζεται το μενού μέσω του οποίου μπορεί να περιηγηθεί ο χρήστης..... | 123 |
| Εικόνα 4.9: Η επιλογή του πίνακα με τα δεδομένα | 123 |
| Εικόνα 4.10: Η επιλογή προβολής χάρτη | 124 |
| Εικόνα 4.11: Η επιλογή προβολής του γραφήματος της ταχύτητας | 125 |
| Εικόνα 4.12: Ένα αποθηκευμένο αγροτεμάχιο | 125 |
| Εικόνα 4.13: Η φόρμα σχεδίασης και αποθήκευσης ενός αγροτεμαχίου | 126 |
| Εικόνα 4.14: Ένα παράδειγμα συμπλήρωσης της φόρμας ψηφιοποίησης αγροτεμαχίου .. | 127 |
| Εικόνα 4.15: Η 1 ^η επιλογή της κατηγορίας “Points” το “Coordinates of a point” | 128 |
| Εικόνα 4.16: Το “Coordinates of a point” αφού έχει επιλεγεί το “My Location” και έχουν επιλεγεί δύο σημεία πάνω στο χάρτη εμφανίζει τις συντεταγμένες τους..... | 128 |
| Εικόνα 4.17: Η επιλογή “Distance between two points” | 129 |
| Εικόνα 4.18: Η επιλογή “Middle point”..... | 129 |

| | |
|---|-----|
| Εικόνα 4.19: Η επιλογή “Buffer zone” | 130 |
| Εικόνα 4.20: Η επιλογή “Length” | 130 |
| Εικόνα 4.21: Η επιλογή “Area” | 131 |
| Εικόνα 4.22: Η επιλογή “Centroid” | 131 |
| Εικόνα 4.23: Η φόρμα καταχώρησης των στοιχείων του ψεκασμού | 132 |
| Εικόνα 4.24: Η φόρμα καταχώρησης των στοιχείων του ψεκασμού μετά τη συμπλήρωσή της | 133 |
| Εικόνα 4.25: Η λειτουργία καταγραφής των χωρικών δεδομένων | 133 |
| Εικόνα 4.26: Η καταγραφή ενός ψεκασμού σε λειτουργία | 134 |
| Εικόνα 4.27: Η επιλογές αποθήκευσης των δεδομένων της καταγραφής | 134 |
| Εικόνα 4.28: Η φόρμα μεταφοράς αρχείου στο διακομιστή (αρχικά)..... | 135 |
| Εικόνα 4.29: Η φόρμα μεταφοράς αρχείου στο διακομιστή (μετά από τις επιλογές του χρήστη) | 135 |
| Εικόνα 4.30: Κωδικός 401 - Unauthorized | 136 |
| Εικόνα 4.31: Κωδικός 404 - Page not Found | 137 |
| Εικόνα 4.32: Κωδικός 405 - Method not allowed | 137 |
| Εικόνα 4.33: Κωδικός 500 - Internal Server Error | 137 |
| Εικόνα 4.34: Το μήνυμα της επιτυχούς δημιουργίας λογαριασμού και η ανακατεύθυνση στη φόρμα εισόδου | 138 |
| Εικόνα 4.35: Το μήνυμα που εμφανίζεται κατά τη λάθος εισαγωγή στοιχείων | 138 |
| Εικόνα 4.36: Το μήνυμα που εμφανίζεται όταν υπάρχει λανθασμένη εισαγωγή στοιχείων..... | 139 |
| Εικόνα 4.37: Νέος χρήστης χωρίς καταχωρημένη έκταση..... | 140 |
| Εικόνα 4.38: Νέος χρήστης με καταχωρημένη έκταση, χωρίς όμως να έχει αποθηκεύσει κάποια επέμβαση ψεκασμού | 140 |
| Εικόνα 4.39: Το μήνυμα που εμφανίζεται για ένα αγροτεμάχιο που δεν έχει καμία επέμβαση ψεκασμού αποθηκευμένη..... | 140 |
| Εικόνα 4.40: Το μήνυμα μιας επιτυχούς αποθήκευσης αγροτεμαχίου..... | 141 |
| Εικόνα 4.41: Το μήνυμα για την περίπτωση αδυναμίας εύρεσης θέσης..... | 141 |
| Εικόνα 4.42: Το μήνυμα για την περίπτωση που δεν υπάρχει άδεια χρήσης του Geolocation | 142 |
| Εικόνα 4.43: Το μήνυμα για τις λάθος τιμές εισαγωγή συντεταγμένων | 142 |
| Εικόνα 4.44: Το μήνυμα που ειδοποιεί για την ύπαρξη του ονόματος του αγροτεμαχίου στη βάση δεδομένων | 143 |
| Εικόνα 4.45: Το μήνυμα για την έλλειψη συντεταγμένων των ορίων του αγροτεμαχίου ... | 143 |
| Εικόνα 4.46: Ο χρήστης δεν εισάγει κάποια τιμή στη φόρμα | 144 |
| Εικόνα 4.47: Ο χρήστης λανθασμένα εισάγει αντί για αριθμό ένα ή περισσότερα γράμματα | 144 |
| Εικόνα 4.48: Το μήνυμα για την περίπτωση της ελλιπούς ή λανθασμένης συμπλήρωσης των πεδίων της φόρμας | 145 |
| Εικόνα 4.49: Τρία διαφορετικά μηνύματα που επεξηγούν το λόγο της αδυναμίας εύρεσης τοποθεσίας..... | 146 |
| Εικόνα 4.50: Αδυναμία εύρεσης της θέσης κατά τη διάρκεια του ψεκασμού..... | 146 |
| Εικόνα 4.51: Το μήνυμα της επιτυχούς αποθήκευσης ενός ψεκασμού | 147 |
| Εικόνα 4.52: Αδυναμία σύνδεσης με το διακομιστή | 147 |
| Εικόνα 4.53: Κακής ποιότητας σύνδεση στο διαδίκτυο..... | 147 |
| Εικόνα 4.54: Το μήνυμα της επιτυχούς μεταφοράς του αρχείου..... | 148 |
| Εικόνα 4.55: Το μήνυμα λάθους τύπου επέκτασης αρχείου | 148 |
| Εικόνα 4.56: Μήνυμα λάθους μορφής αρχείου | 149 |

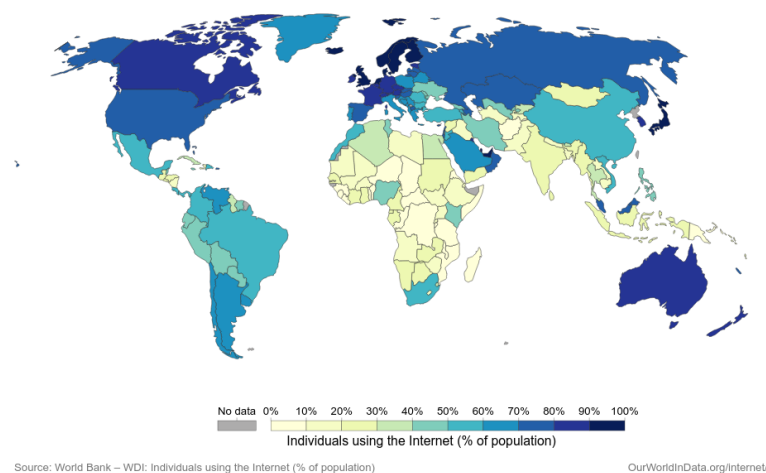
| | |
|--|-----|
| Εικόνα 4.57: Προβλήματα με της σύνδεση στο διαδίκτυο | 149 |
| Εικόνα 4.58: Το μήνυμα που εμφανίζεται στο χρήστη μετά την έξοδό του από το λογαριασμό | 150 |
| Εικόνα 4.59: Ένα δείγμα της γραμματοσειρά Roboto από την εφαρμογή..... | 151 |
| Εικόνα 4.60: Η γραμματοσειρά Nunito | 151 |
| Εικόνα 4.61: Η γραμματοσειρά Michroma | 151 |
| Εικόνα 4.62: Η γραμματοσειρά Courgette..... | 151 |
| Εικόνα 4.63: Οι κωδικοί ταιριάζουν μεταξύ τους | 152 |
| Εικόνα 4.64: Οι κωδικοί δεν ταιριάζουν μεταξύ τους..... | 152 |
| Εικόνα 4.65: Οδηγίες για την εύρεση της τοποθεσίας του αγροτεμαχίου..... | 152 |
| Εικόνα 4.66: Περιγραφές των στοιχείων του γραφήματος της ταχύτητας | 153 |
| Εικόνα 4.67: Η οδηγίες για την επιλογή “Distance between two points” | 153 |
| Εικόνα 4.68: Οι τρεις κύκλοι διαρκώς αυξομειούμενης διαμέτρου | 154 |
| Εικόνα 4.69: Το παλλόμενο μήνυμα-προειδοποίηση για τον έλεγχο της κατάστασης του GPS της συσκευής..... | 155 |
| Εικόνα 4.70: Το μήνυμα για το κουμπί “Check Location” | 155 |
| Εικόνα 4.71: Έλεγχος της σύνδεσης με το διακομιστή | 156 |
| Εικόνα 4.72: Το μήνυμα για τον τρόπο ενεργοποίησης του κουμπιού “Upload” | 156 |
| Εικόνα 4.73: Η αρχική σελίδα | 157 |
| Εικόνα 4.74: Η φόρμα εισόδου στο λογαριασμό του χρήστη | 157 |
| Εικόνα 4.75: Η φόρμα εγγραφής χρήστη..... | 158 |
| Εικόνα 4.76: Η σελίδα παρουσίασης των επεμβάσεων ψεκασμού (Operations)..... | 158 |
| Εικόνα 4.77: Το μενού της εφαρμογής | 158 |
| Εικόνα 4.78: Ο πίνακας με τα δεδομένα του ψεκασμού | 159 |
| Εικόνα 4.79: Η προβολή του χάρτη μιας επέμβασης ψεκασμού | 159 |
| Εικόνα 4.80: Η προβολή του γραφήματος της ταχύτητας ενός ψεκασμού..... | 160 |
| Εικόνα 4.81: Τα στοιχεία ενός αγροκτήματος..... | 160 |
| Εικόνα 4.82: Η φόρμα ψηφιοποίησης ενός αγροκτήματος..... | 161 |
| Εικόνα 4.83: Η επιλογή Spatial tools..... | 161 |
| Εικόνα 4.84: Το εργαλείο “Coordinates of a point” | 162 |
| Εικόνα 4.85: Η φόρμα της επιλογής Upload data | 162 |
| Εικόνα 4.86: Τα αποτελέσματα του ελέγχου των ρυθμίσεων SSL του διακομιστή..... | 170 |
| Εικόνα 4.87: Τα αντίστοιχα αποτελέσματα του ελέγχου μιας δημοφιλούς ιστοσελίδας | 170 |

1 Εισαγωγή

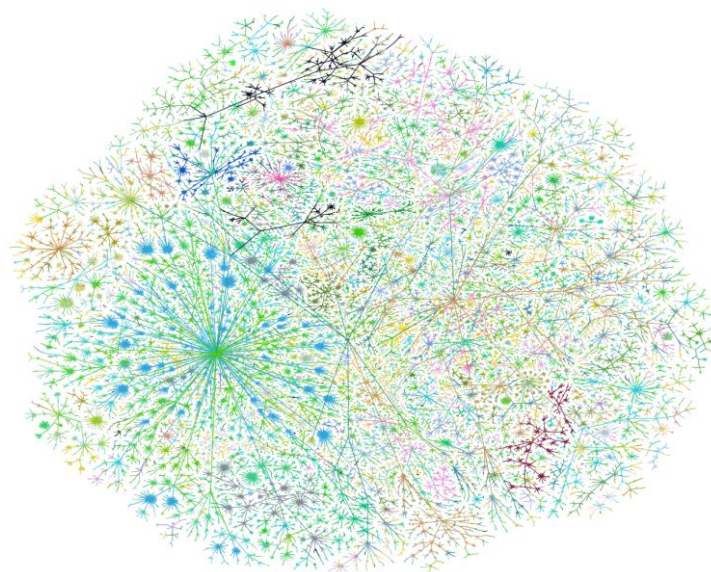
1.1 Διαδίκτυο και Παγκόσμιος Ιστός

1.1.1 Διαδίκτυο (Internet)

Με τον γενικότερο όρο διαδίκτυο εννοούμε ένα δίκτυο από δίκτυα, για παράδειγμα τέσσερα τοπικά δίκτυα μπορούν να συνδεθούν μεταξύ τους σχηματίζοντας ένα διαδίκτυο. Με τον ειδικότερο όρο Διαδίκτυο (Internet) αναφερόμαστε σε ένα παγκόσμιο σύστημα διασυνδεδεμένων δικτύων υπολογιστών. Ουσιαστικά το Διαδίκτυο αυτό καθ' αυτό δεν είναι ένα πραγματικό δίκτυο, αλλά μια αχανής συλλογή από διαφορετικά δίκτυα τα οποία χρησιμοποιούν κοινά πρωτόκολλα, ασφαλώς καθορισμένα από αναγνωρισμένους οργανισμούς.



Εικόνα 1.1: Ποσοστό του πληθυσμού κάθε χώρας που κάνει χρήση του Διαδικτύου



Εικόνα 1.2: Αναπαράσταση του Διαδικτύου από το Internet Mapping Project των Bell Labs
(www.cheswick.com/ches/map/gallery/index.html)

1.1.2 Παγκόσμιος Ιστός (World Wide Web)

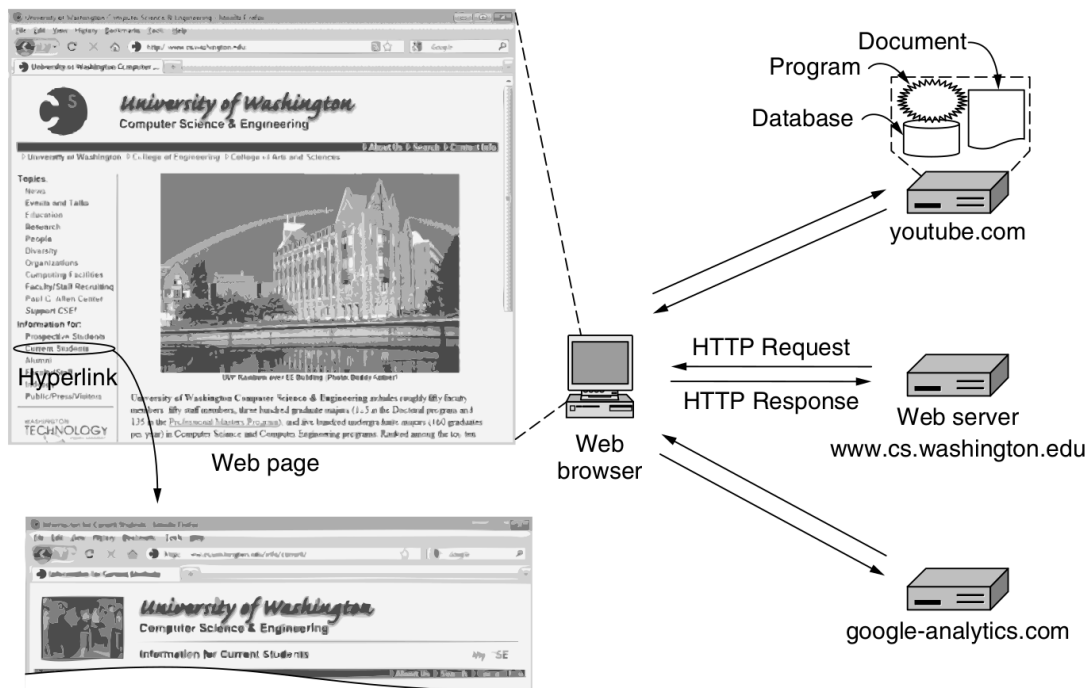
1.1.2.1 Γενικά

Ο Ιστός, όπως συνήθως ονομάζεται ο Παγκόσμιος Ιστός, είναι μια αρχιτεκτονική για την προσπέλαση διασυνδεδεμένων εγγράφων τα οποία κατανέμονται σε εκατομμύρια μηχανήματα σε ολόκληρο το Internet. Ο Ιστός ξεκίνησε από το **CERN** (το Ευρωπαϊκό Κέντρο Πυρηνικής Έρευνας) το 1989. Το 1994 το CERN και το MIT (Massachusetts Institute of Technology) υπέγραψαν τη συμφωνία ίδρυσης της Κοινοπραξίας Παγκόσμιου Ιστού (World Wide Web Consortium – **W3C**), ο οργανισμός W3C είναι αφιερωμένος στην περαιτέρω ανάπτυξη του Ιστού, την τυποποίηση των πρωτοκόλλων και την ενθάρρυνση της διαλειτουργικότητας μεταξύ διαφόρων τοποθεσιών (Tanenbaum & Wetherall, 2011).

1.1.2.2 Επισκόπηση της Αρχιτεκτονικής

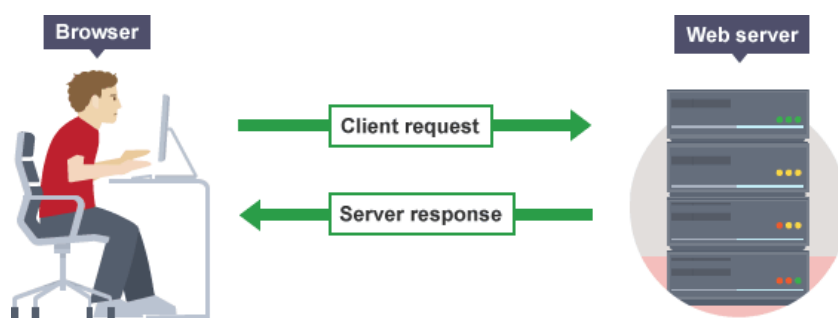
Από την πλευρά του χρήστη, ο Ιστός αποτελείται από μια απεριόριστη, παγκόσμια συλλογή περιεχομένου με τη μορφή ιστοσελίδων (**Web pages**). Κάθε σελίδα μπορεί να περιέχει συνδέσμους προς άλλες σελίδες που μπορεί να βρίσκονται σε οποιοδήποτε σημείο στον κόσμο. Το χαρακτηριστικό να δείχνει η κάθε σελίδα σε μία άλλη, ονομάζεται υπερκείμενο (**hypertext**). Οι σελίδες εμφανίζονται στο χρήστη μέσω ενός προγράμματος που ονομάζεται φυλλομετρητής (**browser**). Μερικοί δημοφιλείς φυλλομετρητές είναι ο Firefox και ο Chrome. Ο φυλλομετρητής προσκομίζει τη σελίδα, ερμηνεύει το περιεχόμενο και παρουσιάζει τη σελίδα, κατάλληλα μορφοποιημένη στην οθόνη του υπολογιστή. Τα τμήματα κειμένου, εικονιδίων, εικόνων κ.λπ. που αναφέρονται με άλλες σελίδες ονομάζονται υπερσύνδεσμοι (**hyperlinks**) (Tanenbaum & Wetherall, 2011).

Στην Εικόνα 2.3 παρουσιάζεται το βασικό μοντέλο λειτουργίας του Ιστού. Ο φυλλομετρητής εμφανίζει μια ιστοσελίδα στον υπολογιστή του πελάτη. Η κάθε σελίδα εμφανίζεται μετά την αποστολή μιας αίτησης σε έναν ή περισσότερους διακομιστές, οι οποίοι απαντούν με τα περιεχόμενα της σελίδας. Αυτό το πρωτόκολλο αίτησης-απάντησης για την προσκόμιση σελίδων βασίζεται σε ένα πρωτόκολλο κειμένου που λειτουργεί μέσω του TCP (Transmission Control Protocol). Αποκαλείται Πρωτόκολλο Μεταφοράς Υπερ-Κειμένου (HyperText Transfer Protocol – **HTTP**). Το περιεχόμενο μπορεί να είναι ένα απλό έγγραφο που διαβάζεται από μια συσκευή αποθήκευσης ή να είναι το αποτέλεσμα ενός ερωτήματος προς τη βάση δεδομένων. Μια σελίδα λέμε ότι είναι **στατική** αν πρόκειται για ένα έγγραφο το οποίο είναι ίδιο κάθε φορά που εμφανίζεται. Αντίθετα, αν παράγεται κατόπιν αιτήσεως από ένα πρόγραμμα ή περιέχει ένα πρόγραμμα τότε έχουμε να κάνουμε με μια **δυναμική** σελίδα (Tanenbaum & Wetherall, 2011).



Εικόνα 1.3: Η αρχιτεκτονική του Ιστού
(Tanenbaum & Wetherall, 2011)

Στη συνέχεια θα εξετάσουμε την πλευρά του πελάτη και την πλευρά του διακομιστή, των οποίων η σχέση παρουσιάζεται στην Εικόνα 2.4. Από τη μια έχουμε τον χρήστη-πελάτη που στέλνει αιτήματα μέσω του φυλλομετρητή (Client request) του στο διακομιστή και αυτός με τη σειρά του στέλνει την ανάλογη απάντηση (Server response) με βάση την οποία εμφανίζει ο φυλλομετρητής το περιεχόμενο της σελίδας.



Εικόνα 1.4: Αναπαράσταση της αλληλεπίδρασης μεταξύ πελάτη (client) και διακομιστή (web server)

(www.bbc.co.uk/education/guides/zh4whyc/revision/7)

1.1.2.3 Η πλευρά του πελάτη

Ο χρήστης στην πλευρά του πελάτη αλληλοεπιδρά με τη σελίδα και το διακομιστή μέσω του φυλλομετρητή του. Ο φυλλομετρητής στην ουσία είναι ένα πρόγραμμα το οποίο μπορεί να εμφανίζει μια ιστοσελίδα και να συλλέγει τα κλικ του ποντικιού πάνω σε αντικείμενα σελίδας. Όταν ο χρήστης επιλέγει ένα αντικείμενο, ο φυλλομετρητής ακολουθεί τον υπερσύνδεσμο και προσκομίζει την αντίστοιχη σελίδα. Για να γίνει εφικτός ο προσδιορισμός και ο εντοπισμός της σελίδας σε κάθε σελίδα αντιστοιχίζεται ένας Ενιαίος Εντοπιστής Πόρων (Uniform Resource Locator – **URL**), ο οποίος στην ουσία πρόκειται για το παγκόσμιο όνομα της σελίδας. Οι διευθύνσεις URL έχουν τρία μέρη: το πρωτόκολλο, το όνομα **DNS** (Domain Name System) του μηχανήματος όπου βρίσκεται η σελίδα και τη διαδρομή που προσδιορίζει με μοναδικό τρόπο τη συγκεκριμένη σελίδα. Για παράδειγμα, έχουμε τη διεύθυνση URL <https://www.aua.gr/index.html>. Η διεύθυνση έχει τρία μέρη: το όνομα του πρωτοκόλλου (https), το όνομα DNS του μηχανήματος στο οποίο βρίσκεται η σελίδα (www.aua.gr) και το όνομα της διαδρομής (index.html) (Tanenbaum & Wetherall, 2011).

Όταν ένας χρήστης πατά πάνω σε κάποιον υπερσύνδεσμο, ο φυλλομετρητής εκτελεί μια ακολουθία βημάτων για να προσκομίσει τη σελίδα του συνδέσμου. Στη συνέχεια περιγράφονται τα βήματα που ακολουθούνται κατά την επιλογή αυτού του συνδέσμου (Tanenbaum & Wetherall, 2011):

1. Ο φυλλομετρητής καθορίζει τη διεύθυνση URL.
2. Ο φυλλομετρητής ζητά από το DNS τη διεύθυνση IP για το διακομιστή www.aua.gr.
3. Το DNS απαντά με τη διεύθυνση 143.233.204.102.
4. Ο φυλλομετρητής εγκαθιδρύει μια TCP σύνδεση με τη θύρα 80 στη διεύθυνση 143.233.204.102.
5. Ο φυλλομετρητής στέλνει μια αίτηση που ζητά το αρχείο /index.html.
6. Ο διακομιστής www.aua.gr στέλνει το αρχείο ως απάντηση HTTP.
7. Αν η σελίδα περιέχει και άλλες διευθύνσεις URL που θα πρέπει να εμφανιστούν, ο φυλλομετρητής τις προσκομίζει επαναλαμβάνοντας την ίδια διαδικασία.
8. Ο φυλλομετρητής εμφανίζει τη σελίδα /index.html.
9. Η σύνδεση TCP αποδεσμεύεται αν δεν υπάρξουν σε σύντομο διάστημα άλλες αιτήσεις για τους ίδιους διακομιστές.

1.1.2.4 Η πλευρά του διακομιστή

Ο διακομιστής Ιστού είναι εκείνος που δέχεται το όνομα ενός αρχείου για αναζήτηση και επιστροφή μέσω του δικτύου. Και στις δύο περιπτώσεις τα βήματα που πρέπει να εκτελέσει ο διακομιστής είναι τα εξής (Tanenbaum & Wetherall, 2011):

1. Αποδοχή μιας σύνδεσης TCP από έναν πελάτη (φυλλομετρητή).
2. Λήψη της διαδρομής για τη σελίδα, που είναι το όνομα αρχείου προς αναζήτηση.
3. Λήψη του αρχείου από το δίσκο του μηχανήματος.
4. Επιστροφή των περιεχομένων του αρχείου στον πελάτη.

5. Αποτελέσματα της σύνδεσης TCP.

1.2 Βάσεις Δεδομένων

1.2.1 Εισαγωγή

Μια Βάση Δεδομένων δεν είναι τίποτα παραπάνω από ένα σύνολο σχετιζόμενων πληροφοριών. Ένας τηλεφωνικός κατάλογος για παράδειγμα, είναι μια βάση δεδομένων από ονόματα, τηλεφωνικούς αριθμούς και διευθύνσεις από ανθρώπους που ζουν σε μια συγκεκριμένη περιοχή. Επειδή, ένα σύστημα βάσης δεδομένων αποθηκεύει τα δεδομένα σε ηλεκτρονική μορφή αντί σε χαρτί, είναι ικανό να ανακτήσει τα δεδομένα πιο γρήγορα, να κατατάξει τα δεδομένα με πολλούς τρόπους και να διανείμει πρόσφατα ενημερωμένη πληροφορία στον χρήστη (Beaulieu A., 2009).

1.2.2 Το Σχεσιακό Μοντέλο

Το 1970 ο E. F. Codd από το ερευνητικό εργαστήριο της IBM έκανε μια δημοσίευση με τίτλο “A Relational Model of Data for Large Shared Data Banks” στην οποία πρότεινε τα δεδομένα να αναπαρίστανται ως σύνολα από πίνακες. Αντί να χρησιμοποιούνται δείκτες για να ανακατευθυνόμαστε μεταξύ των συνδεδεμένων οντοτήτων, μπορούν να χρησιμοποιηθούν τα πλεονάζοντα δεδομένα για να συνδέσουν εγγραφές μεταξύ διαφορετικών πινάκων (Beaulieu A., 2009).

Στην τυπική ορολογία του σχεσιακού μοντέλου, μια γραμμή λέγεται **πλειάδα**, η επικεφαλίδα μιας στήλης ονομάζεται **γνώρισμα** και ολόκληρος ο πίνακας λέγεται **σχέση**. Ο τύπος δεδομένων που περιγράφει τους τύπους τιμών που μπορούν να εμφανιστούν σε μια στήλη λέγεται **πεδίο ορισμού** (Elmasri & Navathe, 2012).

Κάθε πίνακας σε μια σχεσιακή βάση δεδομένων περιέχει ένα στοιχείο το οποίο ταυτοποιεί μοναδικά μια γραμμή μαζί με πρόσθετες πληροφορίες που χρειάζονται για να περιγράψουν πλήρως αυτή την οντότητα, αυτό το στοιχείο ονομάζεται **πρωτεύον κλειδί**. Μερικοί από τους πίνακες περιέχουν και στοιχεία για τη καθοδήγηση προς άλλους πίνακες. Αυτές οι στήλες ονομάζονται **ξένα κλειδιά**. Η διαδικασία με την οποία σχεδιάζεται μια βάση δεδομένων με σκοπό να βεβαιωθεί ότι κάθε ανεξάρτητο κομμάτι πληροφορίας είναι μόνο σε ένα σημείο είναι γνωστή ως **κανονικοποίηση** (Beaulieu A., 2009).

1.3 Γεωγραφικά Πληροφοριακά Συστήματα

1.3.1 Γενικά

Ένα Γεωγραφικό Πληροφοριακό Σύστημα (ΓΠΣ, Geographical Information System – GIS) είναι ένα υπολογιστικό σύστημα που έχει ως σκοπό την είσοδο, το χειρισμό, την αποθήκευση και την έξοδο ψηφιακών χωρικών δεδομένων. Πιο γενικά μπορούμε να πούμε ότι είναι ένα ψηφιακό σύστημα για την λήψη, τη διαχείριση, την ανάλυση και την οπτικοποίηση των χωρικών δεδομένων με σκοπό την εκμετάλλευσή τους για το σχεδιασμό, την οργάνωση και την επίβλεψη του φυσικού και κοινωνικοοικονομικού περιβάλλοντος. Από τεχνική σκοπιά ένα ΓΠΣ αποτελείται από ένα σύστημα για την εισαγωγή δεδομένων σε διανυσματική, πλεγματική και αλφαριθμητική μορφή, μια κεντρική μονάδα επεξεργασίας (Central Processing Unit – CPU) που περιέχει τα προγράμματα για την επεξεργασία, την αποθήκευση και την ανάλυση των δεδομένων (Konecny, 2014).

1.3.2 Χαρτογραφικές Προβολές

Οι χαρτογραφικές προβολές χρησιμοποιούνται με σκοπό την ακριβή αναπαράσταση της σφαιρικής επιφάνειας της γης πάνω σε επίπεδους χάρτες. Ένας χάρτης που δεν χρησιμοποιεί κάποιο προβολικό σύστημα έχει περιορισμούς στην ακρίβεια με την οποία αναπαριστά τη γεωγραφική πληροφορία. Λόγω της καμπυλότητας της επιφάνειας της γης, οι χαρτογραφίες μεγάλων περιοχών χωρίς τις προβολές θα παρουσίαζαν μεγάλη ασάφεια. Η ανάπτυξη των προβολικών συστημάτων ξεκίνησε αρχικά από τη ναυσιπλοΐα, αυτό συνέβη επειδή οι ναυτικοί χρειαζόντουσαν επίπεδους χάρτες, επομένως οι προβολές ήταν ιδιαίτερα σημαντικές σε αυτή την περίπτωση. Ένα αρκετά κοινό προβολικό σύστημα είναι το Μερκατορικό, αν και υπάρχουν πολλά άλλα προβολικά συστήματα στο Μερκατορικό υπάρχει το χαρακτηριστικό ότι οι γραμμές μιας συνεχούς κατεύθυνσης είναι ευθείες (Harvey, 2008).

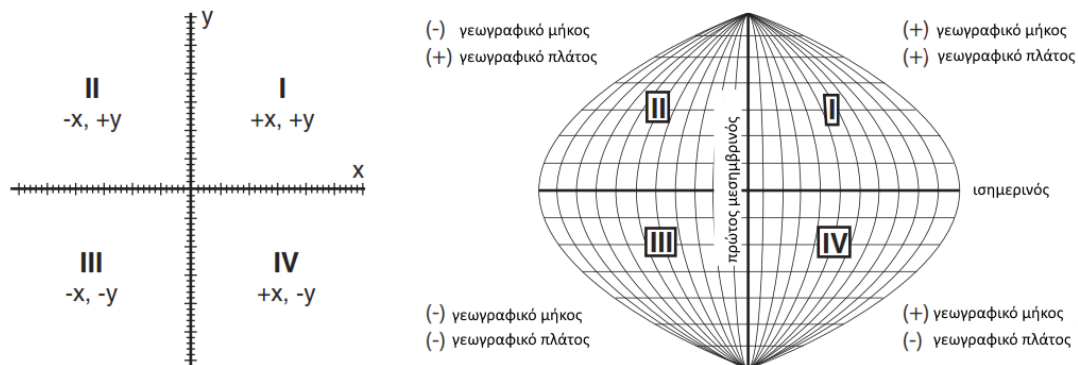
1.3.3 Γεωγραφικές Συντεταγμένες

Το Καρτεσιανό σύστημα συντεταγμένων χρησιμοποιεί τις x και y συντεταγμένες για να προσδιορίσει την τοποθεσία των σημείων πάνω στο επίπεδο. Τα τέσσερα τεταρτημόρια προσδιορίζουν τη θέση των σημείων ανάλογα με το πρόσημο των συντεταγμένων (εικόνα 1.5).

Στο σφαιροειδές σύστημα συντεταγμένων που χρησιμοποιείται στη Γη, το γεωγραφικό πλάτος αντιστοιχεί στις y συντεταγμένες και το γεωγραφικό μήκος αντιστοιχεί στις x συντεταγμένες (εικόνα 1.5).

Το σύστημα του γεωγραφικού μήκους και πλάτους είναι μόνο ένα από τα πολλά συστήματα συντεταγμένων που χρησιμοποιούνται για να προσδιορίσουν μια τοποθεσία πάνω στη Γη. Τα άλλα συστήματα χρησιμοποιούνται για μικρότερες

περιοχές και θεωρούν τη Γη ως επίπεδη. Παρότι αυτό δημιουργεί σφάλμα, το σφάλμα αυτό είναι τόσο μικρό που για κάποιες εφαρμογές δεν θεωρείται σημαντικό. Ένα από αυτά τα συστήματα είναι το Universal Transverse Mercator (UTM), αυτά τα συστήματα χρησιμοποιούνται όπου είναι δύσκολο να βρεθεί το γεωγραφικό μήκος και πλάτος. Με την άνοδο της χρήσης των Παγκόσμιων Δορυφορικών Συστημάτων Πλοήγησης (Global Navigation Satellite System - GNSS) σχεδόν όλες οι περιοχές ορίζονται με το γεωγραφικό τους μήκος και πλάτος (Peterson, 2014).



Εικόνα 1.5: Το Καρτεσιανό και το σφαιροειδές σύστημα συντεταγμένων.

(Peterson, 2014)

1.3.4 Παγκόσμια Δορυφορικά Συστήματα Πλοήγησης

Ο όρος Παγκόσμια Δορυφορικά Συστήματα Πλοήγησης (Global Navigation Satellite System - GNSS) αναφέρεται στα συστήματα που χρησιμοποιούνται για τον προσδιορισμό της θέσης πάνω στη Γη. Επειδή το Global Positioning System (GPS) είναι το πρώτο GNSS σύστημα που αναπτύχθηκε έχει επικρατήσει λανθασμένα να χρησιμοποιείται για να περιγράψει όλα τα GNSS. Τα κύρια GNSS είναι το Αμερικανικής προέλευσης GPS, το Ρωσικής προέλευσης GLONASS, το Κινέζικης προέλευσης BeiDou και το Ευρωπαϊκής προέλευσης Galileo. Ενώ τα τρία πρώτα είναι πλήρως λειτουργικά το Galileo αναμένεται να καταστεί πλήρως λειτουργικό το 2020.

1.3.4.1 Global Positioning System (GPS)

Το GPS όπως και το Διαδίκτυο είναι ένα προϊόν του Ψυχρού Πολέμου. Το Υπουργείο Άμυνας των Ηνωμένων Πολιτειών Αμερικής ήθελε ένα σύστημα το οποίο θα χρησιμοποιούταν από τα υποβρύχια με σκοπό να πλήξουν στόχους στη Σοβιετική Ένωση. Το αρχικό σύστημα ονομαζόταν NAVSTAR GPS και χρησιμοποιούταν μόνο για στρατιωτική χρήση. Για δημόσια χρήση το GPS έγινε πλήρως λειτουργικό το 1995. Το 2000 αποφασίστηκε η αφαίρεση της επιλεκτικής ακρίβειας, όπου για τους πολίτες η ακρίβεια του συστήματος ήταν σημαντικά μειωμένη.

Το GPS αποτελείται από δορυφόρους που βρίσκονται σε τροχιά γύρω από τη Γη σε ένα ύψος 20.200 km μεταξύ 60°N and 60°S γεωγραφικό πλάτος. Ο κάθε δορυφόρος έχει 5,8 m μέγεθος και με τα ηλιακά πάνελ ζυγίζει περίπου ένα τόνο. Με την περιστροφή των δορυφόρων γύρω από τη Γη τους εξασφαλίζεται ότι τουλάχιστον 6 δορυφόροι μπορούν να χρησιμοποιηθούν από κάθε σημείο πάνω στη Γη (Peterson, 2014).

Η ακρίβεια του GPS στα κινητά τηλέφωνα βρίσκεται στα 5 m, όμως η Broadcom (κατασκευάστρια εταιρία chips) ανακοίνωσε το Σεπτέμβριο του 2017 ότι θα κατασκευάσει το chip BCM47755 το οποίο θα είναι το πρώτο ευρείας παραγωγής chip κινητού τηλεφώνου που θα έχει ακρίβεια 30 cm και θα καταναλώνει τη μισή ενέργεια σε σχέση με τα σημερινά. Το BCM47755 θα πρωτοεμφανιστεί σε κινητά τηλέφωνα που θα βγουν σε κυκλοφορία μέσα στο έτος 2018. Η βελτίωση της ακρίβειας του BCM47755 επιτυγχάνεται με την επικοινωνία με το δορυφόρο μέσω ενός νέου σήματος, του L5, το οποίο χρησιμοποιεί διαφορετική συχνότητα σε σχέση με τα μέχρι τώρα chips (Moore, 2017).

1.3.5 Χωρικά Καθορισμένη Ψηφιακή Γεωργία

Ο όρος Ψηφιακή Γεωργία αναφέρεται στην εξέλιξη από τη Γεωργία Ακριβείας προς μια γεωργία όπου τα αγροκτήματα θα είναι διασυνδεδεμένα και θα εκμεταλλεύονται όλα τα διαθέσιμα δεδομένα για τη βελτίωση της αποδοτικότητας της παραγωγής. Ένα σημαντικό μέρος της πληροφορίας που υπάρχει στα αγροκτήματα είναι τα χωρικά δεδομένα. Αυτό συμβαίνει διότι η γεωργία λαμβάνει χώρα σε κάποια περιοχή, επομένως είναι συνυφασμένη με τη χωρική διάσταση. Για το λόγο αυτό η γεωργική παραγωγή παρουσιάζει χωρική παραλλακτικότητα. Συνεπώς, είναι σημαντικό κατά τη λήψη αποφάσεων, σχετικά με την άσκηση των γεωργικών εργασιών, να υπάρχει εκτενής χωρική πληροφορία.

Οι τεχνολογίες που χρησιμοποιούνται από τη Ψηφιακή Γεωργία είναι οι επίγειοι και εναέριοι αισθητήρες, τα ασύρματα δίκτυα επικοινωνιών, το Διαδίκτυο, το Cloud computing και η Τηλεματική. Επίσης, κατά το στάδιο της επεξεργασίας των δεδομένων χρησιμοποιείται λογισμικό το οποίο υλοποιεί χωρικούς και μη χωρικούς αλγορίθμους. Στη συνέχεια μπορούν να ληφθούν αποφάσεις και να δοθούν οδηγίες στα μηχανήματα υψηλής ακριβείας για τη χωρικά διαφοροποιημένη λίπανση, ζιζανιοκτονία, άρδευση και συγκομιδή (Αλεξανδρή, 2018).

Η Ψηφιακή Γεωργία έχει το χαρακτηριστικό ότι μπορεί να υλοποιηθεί πριν από την εγκατάσταση μιας καλλιέργειας, αλλά και μετά τη συγκομιδή των προϊόντων. Πριν την εγκατάσταση της καλλιέργειας, μπορεί να εφαρμοστεί μέσω της επιλογής της πλέον κατάλληλης γεωργικής χρήσης για μια συγκεκριμένη έκταση. Μετά τη συγκομιδή των προϊόντων, είναι εφικτή η ενημέρωση του καταναλωτή για την ταυτότητα του προϊόντος μέσω μιας ετικέτας γραμμωτού κώδικα τύπου QR (Αλεξανδρή, 2018).

1.4 Πληροφοριακά Συστήματα στη Γεωργία

1.4.1 Εισαγωγή

Με βάση τους Sorensen et al. (2010) τα διοικητικά καθήκοντα στον τομέα των αροτραίων καλλιεργειών αλλάζουν σε κάτι καινούργιο, το οποίο απαιτεί αυξημένη προσοχή στο ευρύτερο περιβάλλον. Μεταξύ άλλων, αυτή η αλλαγή προκύπτει από εξωτερικούς παράγοντες (όπως οι κυβερνήσεις και η κοινή γνώμη) οι οποίοι εξασκούν πίεση στον αγροτικό τομέα για τη μεταφορά του επίκεντρου της παραγωγής από την ποσότητα στην ποιότητα και την αειφορία. Αυτή η αλλαγή στις συνθήκες εκτέλεσης των διοικητικών καθηκόντων επέβαλε την εισαγωγή πιο προχωρημένων πληροφοριακών συστημάτων για να επιτευχθεί η συμβατότητα με τους περιορισμούς και τα πρότυπα των σύγχρονων παραγωγικών κατευθύνσεων.

Στα αγροκτήματα έχει αρχίσει να υπάρχει χρήση των Ηλεκτρονικών Υπολογιστών μαζί με το ανάλογο λογισμικό. Πολλοί γεωργοί εισήγαγαν σταδιακά τη χρήση των Παγκόσμιων Δορυφορικών Συστημάτων Πλοήγησης (Global Navigation Satellite System - GNSS) στα γεωργικά τους μηχανήματα και τη χρήση των Γεωγραφικών Πληροφοριακών Συστημάτων (Geographical Information Systems – GIS) για την εξαγωγή συμπερασμάτων από τα χωρικά δεδομένα.

Σύμφωνα με τους Novkovic et al. (2015) επειδή σε μια γεωργική εκμετάλλευση παράγεται μια μεγάλη ποσότητα δεδομένων, για να μπορέσουν οι παραγωγοί να τα εκμεταλλευτούν αποδοτικά θα πρέπει να είναι σε θέση να εκπληρώσουν τις παρακάτω διεργασίες:

1. Συλλογή δεδομένων.
2. Επεξεργασία δεδομένων.
3. Παροχή δεδομένων.
4. Χρήση δεδομένων.

Για να πραγματοποιηθούν οι ανωτέρω διεργασίες χρησιμοποιείται μια ειδική περίπτωση Πληροφοριακού Συστήματος, το Σύστημα Υποστήριξης Λήψης Αποφάσεων – ΣΥΠ (Decision Support System – DSS) (Novkovic et al., 2015).

Σύμφωνα με τους Sorensen et al. (2010) μέχρι τώρα, οι παραγωγοί διαχειρίζονταν τον αυξημένο αυτό διοικητικό φόρτο χρησιμοποιώντας δεδομένα, χωρίς αυτά να έχουν δομηθεί με ένα λειτουργικό τρόπο, ο απώτερος στόχος αυτής της προσπάθειας ήταν να πάρουν ακριβείς αποφάσεις. Είναι συχνό φαινόμενο οι παραγωγοί να αντιμετωπίζουν μια υπερφόρτωση πληροφορίας, η οποία προκύπτει λόγω των διαφορετικών πηγών δεδομένων και της παρουσίας τους σε διαφορετικές μορφές. Η αυξημένη χρήση των υπολογιστών και του Διαδικτύου σε κάποιο βαθμό βελτίωσε την κατάσταση. Προσπάθειες για να βελτιωθεί αυτή η κατάσταση έγιναν με την εισαγωγή Πληροφοριακών Συστημάτων που έτρεχαν στο Διαδίκτυο. Παρ' όλα αυτά τέτοια συστήματα θα πρέπει να γίνουν πιο συνεργιστικά μέσω της αυτόματης λήψης

των δεδομένων και με την ενσωμάτωσή τους σε ένα Πληροφοριακό Σύστημα Διαχείρισης Αγροκτήματος – ΠΣΔΑ (Farm Management Information System – FMIS).

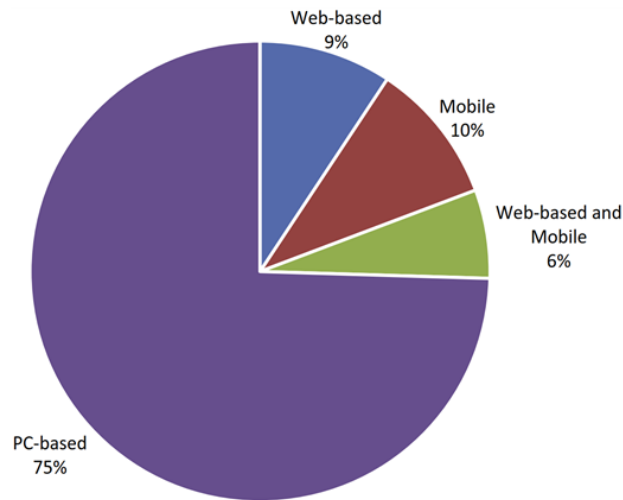
1.4.2 Πληροφοριακά Συστήματα Διαχείρισης Αγροκτήματος (ΠΣΔΑ)

Ένας ορισμός για το ΠΣΔΑ (Boehlje & Eidman, 1984) είναι ο εξής: ΠΣΔΑ είναι ηλεκτρονικά εργαλεία για τη συλλογή δεδομένων και την επεξεργασία τους με απώτερο σκοπό να αποδώσουν πληροφορία υψηλής αξίας για τη λήψη διοικητικών αποφάσεων.

Οι Sørensen et al. (2010) ορίσανε ένα ΠΣΔΑ ως ένα οργανωμένο σύστημα για τη συλλογή, την επεξεργασία, την αποθήκευση και τη διάδοση των δεδομένων σε μια μορφή χρήσιμη για τις διεργασίες και λειτουργίες ενός αγροκτήματος. Απαραίτητα στοιχεία ενός ΠΣΔΑ περιλαμβάνουν αγροτοκεντρικό σχεδιασμό, προσαρμοσμένη διεπαφή χρήστη, λειτουργίες αυτόματης επεξεργασίας δεδομένων, έμπειρη γνώση και προτιμήσεις χρήστη, τυποποιημένη επικοινωνία μεταξύ των δεδομένων και επεκτασιμότητα. Για να βελτιωθεί η λειτουργικότητά τους, δοκιμάστηκαν συστήματα διαχείρισης, δίκτυα δομών βάσεων δεδομένων και αρχιτεκτονικές λογισμικού. Στην πράξη, τα ΠΣΔΑ έχουν βελτιωθεί μέσω της ενσωμάτωσης νέων τεχνολογιών όπως οι διαδικτυακές εφαρμογές και οι εφαρμογές για κινητά τηλέφωνα.

Σύμφωνα με μια έρευνα που πραγματοποίησαν οι Fountas et al. (2015) οι λειτουργίες που βρίσκονται πιο συχνά σε εφαρμογές των ΠΣΔΑ είναι η διαχείριση των εργασιών του αγροκτήματος (63%), αναφορές (57%), οικονομικά (45%), διαχείριση πεδίου (40%), καταγραφή των εμπορευμάτων (38%), η διαχείριση των μηχανημάτων (28%) και η διαχείριση των ανθρώπινων πόρων (25%). Ακόμα, οι λιγότερο χρησιμοποιούμενες λειτουργίες είναι: η ιχνηλασιμότητα (19%), η διασφάλιση της ποιότητας (19%), οι πωλήσεις (18%) και οι καλύτερες πρακτικές (16%).

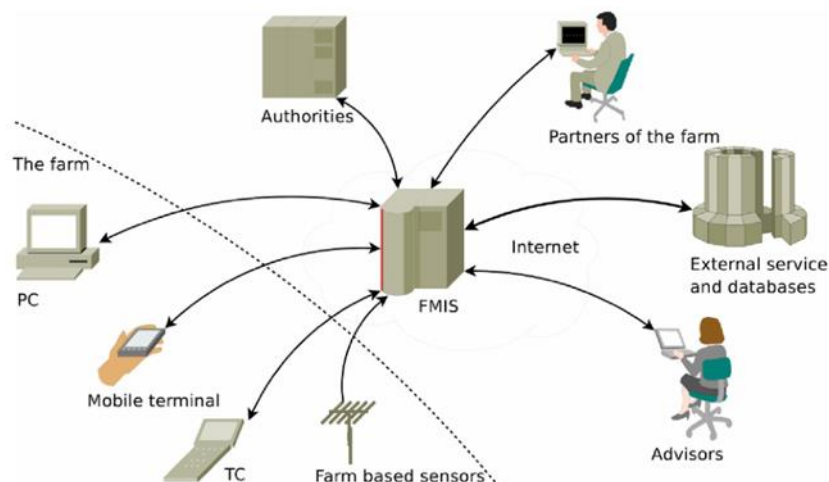
Ακόμα, οι Fountas et al. (2015) διαπίστωσαν μέσα από την ανάλυσή τους ότι το 75% των εφαρμογών ΠΣΔΑ είναι σχεδιασμένες για λειτουργία σε υπολογιστές, το 10% για λειτουργία σε κινητά τηλέφωνα και tablets, το 9% είναι εφαρμογές ιστού (web applications) και το 6% είναι για λειτουργία και σε κινητά και ως εφαρμογές ιστού.



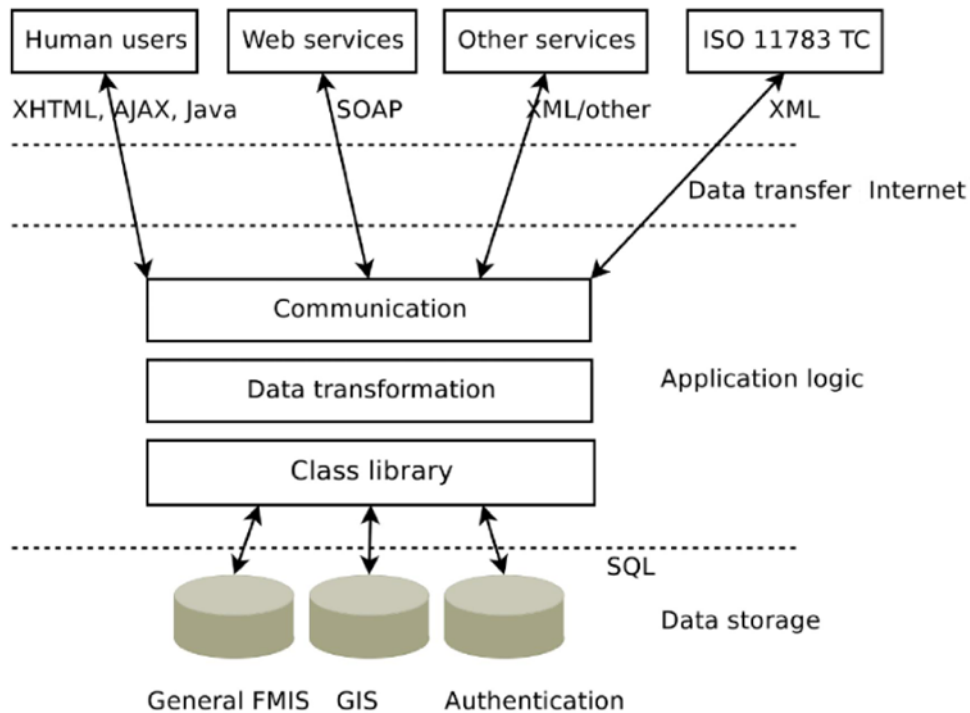
Εικόνα 1.6: Η κατανομή των ΠΣΔΑ ανά πλατφόρμα λειτουργίας σύμφωνα με τη μελέτη των Fountas et al. (2015)

Στην εικόνα 1.6 εμφανίζεται η αρχιτεκτονική ενός ΠΣΔΑ από την οπτική του απλού χρήστη. Αντίστοιχα, στην εικόνα 1.7 εμφανίζεται η αρχιτεκτονική από την πλευρά του υπεύθυνου ανάπτυξης – προγραμματιστή. Από την τεχνική σκοπιά η αρχιτεκτονική αναλύεται από κάτω προς τα πάνω, ξεκινώντας με την αποθήκευση των δεδομένων (Data storage) και συνεχίζοντας προς τη λογική της εφαρμογής (Application logic). Η λογική της εφαρμογής μπορεί να αναλυθεί περαιτέρω σε βιβλιοθήκη κλάσεων, επίπεδα μετασχηματισμού δεδομένων και επικοινωνίας (Data transfer) (Salami & Ahmadi, 2010).

Με βάση τους Salami & Ahmadi (2010) όλα τα δεδομένα αποθηκεύονται σε ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων - ΣΔΒΔ (Relational Database Management System – RDBMS). Οι τρεις βάσεις δεδομένων που φαίνονται στην εικόνα 1.7 είναι: 1) Βάση Πιστοποίησης, 2) Βάση Δεδομένων του Γεωγραφικού Πληροφοριακού Συστήματος και 3) Γενική Βάση Δεδομένου του ΠΣΔΑ.



Εικόνα 1.7: Η αρχιτεκτονική ενός ΠΣΔΑ από την οπτική ενός απλού χρήστη
(Salami & Ahmadi, 2010)

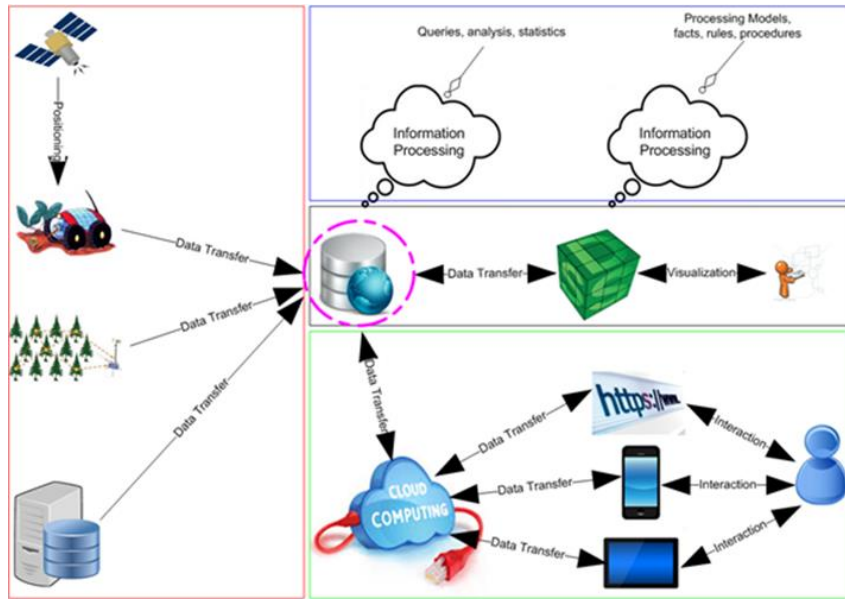


Εικόνα 1.8: Η αρχιτεκτονική ενός ΠΣΔΑ από την οπτική του υπεύθυνου ανάπτυξης
(Salami & Ahmadi, 2010)

1.4.3 Υλοποιήσεις

Στη συνέχεια ακολουθεί η αναφορά υλοποιήσεων πληροφοριακών συστημάτων στη γεωργία.

Οι Tsiropoulos & Fountas (2015) ανέπτυξαν ένα ΠΣΔΑ για την σε πραγματικό χρόνο καταγραφή των δεδομένων, την ανάλυση και την ενσωμάτωση κανόνων λήψης αποφάσεων για συγκεκριμένες εργασίες πεδίου. Το ΠΣΔΑ που αναπτύχθηκε αποτελείται από τρία βασικά στοιχεία: Τη γεωβάση του συστήματος, την εφαρμογή ιστού και τη εφαρμογή για κινητά τηλέφωνα με Λειτουργικό Σύστημα Android. Μέσω της εφαρμογής του Android είναι εφικτή η αποθήκευση, μεταξύ άλλων, των ορίων των αγροκτημάτων, των θέσεων των δένδρων και των διαδρομών μέσα στο αγρόκτημα. Επίσης, μέσω της εφαρμογής ιστού υπάρχει πρόσβαση στα δεδομένα και σε αναφορές του συστήματος.



Εικόνα 1.9: Η ροή των διεργασιών του ΠΣΔΑ για τα σπρωφόρα δέντρα (Tsiropoulos & Fountas, 2015)



Εικόνα 1.10: Εισαγωγή των δεδομένων των δένδρων (Tsiropoulos & Fountas, 2015)

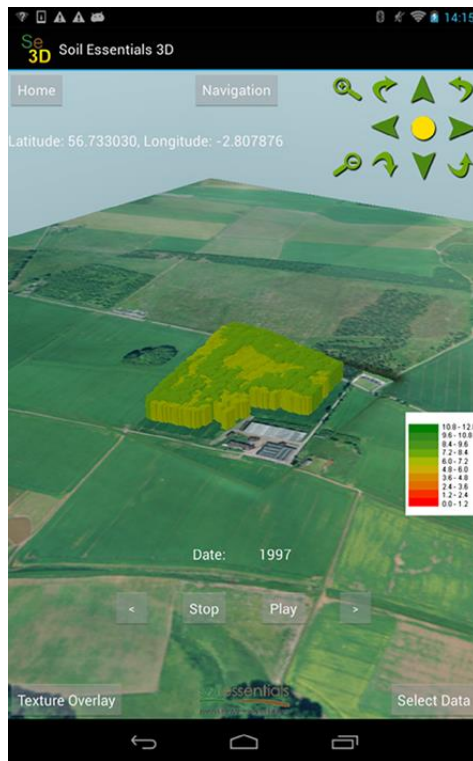


Εικόνα 1.11: Ο χάρτης της θερμοκρασίας της κόμης όπως παρουσιάζεται στην εφαρμογή ιστού

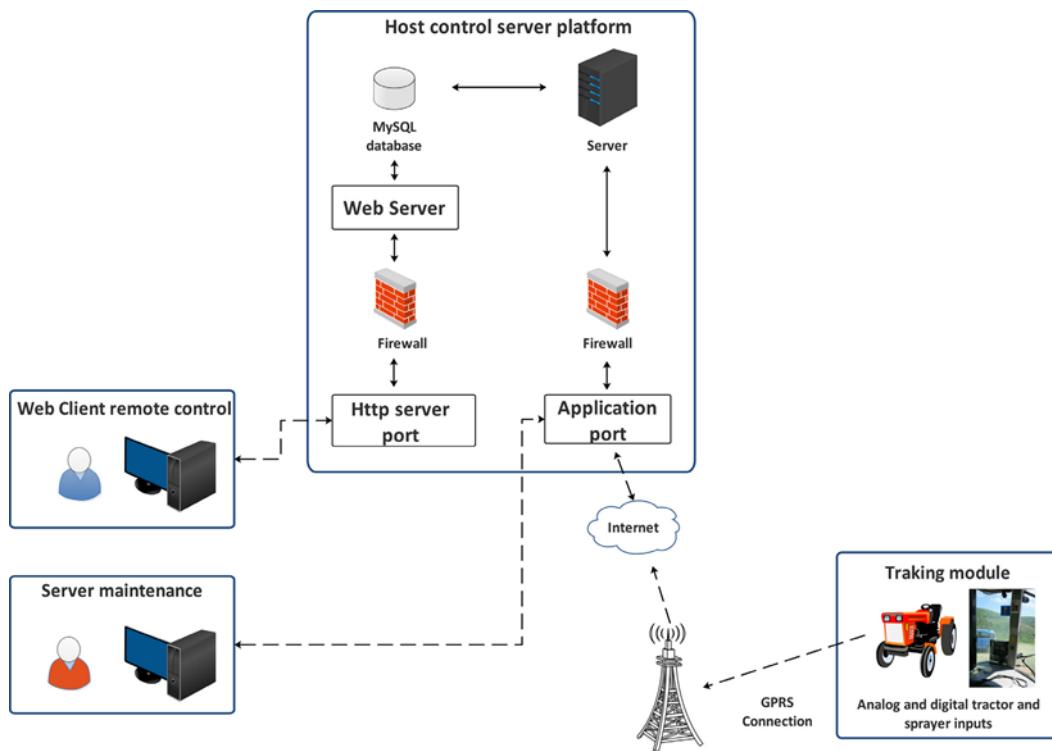
(Tsiropoulos & Fountas, 2015)

Οι Stojanovic et al. (2017) δημιούργησαν μια εφαρμογή για κινητά με Λειτουργικό Σύστημα Android, την AGRI-AG, η οποία αναπαράγει σε τρισδιάστατη μορφή χάρτες αποδόσεων (yield maps) χρησιμοποιώντας τους αλγόριθμους Kriging. Η εφαρμογή χρησιμοποιεί δεδομένα από φωτογραφίες για τη δημιουργία του DEM (Digital Elevation Model) και αρχεία CSV (Comma Separated Value) με τις τιμές της απόδοσης. Τα δεδομένα αυτά αποθηκεύονται σε ένα απομακρυσμένο διακομιστή και η εφαρμογή μέσω είτε WebSockets, είτε του HTTP πρωτοκόλλου συνδέεται με το διακομιστή για τη λήψη των δεδομένων. Στη συνέχεια παράγει τους τρισδιάστατους χάρτες απόδοσης. Επιπλέον, η εφαρμογή έχει τη δυνατότητα να παράξει και χρονοσειρές με τα δεδομένα της απόδοσης.

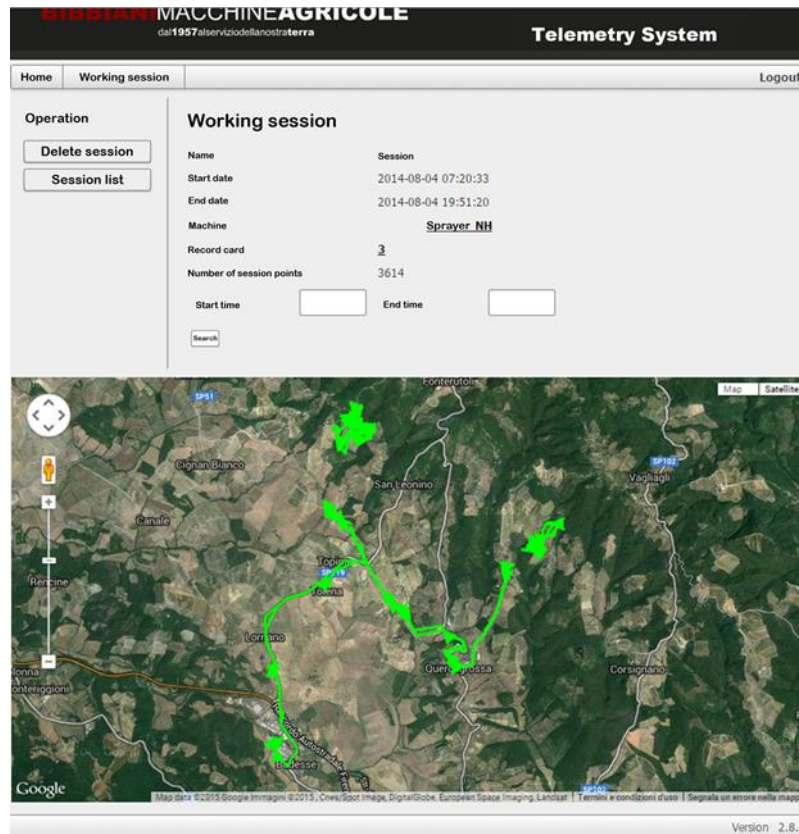
Οι Sarri et al. (2017) ανέπτυξαν ένα πρωτότυπο σύστημα τηλεμετρίας για την καταγραφή των επεμβάσεων ψεκασμού στους αμπελώνες. Μέσω αυτού του συστήματος οι παραγωγοί μπορεί να παρακολουθούν την απόδοση των επεμβάσεων ψεκασμού σε πραγματικό χρόνο και να λαμβάνουν χρήσιμα δεδομένα. Το σύστημα αποτελείται από τη μονάδα λήψης των δεδομένων, ένα διακομιστή για τον έλεγχο του συστήματος και την αποθήκευση των δεδομένων, μια μονάδα GSM/GPRS/GPS (Global System for Mobile communication/General Packet Radio Service/Global Positioning System) για τη μετάδοση των δεδομένων και ένα GNSS (Global Navigation Satellite Systems) για τις πληροφορίες θέσης του ψεκασμού. Μεταξύ άλλων τα δεδομένα τα οποία μπορούν να ληφθούν είναι οι συντεταγμένες, η ταχύτητα του ψεκαστήρα και η παροχή των ακροφυσίων.



Εικόνα 1.12: Ένα χάρτης απόδοσης της εφαρμογής AGRI-AG (Stojanovic et al., 2017)



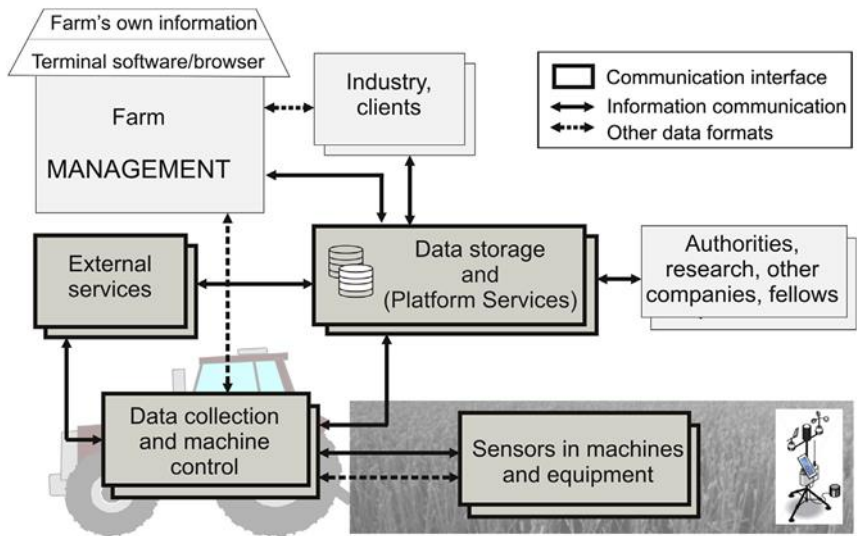
Εικόνα 1.13: Η αρχιτεκτονική της πλατφόρμας του διακομιστή ελέγχου του συστήματος (Sarri et al., 2017)



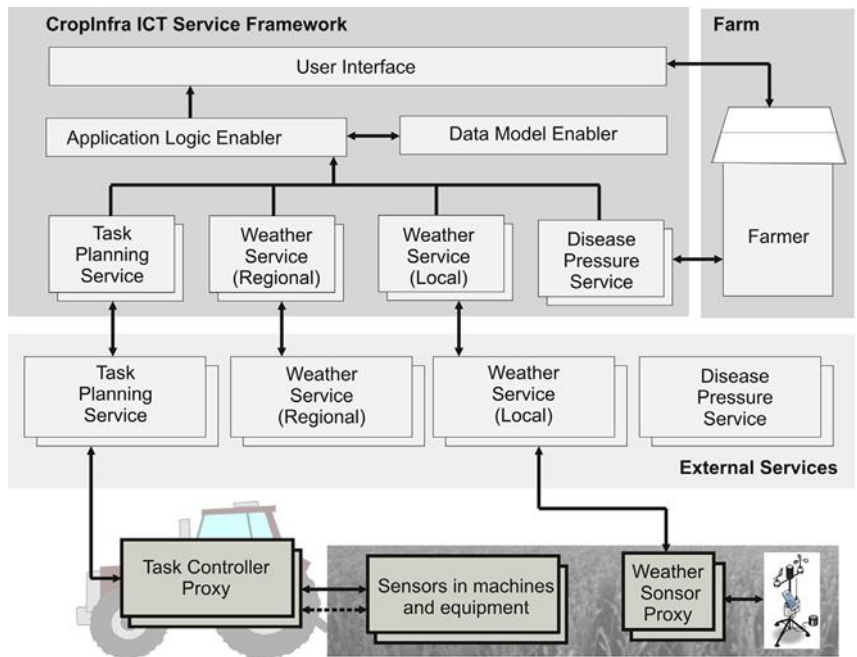
Εικόνα 1.14: Το γραφικό περιβάλλον του συστήματος τηλεμετρίας με την καταγραφή της διαδρομής (Sarri et al., 2017)

Οι Pesonen et al. (2013) δημιούργησαν την πλατφόρμα Cropinfra μέσω της οποίας προσπάθησαν να διερευνήσουν το πώς οι υπηρεσίες που βασίζονται στο διαδίκτυο θα μπορούσαν να επιτρέψουν τη δημιουργία διασυνδεδεμένων συστημάτων που θα βοηθήσουν τους γεωργούς να λειτουργούν πιο αποδοτικά και να καλύψουν τις ανάγκες που υπάρχουν στα αγροκτήματα χρησιμοποιώντας σύγχρονες τεχνολογίες.

Η πλατφόρμα Cropinfra είναι μια υπηρεσία που αναπτύσσεται σε πολλαπλά επίπεδα. Αποτελείται από τέσσερα επίπεδα, τα οποία είναι: 1) Οι αισθητήρες που υπάρχουν στα μηχανήματα και τον εξοπλισμό, 2) η συλλογή δεδομένων και ο έλεγχος των μηχανημάτων, 3) η αποθήκευση των δεδομένων και 4) οι εξωτερικές υπηρεσίες (π.χ. οι προγνώσεις του καιρού).



Εικόνα 1.15: Η επιχειρησιακή αρχιτεκτονική του συστήματος CropInfra (Pesonen et al.,2013)



Εικόνα 1.16: Η αρχιτεκτονική του συστήματος για τον ψεκασμό ακριβείας (Pesonen et al.,2013)

1.5 Εφαρμογή υγρών φυτοπροστατευτικών

1.5.1 Γενικά

Στον αγρό η εφαρμογή των υγρών φυτοπροστατευτικών γίνεται μέσω των κατάλληλων μηχανημάτων. Τα μηχανήματα εφαρμογής των υγρών φυτοπροστατευτικών είναι μηχανήματα που διασκορπίζουν το ψεκαστικό διάλυμα υπό τη μορφή λεπτών σταγονιδίων πάνω στην επιφάνεια των φυτικών οργάνων. Πριν από την εφαρμογή η δραστική ουσία διαλύεται σε νερό. Επίσης, τα σκευάσματα μπορεί να περιέχουν και επιπρόσθετα στοιχεία εκτός της δραστικής ουσίας τα οποία μπορεί είτε να αυξάνουν την αποτελεσματικότητά της, είτε τη διασπορά και την προσκόλληση στον στόχο.

Σε ένα ψεκασμό ο βασικός σκοπός είναι η όσο το δυνατόν μεγαλύτερη κάλυψη του στόχου, με όσο γίνεται μεγαλύτερη ομοιομορφία στη διασπορά του διαλύματος. Για να επιτευχθεί μεγαλύτερη επιφάνεια κάλυψης θα πρέπει και η διάμετρος των σταγόνων να είναι όσο το δυνατόν μικρότερη. Από την άλλη όμως, όσο μικρότερες είναι οι σταγόνες τόσο πιο πιθανός ο κίνδυνος με ένα ρεύμα αέρα να παρασυρθεί το ψεκαστικό διάλυμα εκτός στόχου. Ένα άλλο πρόβλημα από το μικρό μέγεθος των σταγόνων είναι ότι το νερό εξατμίζεται πιο γρήγορα σε θερμές και ξηρές συνθήκες. Όταν το νερό εξατμίζεται τότε η δραστική ουσία που περιείχε η σταγόνα συνεχίζει να αιωρείται με αποτέλεσμα να μεταφέρεται μακριά. Για αυτούς του λόγους η μικρότερη δυνατή διάμετρος των σταγόνων δεν είναι πάντα επιθυμητή και θα πρέπει να επιλέγεται η κατάλληλη διάμετρος ανάλογα με τα χαρακτηριστικά του ψεκασμού (Γέμτος & Καβαλάρης, 2015).

Επειδή, η εφαρμογή των ψεκαστικών γίνεται σε διάφορες καλλιέργειες και κάτω από διάφορες συνθήκες έχουν αναπτυχθεί διάφορα είδη ψεκαστικών μηχανημάτων. Υπάρχουν οι επινώτιοι ψεκαστήρες που τους μεταφέρει ο χειριστής και μπορεί να είναι χειροκίνητοι ή να έχουν μια μηχανή εσωτερικής καύσης. Τα μηχανοκίνητα ψεκαστικά που φέρονται ή έλκονται από τους γεωργικούς ελκυστήρες διακρίνονται σε δύο κατηγορίες: 1) αν η κατεύθυνση του ψεκασμού είναι προς τα κάτω, τότε χαρακτηρίζονται ως μεγάλων καλλιεργειών και 2) αν η κατεύθυνση του ψεκασμού είναι προς τα πάνω και πλάγια, τότε ονομάζονται δενδρωδών καλλιεργειών (Γέμτος & Καβαλάρης, 2015).

1.5.2 Μηχανοκίνητα ψεκαστικά μηχανήματα

1.5.2.1 Περιγραφή

Τα κύρια μέρη ενός ψεκαστικού μηχανήματος είναι τα εξής (Γέμτος & Καβαλάρης, 2015):

1. Πλαίσιο
2. Δοχείο

3. Φίλτρα
4. Αντλία
5. Χειριστήρια
6. Σωλήνες διανομής
7. Σύστημα έδρασης των ακροφυσίων
8. Ακροφύσια

Το πλαίσιο χρησιμοποιείται ως το στοιχείο 1) πάνω στο οποίο στηρίζονται όλα τα υπόλοιπα εξαρτήματα του ψεκαστικού και 2) που προσδένεται πάνω στον ελκυστήρα.

Το δοχείο είναι το μέσο στο οποίο αποθηκεύεται το ψεκαστικό υγρό και η χωρητικότητά του κυμαίνεται από 400 έως 1000 λίτρα.

Τα φίλτρα βοηθάνε στο καθάρισμα του νερού και του ψεκαστικού διαλύματος, έτσι ώστε να αποφεύγεται το βούλωμα των ακροφυσίων και οι φθορές στην αντλία.

Η αντλία είναι υπεύθυνη για την απορρόφηση του ψεκαστικού διαλύματος από το δοχείο και τη μεταφορά του προς τα ακροφύσια.

Τα χειριστήρια ελέγχουν και ρυθμίζουν την πίεση του ψεκασμού. Τα χειριστήρια αποτελούνται από μια σειρά από εξαρτήματα, όπως ο ρυθμιστής πίεσης, οι βαλβίδες ελέγχου του ψεκαστικού διαλύματος, τα μανόμετρα κ.ά.

Οι σωλήνες διανομής του ψεκαστικού διαλύματος μπορεί να είναι ελαστικοί, συμπαγείς πλαστικοί ή μεταλλικοί και συνδέουν το χειριστήριο με τα ακροφύσια.

Το ψεκαστικό διάλυμα εξέρχεται από τα ακροφύσια, τα οποία έχουν οπές εξόδου μικρής διαμέτρου για να αυξάνεται η πίεση και να γίνεται διάσπαση του διαλύματος σε σταγόνες (Γέμτος & Καβαλάρης, 2015).

1.5.2.2 Ψεκαστικά για δενδρώδεις καλλιέργειες

Λόγω του μεγάλου ύψους των δένδρων απαιτούνται και τα ανάλογα ψεκαστικά μηχανήματα. Στους σύγχρονους οπωρώνες το ύψος των δένδρων είναι χαμηλότερο από τους παλαιότερους και φθάνει έως τα 5 μέτρα. Τα μηχανήματα κινούνται ανάμεσα από τις γραμμές των δένδρων και χρησιμοποιούν ένα ρεύμα αέρα για να παρασύρει τις σταγόνες και να φθάσουν έως την κόμη των δένδρων. Επίσης, λόγω της ανάδευσης των φύλλων από το ρεύμα αέρα είναι δυνατό να φθάσουν οι σταγόνες έως το εσωτερικό φύλλωμα της κόμης. Για να επιτευχθεί αυτό το είδος διανομής, υπάρχουν δύο τύποι συστημάτων. Ο πρώτος είναι με τους εκτοξευτήρες διατεταγμένους σε ένα ημικυκλικό ιστό και ο δεύτερος με το ιστό να περιβάλλει το δένδρο ή το πρέμνο και τα ακροφύσια να πλησιάζουν το στόχο. Ο δεύτερος τύπος έχει εφαρμογή σε χαμηλούς οπωρώνες και αμπελώνες (Γέμτος & Καβαλάρης, 2015).

1.5.2.3 Ρύθμιση ψεκαστικών

Για τη ρύθμιση της δόσης ψεκασμού πρώτα πρέπει να μετρηθεί η παροχή του ψεκαστικού για τα ακροφύσια που θα χρησιμοποιηθούν. Αυτό γίνεται με μέτρηση της παροχής των ακροφυσίων για συγκεκριμένη πίεση λειτουργίας ενώ το ψεκαστικό λειτουργεί με καθαρό νερό. Το πλάτος του ψεκασμού καθορίζεται από το πλάτος του ιστού ή από το γινόμενο του αριθμού των ακροφυσίων επί τη σχετική τους απόσταση (Γέμτος & Καβαλάρης, 2015).

Η ρύθμιση του ψεκασμού δίνεται από την παρακάτω σχέση (Γέμτος & Καβαλάρης, 2015):

$$P = \frac{Q}{A \cdot u}$$

P = ρύθμιση του ψεκαστικού (L/στρέμμα)

Q = παροχή του συνόλου των ακροφυσίων (L/h)

A = απόσταση των δένδρων

u = ταχύτητα εργασίας (km/h)

1.6 Σκοπός

Μέσω αυτής της εργασίας έγινε μια προσπάθεια να αναδειχθούν κάποιες από τις δυνατότητες που υπάρχουν για χρήση διαδικτυακών εφαρμογών (Web applications) στον τομέα της Γεωργίας. Λόγω του ότι 1) ζούμε σήμερα στην εποχή της Τεχνολογίας και οι περισσότεροι άνθρωποι έχουν αρχίσει να εξοικειώνονται σε μεγάλο βαθμό με τη χρήση του Διαδικτύου και 2) στο χώρο της γεωργίας έχουν αρχίσει να κάνουν την εμφάνισή τους εφαρμογές της Γεωργίας Ακριβείας (Precision Agriculture), κρίθηκε σκόπιμο να δημιουργηθεί μια εφαρμογή που θα κάνει χρήση σύγχρονων ψηφιακών τεχνολογιών και θα έχει κάποια χρησιμότητα για τους γεωργούς-χρήστες της. Με αυτό τον τρόπο σταδιακά θα μπορέσει να γίνει αντιληπτή από τους τελικούς χρήστες η χρησιμότητα ανάλογων εφαρμογών.

Στη συγκεκριμένη εργασία δημιουργήθηκε μια διαδικτυακή εφαρμογή που έχει σαν σκοπό αρχικά να παρουσιάζει τα δεδομένα επεμβάσεων ψεκασμού του χρήστη. Επομένως, ο γεωργός θα μπορεί να έχει σε ένα σημείο, μαζεμένα, όλα τα δεδομένα από επεμβάσεις ψεκασμού που έχει πραγματοποιήσει κατά το παρελθόν. Κάτι τέτοιο είναι δυνατό να τον βοηθήσει στη βελτιστοποίηση της απόδοσης των μελλοντικών του επεμβάσεων ψεκασμού. Εκτός από την παρουσίαση και καταγραφή των επεμβάσεων ψεκασμού, ο χρήστης της εφαρμογής μπορεί να ψηφιοποιήσει τα αγροκτήματά του και να κάνει κάποιους βασικούς χωρικούς υπολογισμούς.

Ένας γεωργός που μέχρι τώρα δεν είχε κάποια δυνατότητα για καταγραφή των δεδομένων από τους ψεκασμούς του, θα μπορεί μέσω της εφαρμογής να έχει ένα

ιστορικό όπου θα βλέπει —χωρικά και μη— δεδομένα από καταγεγραμμένους ψεκασμούς. Αυτή η δυνατότητα θα τον βοηθήσει στο να κάνει στο μέλλον καλύτερες επιλογές είτε σε υλικό, είτε σε χειρισμούς ψεκασμού.

Τέλος, έγινε προσπάθεια κατά το σχεδιασμό της εφαρμογής η χρήση της να είναι όσο το δυνατόν πιο φιλική στον τελικό χρήστη. Αυτό έγινε διότι, κατά καιρούς πολλές χρήσιμες εφαρμογές, με δύσχρηστη όμως διεπαφή χρήστη (User Interface), δεν αξιοποιήθηκαν ουσιαστικά από τους χρήστες τους, αφού αυτοί απέφευγαν να τις χρησιμοποιήσουν λόγω της κακής εμπειρίας χρήσης τους.

2 Υλικά και Μέθοδοι

2.1. Η γλώσσα προγραμματισμού Python

2.1.1 Εισαγωγή

Η Python, μια γενικού-σκοπού (general-purpose) διερμηνευόμενη (interpreted) γλώσσα προγραμματισμού, δημιουργήθηκε από τον Guido van Rossum τη δεκαετία του 1990. Είναι πολύ υψηλού επιπέδου (VHLL – Very High-Level Language), δυναμική και αντικειμενοστρεφής. Προσφέρει υψηλή παραγωγικότητα σε όλες τις φάσεις του κύκλου της ανάπτυξης λογισμικού: στην ανάλυση, στο σχεδιασμό, στη συγγραφή του κώδικα, στον έλεγχο (testing), στην απασφαλμάτωση (debugging), στη βελτιστοποίηση, στην τεκμηρίωση (documentation), στην ανάπτυξη και στη συντήρηση (Martelli et al., 2017).

2.1.2 Συντακτικό

2.1.2.1 Βασική δομή ενός σεναρίου (script)

Ένα από τα πρώτα πράγματα που θα παρατηρήσει κάποιος στην αρχή κάθε script είναι οι εντολές *import*. Αυτές οι γραμμές κώδικα φορτώνουν τα πρόσθετα αρθρώματα (modules) έτσι ώστε τα scripts να μπορούν να τα χρησιμοποιήσουν. Ένα άρθρωμα είναι βασικά μια βιβλιοθήκη κώδικα την οποία μπορούμε να τη χρησιμοποιήσουμε μέσα στο δικό μας script. Το πρώτο πράγμα που θα χρειαστεί να κάνει κανείς για να χρησιμοποιήσει ένα άρθρωμα είναι να το φορτώσει χρησιμοποιώντας την εντολή *import*. Μετά θα έχει πρόσβαση στα αντικείμενα του αρθρώματος χρησιμοποιώντας πριν το εκάστοτε αντικείμενο το όνομα του αρθρώματος ακολουθούμενο από μια τελεία. Στο παρακάτω παράδειγμα (όταν οι γραμμές ξεκινάνε με >>> είμαστε σε περιβάλλον διερμηνευτή) φορτώνεται το *random* άρθρωμα και μετά χρησιμοποιείται η μέθοδος *gauss* με σκοπό να πάρουμε ένα τυχαίο αριθμό από την τυπική κανονική κατανομή:

```
>>>import random
>>>random.gauss(0, 1)
-0.22186423850882403
```

Στα scripts της Python σε αντίθεση με άλλες γλώσσες δεν υπάρχουν η άνω τελεία (; - semicolon) και τα άγκιστρα ({ } – curly braces) για το τέλος μιας γραμμής και για ένα μπλοκ κώδικα αντίστοιχα. Η Python χρησιμοποιεί τις εσοχές (indentation) για τον προσδιορισμό της δομής του κώδικα (Garrard, 2016). Ακολουθεί ένα παράδειγμα όπου φαίνεται η διαφορά μεταξύ της Python και της C πάνω σε αυτό το θέμα:

Python

```
while n > 0:  
    r *=n  
    n -=1
```

C

```
while (n > 0) {  
    r *= n;  
    --n;  
}
```

Επίσης, τα σχόλια σε ένα script ξεκινάνε με το σύμβολο # και ότι ακολουθεί μετά αγνοείται από την Python, ένα παράδειγμα:

```
random.gauss(0, 1) #random number -> std normal distribution
```

2.1.2.2 Μεταβλητές

Η πιο συχνά χρησιμοποιούμενη εντολή στην Python είναι η εκχώρηση, για παράδειγμα έστω ότι έχουμε μια μεταβλητή η οποία καλείται x και θέλουμε να της εκχωρήσουμε την τιμή 5:

```
x = 5
```

Δεν χρειάζεται να γίνει δήλωση μεταβλητής όπως σε άλλες γλώσσες, οι μεταβλητές δημιουργούνται αυτόματα στο σημείο στο οποίο εκχωρούνται οι τιμές τους για πρώτη φορά.

Οι μεταβλητές μπορούν να έχουν οποιοδήποτε αντικείμενο, σε αντίθεση με τη C ή άλλες γλώσσες που μπορούν να έχουν μόνο ένα τύπο μεταβλητής. Το ακόλουθο είναι απολύτως σωστό στην Python:

```
>>> x = "Hello"  
>>> print(x)  
Hello  
>>> x = 5  
>>> print(x)  
5
```

Κάθε καινούργια εκχώρηση της μεταβλητής αναιρεί τις προηγούμενες. Ακόμα, η εντολή *del* διαγράφει τη μεταβλητή.

```
>>> x = 5  
>>> print(x)  
5  
>>> del x
```

Στο παραπάνω παράδειγμα μετά την εντολή *del x* αν προσπαθήσουμε να χρησιμοποιήσουμε τη μεταβλητή *x* τότε θα υπάρξει σφάλμα.

2.1.2.3 Εκφράσεις

Η Python υποστηρίζει αριθμητικές και παρόμοιες εκφράσεις (expressions). Το ακόλουθο τμήμα κώδικα υπολογίζει τη μέση τιμή του 3 και του 5, αποθηκεύοντας το αποτέλεσμα στη μεταβλητή *z*:

```
>>> x = 3
>>> y = 5
>>> z = (x + y) / 2
4.0
```

Σε αντίθεση με τους αριθμητικούς κανόνες που ισχύουν στη C, οι αριθμητικοί τελεστές που χρησιμοποιούνται με ακεραίους μόνο, δεν επιστρέφουν πάντα ένα ακέραιο. Παρόλο που όλες οι τιμές είναι ακέραιοι, η διαίρεση (από την Python 3 και μετά) επιστρέφει ένα αριθμό κινητής-υποδιαστολής, έτσι το κλασματικό μέρος δεν περικόπτεται. Εάν χρειάζεται η κλασική διαίρεση μεταξύ ακεραίων, όπου θα επιστρέφεται ένας αριθμός που θα έχει γίνει περικοπή, μπορεί να χρησιμοποιηθεί ο τελεστής `//` (Ceder, 2010).

2.1.2.4 Τύποι Δεδομένων

Πριν την αναφορά στους τύπους δεδομένων είναι σημαντικό να υπάρχει μια εικόνα του τρόπου με τον οποίο διαιρούνται ιεραρχικά τα προγράμματα της Python. Τα προγράμματα αυτά μπορούν να αναλυθούν σε αρθρώματα (modules), εντολές (statements), εκφράσεις (expressions) και αντικείμενα (objects), ως εξής:

1. Τα προγράμματα συντίθεται από αρθρώματα (modules).
2. Τα αρθρώματα (modules) περιέχουν εντολές.
3. Οι εντολές αποτελούνται από εκφράσεις.
4. Οι εκφράσεις δημιουργούν και επεξεργάζονται αντικείμενα.

Στον πίνακα 2.1 αναφέρονται οι τύποι αντικειμένων της Python και κάποια παραδείγματά τους.

Πίνακας 2.1: Τύποι αντικειμένων της Python (Lutz, 2013)

| Τύπος αντικειμένου | Παραδείγματα |
|-------------------------|---|
| Numbers (Αριθμοί) | 1234 , 3.1415 , 3+4j , 0b111 , Decimal() , Fraction() |
| Strings (Συμβολοσειρές) | 'spam' , "Bob's" , b'a\χ01c' , u'sp\χc4m' , str('abcd') |
| Lists (Λίστες) | [1, [2, 'three'], 4.5] , list(range(10)) |

| | |
|---------------------------|---|
| Tuples (Πλειάδες) | (1, 'spam', 4, 'U') , tuple('spam') , namedtuple |
| Sets (Σύνολα) | set('abc') , {'a', 'b', 'c'} |
| Dictionaries (Λεξικά) | {'food': 'spam', 'taste': 'yum'} , dict(hours=10) |
| Files (Αρχεία) | open('eggs.txt') , open(r'C:\ham.bin', 'wb') |
| Διάφορα | Boolean, None |
| Προγραμματιστικές μονάδες | Συναρτήσεις, Αρθρώματα, Εξαιρέσεις |

Στον πίνακα φαίνεται ότι οι προγραμματιστικές μονάδες (συναρτήσεις, αρθρώματα και εξαιρέσεις) σε αντίθεση με άλλες γλώσσες είναι αντικείμενα, όπως είναι για παράδειγμα και οι συμβολοσειρές.

Για να γίνει πιο κατανοητή αυτή η διαφορά σε σχέση με άλλες γλώσσες ακολουθεί μια σύγκριση μεταξύ της Python και της Java. Στο παράδειγμα που ακολουθεί, στην αρχή εμφανίζεται η ομοιότητα τους στις συμβολοσειρές, όπου και στις δύο γλώσσες μια συμβολοσειρά είναι ένα στιγμιότυπο του τύπου object. Στη συνέχεια εμφανίζεται η διαφορά τους, όπου για παράδειγμα στις συναρτήσεις στη Python είναι αντικείμενα, ενώ στη Java δεν ισχύει το ίδιο.

Έστω ότι έχουμε μια συμβολοσειρά με όνομα *s*, στη συνέχεια ελέγχουμε αν είναι ένα στιγμιότυπο τύπου Object.

| Python | Java |
|---|---|
| <pre>s = 'Hello'</pre> <pre>print(s.__class__.__name__)</pre> | <pre>String s = "Hello";</pre> <pre>System.out.println(s.getClass().getName());</pre> |
| Εκτυπώνει: <code>'str'</code> | Εκτυπώνει: <code>java.lang.String</code> |
| <pre>print(isinstance(s,object))</pre> | <pre>System.out.println(s instanceof Object);</pre> |
| Εκτυπώνει: <code>True</code> | Εκτυπώνει: <code>true</code> |

Όπως βλέπουμε και στις δύο περιπτώσεις τα αντικείμενα των συμβολοσειρών είναι τύπου Object.

Έστω ότι έχουμε τη συνάρτηση *calculate* η οποία επιστρέφει υψωμένο στο τετράγωνο ένα αριθμό που της δίνουμε ως όρισμα.

| Python | Java |
|---|---|
| <pre>def calculate(x): return x*x print(isinstance(calculate, object))</pre> | <pre>public static int calculate(int x){ return x*x; } System.out.println(calculate instanceof Object);</pre> |
| <p>Εκτυπώνει: <code>True</code></p> | <pre>Testing.java:13: error: cannot find symbol System.out.println(calculate instanceof Object); ^ symbol: variable calculate location: class Testing 1 error</pre> |

Στη Python η συνάρτηση *calculate* είναι ένα αντικείμενο τύπου `object`. Στη Java συμβαίνει φυσιολογικά σφάλμα, αφού μια συνάρτηση δεν μπορεί να είναι αντικείμενο.

Από τα παραπάνω παραδείγματα φαίνεται ένα άλλο πλεονέκτημα της Python σε σχέση με άλλες γλώσσες, ο κώδικας της καταλαμβάνει λιγότερο χώρο, με άλλα λόγια έχουμε την ίδια λειτουργικότητα γράφοντας λιγότερο.

Το χαρακτηριστικό των `scripts` της Python να είναι τα πάντα ένα αντικείμενο, είτε πρόκειται για απλούς αριθμούς, είτε για τους τελεστές των μεταξύ τους πράξεων, ήταν βασικό στοιχείο που χρησιμοποιήθηκε στην ίδια τη γλώσσα και υπάρχει για να το εκμεταλλεύονται οι χρήστες της. Για να τονιστεί αυτό ακολουθεί ένα παράδειγμα όπου φαίνεται πώς εκμεταλλεύεται η γλώσσα τον τελεστή `+` για διαφορετικούς τύπους αντικειμένων:

```
1 >>> x = 1
2 >>> x + 2
3 3
4 >>> x.__add__(2)
5 3
6 >>> y = 'a'
```

```

7   >>> y + 'b'
8   'ab'
9   >>> y.__add__('b')
10  'ab'

```

Στο παραπάνω παράδειγμα βλέπουμε τον τελεστή + να χρησιμοποιείται και από αντικείμενα τύπου int και από αντικείμενα τύπου str. Αυτό είναι εφικτό επειδή στην ουσία στην Python οι τελεστές είναι μέθοδοι οι οποίες ορίζονται στις αντίστοιχες κλάσεις. Για να είναι εφικτή αυτή η λειτουργία χρησιμοποιούνται οι μέθοδοι υπερφόρτωσης τελεστών. Δηλαδή όταν στη γραμμή 2 προσθέτουμε το x με το 2 αυτό που συμβαίνει εσωτερικά είναι αυτό που φαίνεται στη γραμμή 4. Αντίστοιχα, όταν κάνουμε σύνδεση δύο συμβολοσειρών (string concatenation) στη γραμμή 7, αυτό που γίνεται εσωτερικά φαίνεται στη γραμμή 9.

| Έκφραση | Μέθοδος που καλείται | Έκφραση | Μέθοδος που καλείται |
|-------------|----------------------|----------|----------------------|
| +a | a.__pos__() | a > b | a.__gt__(b) |
| -a | a.__neg__() | a >= b | a.__ge__(b) |
| ~a | a.__invert__() | a & b | a.__and__(b) |
| abs(a) | a.__abs__() | a b | a.__or__(b) |
| a + b | a.__add__(b) | a ^ b | a.__xor__(b) |
| a - b | a.__sub__(b) | a << b | a.__lshift__(b) |
| a * b | a.__mul__(b) | a >> b | a.__rshift__(b) |
| a / b | a.__truediv__(b) | a (bool) | a.__bool__() |
| a // b | a.__floordiv__(b) | len(a) | a.__len__() |
| a % b | a.__mod__(b) | a in s | s.__contains__(a) |
| divmod(a,b) | a.__divmod__(b) | d[k] | d.__getitem__(k) |
| a ** b | a.__pow__(b) | d[k] = v | d.__setitem__(k, v) |
| a == b | a.__eq__(b) | iter(a) | a.__iter__() |
| a != b | a.__ne__(b) | next(a) | a.__next__() |
| a < b | a.__lt__(b) | str(a) | a.__str__() |
| a <= b | a.__le__(b) | repr(a) | a.__repr__() |

Εικόνα 2.1: Οι ειδικές μέθοδοι που καλούνται από τη Python

(Toal et al., 2017)

Αριθμοί

Οι αριθμητικοί τύποι της γλώσσας συμπεριλαμβάνουν: τους ακεραίους, τους κινητής υποδιαστολής και τους μιγαδικούς αριθμούς. Όλοι οι αριθμοί στην Python είναι αμετάβλητα αντικείμενα, έτσι όταν εκτελείται μια πράξη σε ένα αριθμητικό αντικείμενο παράγεται ένα νέο αριθμητικό αντικείμενο (Martelli et al., 2017).

Ακέραιοι (Integers)

Οι ακέραιοι μπορεί να είναι σε δεκαδική, δυαδική, οκταδική ή δεκαεξαδική μορφή.

Παραδείγματα:

| | |
|--------------|--------------------|
| Δεκαδικοί | 1, 23, 3493 |
| Δυαδικοί | 0b010101, 0b110010 |
| Οκταδικοί | 0o1, 0o27, 0o6645 |
| Δεκαεξαδικοί | 0x1, 0x17, 0xDA5 |

Κινητής υποδιαστολής (Floating-point)

Ένας αριθμός κινητής υποδιαστολής είναι μια ακολουθία από δεκαδικά ψηφία τα οποία περιέχουν ένα δεκαδικό σημείο (.) ή ένα εκθετικό πρόσφυμα (e/E, ακολουθούμενο προαιρετικά από ένα + ή – το οποίο ακολουθείται από ένα ή περισσότερα ψηφία) ή και τα δύο.

Παραδείγματα:

0., 0.0, .0, 1., 1.0, 1e0, 1.e0, 1.0e0

Μιγαδικοί (Complex)

Ένας μιγαδικός αριθμός αποτελείται από 2 τιμές κινητής-υποδιαστολής, μια για το πραγματικό και μια για το φανταστικό μέρος. Ο αριθμός του φανταστικού μέρους ακολουθείται από το χαρακτήρα j/J.

Παραδείγματα για το φανταστικό μέρος:

0j, 0.j, 0.0j, .0j, 1j, 1.j, 1.0j, 1e0j, 1.e0j, 1.0e0j

Συμβολοσειρές

Οι συμβολοσειρές (strings) υποδεικνύονται με τη χρήση διπλών εισαγωγικών (double quotes). Η παρακάτω γραμμή εκχωρεί τη συμβολοσειρά "Hello World" στη μεταβλητή x:

```
x = "Hello World"
```

Ανάστροφοι κάθετοι (backslash) μπορούν να χρησιμοποιηθούν μέσα στις συμβολοσειρές για χαρακτήρες με ξεχωριστή σημασιολογία. \n είναι ο χαρακτήρας της καινούργιας γραμμής, \t είναι ο χαρακτήρας του tab, \\ είναι ο χαρακτήρας για μια μονή κανονική κάθετο και το \" είναι ο χαρακτήρας για τα διπλά εισαγωγικά.

Επίσης, μπορούν να χρησιμοποιηθούν μονά εισαγωγικά (single quotes) αντί των διπλών. Οι επόμενες δύο γραμμές έχουν ακριβώς την ίδια λειτουργία:

```
x = "Hello World"
x = 'Hello World'
```

Η μοναδική διαφορά είναι ότι δεν χρειάζονται οι ανάστροφοι κάθετοι για τους “ χαρακτήρες στις συμβολοσειρές με μονά εισαγωγικά και για τους ‘ χαρακτήρες στις συμβολοσειρές με διπλά εισαγωγικά:

```
x = "Don't"
x = 'Can\'t'
x = "\\"
```

Ακόμα, η Python δίνει τη δυνατότητα να χρησιμοποιηθούν συμβολοσειρές με τριπλά εισαγωγικά, οι οποίες δίνουν τη δυνατότητα να σπάει μια συμβολοσειρά σε πολλαπλές γραμμές και να χρησιμοποιούνται μέσα της χαρακτήρες με μονά ή διπλά εισαγωγικά χωρίς τη χρήση ανάστροφων καθέτων.

```
x = """1st line
      2nd line
      3rd line"""
```

Εδώ το x είναι ολόκληρη η συμβολοσειρά που εκτείνεται σε τρεις γραμμές (Ceder, 2010).

Λίστες (Lists)

Μια λίστα είναι μια μεταβλητή ακολουθία από στοιχεία. Τα στοιχεία σε μια λίστα είναι αντικείμενα τα οποία μπορούν να είναι και διαφορετικού τύπου. Η δήλωση μιας λίστας γίνεται χρησιμοποιώντας μια σειρά από εκφράσεις (τα στοιχεία της λίστας) που χωρίζονται μεταξύ τους με κόμμα και είναι μέσα σε αγκύλες.

Παραδείγματα:

```
[42, 3.14, 'hello']
```

```
[100]
```

```
[]
```

Επίσης, μια λίστα μπορεί να δημιουργηθεί και με την κλήση της μεθόδου *list()* (Martelli et al., 2017).

Πλειάδες (Tuples)

Μια πλειάδα είναι μια αμετάβλητη ακολουθία από στοιχεία. Τα στοιχεία μπορεί να ανήκουν σε διαφορετικούς τύπους. Η δήλωση μιας πλειάδας γίνεται με μια σειρά

από εκφράσεις (τα στοιχεία της πλειάδας) που χωρίζονται μεταξύ τους με κόμμα και είναι μέσα σε παρένθεση.

Παραδείγματα:

```
(100, 200, 300)
```

```
(3.14,)
```

```
()
```

Επίσης, μια πλειάδα μπορεί να δημιουργηθεί και με την κλήση της μεθόδου *tuple()* (Martelli et al., 2017).

Σύνολα (Sets)

Με τα σύνολα, αναπαρίστανται συλλογές από μοναδικά στοιχεία, τα οποία μπορεί να είναι διαφορετικών τύπων, αλλά πρέπει να είναι κατακερματιζόμενα. Τα στιγμιότυπα του τύπου *set* είναι μεταβλητά. Ένα σύνολο δηλώνεται με μια σειρά εκφράσεων που χωρίζονται με κόμμα και βρίσκονται μέσα σε άγκιστρα.

Παραδείγματα:

```
{42, 3.14, 'hello'}
```

```
{100}
```

Ένα κενό σύνολο δηλώνεται με τη μέθοδο *set()* (Martelli et al., 2017).

Λεξικά (Dictionaries)

Τα λεξικά είναι συλλογές αντικειμένων, είναι ένα ζεύγος κλειδιού και τιμής. Με τον τύπο δεδομένων λεξικό τα αντικείμενα αποθηκεύονται με βάση το κλειδί τους και όχι με βάση τη σχετική τους θέση.

Παραδείγματα:

```
{'x':42, 'y':3.14, 'z':7}
```

```
{1:2, 3:4}
```

```
{1:'za', 'br':23}
```

```
{}
```

Επίσης, λεξικά μπορούν να δημιουργηθούν και με τη χρήση της μεθόδου *dict()* (Martelli et al., 2017).

2.1.2.5 Έλεγχος ροής

Ο έλεγχος της ροής ενός προγράμματος είναι η σειρά με την οποία εκτελείται ο πηγαίος κώδικας. Ο έλεγχος της ροής ενός προγράμματος στην Python βασίζεται στις υπό συνθήκη εντολές, στους βρόγχους και στις κλήσεις συναρτήσεων. Επίσης, η ροή του προγράμματος επηρεάζεται από την πρόκληση και τον χειρισμό των εξαιρέσεων (exceptions).

Η εντολή `if`

Πολύ συχνά χρειάζεται να εκτελεστούν κάποιες εντολές μόνο όταν ισχύει κάποια συνθήκη ή επιλέγονται να εκτελεστούν κάποιες εντολές εξαρτώμενες από αμοιβαίως αποκλειόμενες (mutually exclusive) συνθήκες. Η σύνθετη εντολή `if`, αποτελούμενη από τις προτάσεις `if`, `elif` και `else`, δίνει τη δυνατότητα της εκτέλεσης υπό συνθήκη τμημάτων εντολών. Η σύνταξη της εντολής `if`:

```
if έκφραση:
    εντολή/εντολές
elif έκφραση:
    εντολή/εντολές
elif έκφραση:
    εντολή/εντολές
...
else:
    εντολή/εντολές
```

Οι προτάσεις `elif` και `else` είναι προαιρετικές. Σε αντίθεση με άλλες γλώσσες η Python δεν έχει την εντολή `switch`.

Παράδειγμα χρήσης της εντολής:

```
if x < 0:
    print('x: negative')
elif x % 2:
    print('x: positive and odd')
else:
    print('x: even and non-negative')
```

Στο παραπάνω παράδειγμα εκτυπώνονται τα αντίστοιχα μηνύματα ανάλογα με το αν το `x` είναι θετικός ή αρνητικός αριθμός και άρτιος ή περιττός.

Η εντολή `while`

Η εντολή `while` επαναλαμβάνει την εκτέλεση μια εντολή ή μιας ομάδας εντολών για όσο μια συνθήκη είναι αληθής. Το συντακτικό της εντολής `while`:

while έκφραση:
εντολή/εντολές

Παράδειγμα χρήσης της εντολής:

```
x = 'spam'  
while x:  
    print(x, end=' ')  
    x = x[1:]
```

Στο παραπάνω παράδειγμα σε κάθε επανάληψη εκτυπώνεται μια συμβολοσειρά από την οποία αφαιρείται κάθε φορά το 1^ο γράμμα της συμβολοσειράς της μεταβλητής x. Δηλαδή, εκτυπώνεται το: spam ram am m

Η εντολή for

Η εντολή *for* επαναλαμβάνει την εκτέλεση μιας εντολής ή μιας ομάδας εντολών ελεγχόμενη από μια επαναληπτική έκφραση. Το συντακτικό της εντολή *for*:

for μεταβλητή_στόχος **in** ακολουθία:
εντολή/εντολές

Παράδειγμα χρήσης της εντολής:

```
z = 'asdf'  
for char in z:  
    print(char)
```

Στο παραπάνω παράδειγμα σε κάθε επανάληψη εκτυπώνεται με τη σειρά ένας χαρακτήρας από την αλφαριθμητική ακολουθία (συμβολοσειρά) 'asdf'.

Οι εντολές break, continue και pass

Η εντολή **break** επιτρέπεται να χρησιμοποιείται μόνο εντός ενός βρόγχου. Όταν η **break** εκτελείται ο βρόγχος τερματίζει.

Παράδειγμα χρήσης της εντολής:

```
while True:  
    name = input('Enter name:')  
    if name == 'stop':
```

```
        break
    print('Hello', name)
```

Στο παραπάνω παράδειγμα όταν χρήστης δώσει ως είσοδο τη συμβολοσειρά 'stop' τότε σταματάει η εκτέλεση του βρόγχου.

Η εντολή **continue** μπορεί να χρησιμοποιηθεί μόνο εντός ενός βρόγχου. Όταν η continue εκτελείται τότε τερματίζει η εκτέλεση της επανάληψης που εκτελείται εκείνη τη στιγμή και η εκτέλεση του προγράμματος περνάει στην επόμενη επανάληψη.

Παράδειγμα χρήσης της εντολής:

```
x = 10
while x:
    x = x-1
    if x % 2 != 0:
        continue
    print(x, end=' ')
```

Στο παραπάνω παράδειγμα εκτυπώνονται μόνο οι άρτιοι αριθμοί που είναι μεγαλύτεροι του 0, δηλαδή εκτυπώνεται: 8 6 4 2 0

Επειδή το σώμα μια σύνθετης εντολής στη Python δεν μπορεί να είναι άδειο υπάρχει η εντολή **pass**. Η εντολή pass δεν εκτελεί κάποια ενέργεια, χρησιμοποιείται σε περιπτώσεις που σε ένα τμήμα του πηγαίου κώδικα απαιτείται από το συντακτικό της γλώσσας μια εντολή, αλλά με βάση τη λογική του προγράμματος δεν χρειάζεται η εκτέλεση κάποιας ενέργειας.

Παράδειγμα χρήσης της εντολής:

```
if condition1(x):
    process1(x)
elif x>23:
    pass
elif condition3(x):
    process3(x)
else:
    process_default(x)
```


Στο παραπάνω παράδειγμα στην περίπτωση που το x είναι μεγαλύτερο του 23 δεν θέλουμε να εκτελεστεί κάποια ενέργεια και χρησιμοποιούμε την *pass* (Martelli et al., 2017).

2.1.2.6 Συναρτήσεις

Μια συνάρτηση είναι μια ομαδοποίηση εντολών οι οποίες εκτελούνται όταν ζητηθεί. Η Python προσφέρει πολλές έτοιμες συναρτήσεις και δίνει τη δυνατότητα στους χρήστες της να ορίσουν τις δικές τους. Όταν ζητείται να εκτελεστεί μια συνάρτηση αυτή η διαδικασία ονομάζεται κλήση συνάρτησης. Μια συνάρτηση στην Python πάντα επιστρέφει μια τιμή είτε None, είτε κάποια άλλη τιμή όπως έχει οριστεί από τον δημιουργό της. Όταν μια συνάρτηση ορίζεται μέσα σε μια κλάση τότε ονομάζεται μέθοδος. Στην Python οι συναρτήσεις είναι αντικείμενα και ως εκ τούτου χειρίζονται όπως τα άλλα αντικείμενα. Έτσι, μπορούμε να περάσουμε μια συνάρτηση ως όρισμα σε μια κλήση κάποιας άλλης μεθόδου. Αντίστοιχα, μια συνάρτηση μπορεί να επιστρέφει μια άλλη συνάρτηση ως αποτέλεσμα (Martelli et al., 2017).

Η εντολή `def` είναι ο πιο κοινός τρόπος για να ορίσουμε μια συνάρτηση. Το συντακτικό της `def`:

```
def function_name(παράμετροι):  
    εντολή/εντολές
```

Πάραδειγμα ορισμού και κλήσης μιας συνάρτησης:

```
def twice(x):  
    return x*2
```

```
twice(2)
```

Στο παραπάνω παράδειγμα ορίζουμε τη συνάρτηση `twice` η οποία επιστρέφει το όρισμα που της δίνουμε και επιστρέφει την τιμή του ορίσματος υψωμένη στο τετράγωνο. Στη συγκεκριμένη περίπτωση η συνάρτηση καλείται με όρισμα το 2 (Martelli et al., 2017).

2.1.2.7 Κλάσεις

Μια κλάση είναι ένα τύπος που ορίζεται από το χρήστη και για τον οποίο μπορούμε να δημιουργήσουμε στιγμιότυπα, δηλαδή αντικείμενα αυτού του τύπου. Η Python υποστηρίζει αυτές τις δυνατότητες μέσω της κλάσης και του στιγμιότυπου του αντικειμένου.

Στην Python μια κλάση είναι ένα αντικείμενο με τα εξής χαρακτηριστικά (Martelli et al., 2017):

- Μπορούμε να καλέσουμε ένα αντικείμενο κλάσης σαν να ήταν συνάρτηση. Αυτή η κλήση επιστρέφει ένα αντικείμενο γνωστό ως το στιγμιότυπο της κλάσης, η κλάση αντίστοιχα είναι γνωστή ως ο τύπος του στιγμιότυπου.
- Μια κλάση έχει ιδιότητες τις οποίες ονομάζουμε αυθαίρετα και στη συνέχεια να δεσμεύσουμε και χρησιμοποιήσουμε.
- Οι ιδιότητες μιας κλάσης που αναφέρονται σε συναρτήσεις ονομάζονται μέθοδοι.
- Μια κλάση μπορεί επίσης να κληρονομεί άλλες κλάσεις, δηλαδή μπορεί να χρησιμοποιεί ιδιότητες που έχουν οριστεί σε άλλες κλάσεις.

Παράδειγμα ορισμού και χρήσης μια κλάσης:

```

1  from math import sqrt
2
3  class Point:
4      def __init__(self, x, y):
5          self.x = x
6          self.y = y
7
8      def calculate_distance(self, other):
9          return sqrt((self.x-other.x)**2 + (self.y-other.y)**2)
10
11
12  p1 = Point(1,2)
13  p2 = Point(2,4)
14  print(p1.calculate_distance(p2))

```

Στο παραπάνω παράδειγμα δημιουργούμε την κλάση *Point* (γραμμές 3-9), η οποία έχει ως μεταβλητές στιγμιότυπου τις συντεταγμένες *x* και *y* (γραμμές 5 και 6). Η *calculate_distance* (γραμμές 8 και 9) είναι μια μέθοδος της κλάσης, η οποία επιστρέφει την Ευκλείδεια απόσταση μεταξύ δύο σημείων. Τη χρησιμοποιούμε δημιουργώντας δύο αντικείμενα, τα *p1* και *p2* (γραμμές 12 και 13). Τέλος, υπολογίζουμε και εκτυπώνουμε τη μεταξύ τους απόσταση (γραμμή 14).

2.1.2.8 Εξαιρέσεις

Η Python χρησιμοποιεί τις εξαιρέσεις (exceptions) με σκοπό να δηλώσει τα σφάλματα. Μια εξαίρεση είναι ένα αντικείμενο το οποίο υποδεικνύει ένα σφάλμα ή μια αντικανονικότητα. Όταν εντοπίζεται ένα σφάλμα η Python υψώνει μια εξαίρεση, δηλαδή δίνει σήμα ότι υπάρχει μια μη κανονική συνθήκη.

Ο χειρισμός μιας εξαίρεσης είναι η λήψη του αντικειμένου της εξαίρεσης και η ανάληψη ενεργειών τέτοιων ώστε να αντιμετωπιστεί αυτή η μη κανονική κατάσταση. Εάν ένα πρόγραμμα δεν διαχειριστεί μια εξαίρεση τότε αυτό τερματίζει με ένα μήνυμα σφάλματος. Για αυτό με τη διαχείριση των εξαιρέσεων ένα πρόγραμμα μπορεί να συνεχίσει να τρέχει παρά την εμφάνιση σφαλμάτων.

Η εντολή `try` είναι υπεύθυνη για το χειρισμό των εξαιρέσεων. Η μορφή της διαχείρισης είναι μια πρόταση `try` ακολουθούμενη από μια ή περισσότερες προτάσεις `except`. Το γενικό συντακτικό της είναι (Martelli et al., 2017):

```
try:  
    εντολή/εντολές  
except [έκφραση [as μεταβλητή_στόχος]]:  
    εντολή/εντολές
```

Παράδειγμα χειρισμού μιας εξαίρεσης:

Έστω ότι έχουμε τη συνάρτηση `fetcher`, η οποία επιστρέφει ένα στοιχείο ενός αντικειμένου (`obj`) με βάση το δείκτη που του ορίζουμε κατά την κλήση (`index`)

```
def fetch(obj, index):  
    return obj[index]
```

Στην περίπτωση που ο δείκτης είναι εντός των ορίων του αντικειμένου το πρόγραμμα τρέχει κανονικά, για παράδειγμα:

```
x = 'spam'  
fetch(x, 3)
```

Εδώ επιστρέφει το 'm'.

Εάν όμως ο δείκτης υποδεικνύει θέση εκτός των ορίων του αντικειμένου τότε εκδηλώνεται σφάλμα, για παράδειγμα:

```
fetcher(x, 4)
```

Θα έχουμε το παρακάτω σφάλμα

```
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
  File "<stdin>", line 2, in fetcher  
IndexError: string index out of range
```

Όπου το `IndexError` είναι η εξαίρεση που αντιστοιχεί σε αυτό το σφάλμα και η έκφραση "string index out of range" η περιγραφή της.

Για να χειριστούμε το σφάλμα πρέπει να χρησιμοποιήσουμε την εντολή try/except ως εξής:

```
try:
    fetch(x,4)
except IndexError as er:
    print(er)
```

Με αυτόν τον τρόπο θα εκτυπωθεί η περιγραφή του σφάλματος και το πρόγραμμα θα μπορέσει να συνεχίσει την εκτέλεσή του.

2.1.2.9 Αρχεία

Τα αρχεία είναι διαμερίσματα στο χώρο αποθήκευσης που τους έχει δοθεί όνομα και τα διαχειρίζεται το λειτουργικό σύστημα.

Όταν θέλουμε να διαβάσουμε τα περιεχόμενα ενός αρχείου χρησιμοποιούμε τη συνάρτηση open().

Παράδειγμα ανάγνωσης κειμένου αρχείου:

```
with open('file_name.txt', 'rt') as f:
    data = f.read()
```

Για να μπει σε λειτουργία διαβάσματος επιλέγεται ως 2^ο όρισμα το 'rt', ενώ αν θέλουμε να γράψουμε στο αρχείο το 2^ο όρισμα θα είναι το 'wt', σε αυτή την περίπτωση ότι υπάρχει ήδη στο αρχείο θα σβηστεί.

Παράδειγμα εγγραφής κειμένου σε αρχείο:

```
with open('file_name.txt', 'wt') as f:
    f.write(text1)
```

Αντίστοιχα, αν θέλουμε να συνεχίσουμε να εγγράψουμε από το τέλος του αρχείου χρησιμοποιούμε το 'at' ως 2^ο όρισμα.

Παράδειγμα προσθήκης κειμένου σε αρχείο:

```
with open('file_name.txt', 'at') as f:
    f.write(text1)
```

2.2 Flask Web Framework

Το Flask είναι ένα μικρό διαδικτυακό framework. Είναι αρκετά μικρό έτσι ώστε όταν κάποιος εξοικειωθεί μαζί του να μπορεί να διαβάσει και να κατανοήσει όλο τον πηγαίο κώδικά του. Το Flask σχεδιάστηκε από την αρχή έτσι ώστε να είναι ένα επεκτάσιμο framework, για αυτό προσφέρει ένα σταθερό πυρήνα μαζί με βασικές υπηρεσίες, ενώ οι επεκτάσεις του προσφέρουν τα υπόλοιπα. Επειδή, μπορεί κανείς να διαλέξει τα πακέτα επεκτάσεων που τον ενδιαφέρουν δεν κινδυνεύει να έχει πράγματα τα οποία του περισσεύουν και δεν τα χρειάζεται (Grinberg, 2014).

Το Flask έχει δύο κύριες εξαρτήσεις. Για τη δρομολόγηση, την απασφαλμάτωση και για τη Διεπαφή Πύλης Διακομιστή Ιστού (Web Server Gateway Interface – WSGI) χρησιμοποιεί το Werkzeug, ενώ για την υποστήριξη προτύπων χρησιμοποιεί το Jinja2. Το Werkzeug και το Jinja2 εξελίσσονται από τον ίδιο προγραμματιστή που εξελίσσει και το Flask, τον Armin Ronacher (Grinberg, 2014).

Δεν υπάρχει υποστήριξη στο Flask για τη διαχείριση στις βάσεις δεδομένων, της επαλήθευσης των φορμών του ιστού, την ταυτοποίηση των χρηστών και άλλες διεργασίες υψηλού επιπέδου. Αυτές και άλλες πολλές υπηρεσίες κλειδιά τις οποίες οι περισσότερες εφαρμογές ιστού χρειάζονται παρέχονται μέσω επεκτάσεων οι οποίες συνεργάζονται με τα πακέτα του πυρήνα του Flask (Grinberg, 2014).

Παράδειγμα μιας βασικής εφαρμογής του Flask:

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    return '<h1>Hello World!</h1>'

if __name__ == '__main__':
    app.run()
```

Εάν ανοίξουμε το φυλλομετρητή μας στη διεύθυνση `http://127.0.0.1:5000/`, τότε θα δούμε το μήνυμα 'Hello World'.

Επεξήγηση του πηγαίου κώδικα:

```
from flask import Flask
```

Όλες οι εφαρμογές Flask χρειάζεται να δημιουργήσουν ένα στιγμιότυπο της εφαρμογής το οποίο είναι ένα αντικείμενο της κλάσης *Flask*.

```
app = Flask(__name__)
```

Το μοναδικό όρισμα που απαιτείται για τον κατασκευαστή της κλάσης *Flask* είναι το όνομα του κύριου αρθρώματος ή πακέτου της εφαρμογής. Για τις περισσότερες εφαρμογές η μεταβλητή `__name__` είναι η κατάλληλη.

```
@app.route('/')
def index():
    return '<h1>Hello World!</h1>'
```

Για να καθορίσουμε τη διαδρομή μιας εφαρμογής χρησιμοποιούμε το decorator *app.route*. Η συνάρτηση *index* διαχειρίζεται το βασικό URL και είναι υπεύθυνη για την εμφάνιση του μηνύματος στο φυλλομετρητή.

```
if __name__ == '__main__':
    app.run()
```

Το τμήμα αυτό είναι υπεύθυνο για την έναρξη του ενσωματωμένου στο Flask διακομιστή ιστού. Η συνθήκη `__name__ == '__main__'` έχει τοποθετηθεί για να βεβαιωθεί ότι ο διακομιστής ιστού θα ξεκινήσει μόνο αν το script εκτελείται άμεσα (Grinberg, 2014).

2.3 SQL

2.3.1 Εισαγωγή στην SQL

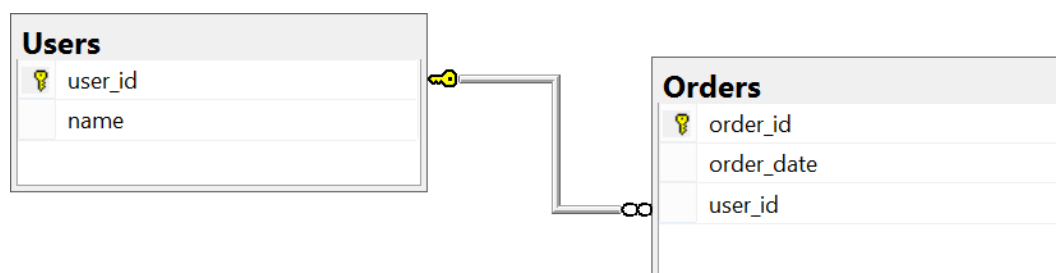
Η SQL (Structured Query Language) είναι η πρωταρχική γλώσσα που χρησιμοποιείται στις σχεσιακές βάσεις δεδομένων, έχει στοιχεία τόσο γλώσσας όσο και λογικής. Ως γλώσσα η SQL συμπεριλαμβάνει ένα μοναδικό συντακτικό με πολλές λέξεις της Αγγλικής γλώσσας, όπως WHERE, FROM και HAVING. Ως λογική έκφραση, προσδιορίζει τις λεπτομέρειες του πώς ανακαλούνται ή ενημερώνονται τα δεδομένα σε μια σχεσιακή βάση δεδομένων. Η SQL είναι η γλώσσα πρότυπο για τη διαχείριση και την εκμετάλλευση των δεδομένων στις σχεσιακές βάσεις. Πιο απλοποιημένα, μπορούμε να πούμε ότι η SQL είναι η γλώσσα η οποία επιτρέπει στους χρήστες της να αλληλοεπιδρούν με μια σχεσιακή βάση. Έχει μια μακρά ιστορία ανάπτυξης από διάφορους οργανισμούς από τη δεκαετία του 1970. Το 1986 το Αμερικάνικο Ινστιτούτο Εθνικών Προτύπων (ANSI – American National Standards Institute) εξέδωσε το πρώτο σύνολο των προδιαγραφών της γλώσσας και από τότε έχουν υπάρξει πολλές αναθεωρήσεις (Rockoff L., 2017).

Γενικότερα στην SQL υπάρχουν τρία βασικά στοιχεία. Το πρώτο είναι η Γλώσσα Χειρισμού Δεδομένων (**DML** – Data Manipulation Language). Αυτό το τμήμα της γλώσσας επιτρέπει την ανάκτηση, την ενημέρωση, την προσθήκη ή τη διαγραφή των δεδομένων σε μια βάση. Το δεύτερο στοιχείο είναι η Γλώσσα Ορισμού Δεδομένων (**DDL** – Data Definition Language), το οποίο μας επιτρέπει να δημιουργούμε και να

αλλάζουμε την ίδια τη βάση. Για παράδειγμα η Γλώσσα Ορισμού Δεδομένων μας παρέχει την εντολή ALTER η οποία μας δίνει τη δυνατότητα να προχωρούμε σε αλλαγές στο σχεδιασμό των πινάκων σε μια βάση δεδομένων. Το τρίτο στοιχείο είναι η Γλώσσα Ελέγχου Δεδομένων (**DCL** – Data Control Language) η οποία συμβάλλει στην ασφάλεια της βάσης. Πολλές μεγάλες εταιρίες λογισμικού όπως η Microsoft και η Oracle εξέλιξαν το πρότυπο για τους δικούς τους σκοπούς και προσθέσανε πολλές επεκτάσεις και τροποποιήσεις στη γλώσσα. Παρόλο που η κάθε εταιρία έχει το δικό της μοναδικό διερμηνευτή της SQL, από κάτω συνεχίζει να υπάρχει η ίδια θεμελιώδης γλώσσα. Η SQL ως μια γλώσσα υπολογιστή είναι διαφορετική από άλλες συνήθεις γλώσσες όπως η Python ή η C. Αυτές οι γλώσσες τείνουν να είναι πιο διαδικαστικές, ενώ η SQL είναι μια πιο δηλωτική γλώσσα. Στην SQL ο επιθυμητός σκοπός δηλώνεται συχνά με μια και μόνο εντολή. Η απλούστερη δομή της SQL είναι εφικτή επειδή καταπιάνεται μόνο με τις σχεσιακές βάσεις δεδομένων και όχι με την ολότητα των συστημάτων υπολογιστών (Rockoff L., 2017).

2.3.2 Το συντακτικό της SQL

Στη συνέχεια μέσω απλών παραδειγμάτων ακολουθεί μια παράθεση των βασικών εντολών της SQL. Σε όλα τα παραδείγματα θα χρησιμοποιηθεί η ίδια απλή βάση, το διάγραμμα της οποίας ακολουθεί:



Εικόνα 2.2: Το διάγραμμα των πινάκων της βάσης δεδομένων των παραδειγμάτων

Στη βάση αυτή έχουμε δύο πίνακες, τον πίνακα Users και τον πίνακα Orders. Οι δύο αυτοί πίνακες έχουν σχέση 1:N μεταξύ τους και ο δεύτερος συνδέεται με τον πρώτο με τη χρήση του ξένου κλειδιού user_id.

Δημιουργία Πίνακα

Ο πίνακας Users αποτελείται από τις στήλες: 1) user_id με τύπο δεδομένων INT και λόγω της εντολής NOT NULL δεν μπορεί να μείνει κενή κατά την εισαγωγή δεδομένων, 2) name με τύπο δεδομένων VARCHAR μήκους 15 χαρακτήρων. Έχει ως πρωτεύων κλειδί τη στήλη user_id:

```
CREATE TABLE Users (  
    user_id INT NOT NULL,  
    name VARCHAR(15),  
    CONSTRAINT pk_users PRIMARY KEY (user_id)
```

```
);
```

Ο πίνακας Orders αποτελείται από τις στήλες: 1) order_id με τύπο δεδομένων INT και λόγω της εντολής NOT NULL δεν μπορεί να μείνει κενή κατά την εισαγωγή δεδομένων, 2) order_date με τύπο δεδομένων DATE και 3) user_id με τύπο δεδομένων INT και λόγω της εντολής NOT NULL δεν μπορεί να μείνει κενή. Η στήλη order_id είναι το πρωτεύων κλειδί, η user_id είναι το ξένο κλειδί και αναφέρεται στον πίνακα Users:

```
CREATE TABLE Orders (  
    order_id INT NOT NULL,  
    order_date DATE,  
    user_id INT NOT NULL,  
    CONSTRAINT pk_orders PRIMARY KEY (order_id),  
    CONSTRAINT fk_orders FOREIGN KEY (user_id)  
    REFERENCES Users(user_id)  
);
```

Παρουσίαση της γενικής μορφής των παραδειγμάτων:

```
CREATE TABLE όνομα_πίνακα (  
    όνομα_στήλης τύπος_δεδομένων [ιδιότητες],  
    CONSTRAINT όνομα_περιορισμού [τύπος_περιορισμού  
    [περιορισμός]]);
```

Τροποποίηση Πίνακα

Προσθήκη της στήλης address στον πίνακα Users:

```
ALTER TABLE Users  
    ADD address VARCHAR(15);
```

Παρουσίαση της γενικής μορφής του παραδείγματος:

```
ALTER TABLE όνομα_πίνακα  
    ADD όνομα_στήλης τύπος_δεδομένων [ιδιότητες];
```

Διαγραφή Πίνακα

Διαγραφή του πίνακα με όνομα Users:


```
DROP TABLE Users;
```

Παρουσίαση της γενικής μορφής του παραδείγματος:

```
DROP TABLE όνομα_πίνακα;
```

Εισαγωγή Δεδομένων

Εισαγωγή τριών εγγραφών μαζί στον πίνακα Users:

```
INSERT INTO Users (user_id, name)
VALUES (0, 'John Doe'),
       (1, 'Alice Bar'),
       (2, 'Bob Foo');
```

Εισαγωγή τεσσάρων εγγραφών μαζί στον πίνακα Orders:

```
INSERT INTO Orders (order_id, order_date, user_id)
VALUES (0, '01-02-2016', 0),
       (1, '02-02-2016', 1),
       (2, '04-02-2016', 0),
       (3, '08-02-2016', 2);
```

Μετά την εισαγωγή των δεδομένων οι πίνακες έχουν την παρακάτω μορφή:

Users

| user_id | name |
|---------|------|
|---------|------|

| | |
|---|-----------|
| 0 | John Doe |
| 1 | Alice Bar |
| 2 | Bob Foo |

Orders

| order_id | order_date | user_id |
|----------|------------|---------|
| 0 | 01-02-2016 | 0 |
| 2 | 04-02-2016 | 0 |
| 1 | 02-02-2016 | 1 |
| 3 | 08-02-2016 | 2 |

Παρουσίαση της γενικής μορφής του παραδείγματος:

```
INSERT INTO όνομα_πίνακα (όνομα_στήλης1, όνομα_στήλης2,...όνομα_στήληςN)
```

```
VALUES (τιμή_στήλης1, τιμή_στήλης2,...τιμή_στήληςN) [,
        (τιμή_στήλης1, τιμή_στήλης2,...τιμή_στήληςN) ,
        ...
        (τιμή_στήλης1, τιμή_στήλης2,...τιμή_στήληςN) ] ;
```

Τροποποίηση Δεδομένων

Αλλαγή στη στήλη order_date για την εγγραφή με order_id = 3:

```
UPDATE Orders
SET order_date = '10-02-2016'
WHERE order_id = 3;
```

Παρουσίαση της γενικής μορφής του παραδείγματος:

```
UPDATE όνομα_πίνακα
SET όνομα_στήλης = τιμή
WHERE συνθήκη_αναζήτησης;
```

Διαγραφή Δεδομένων

Διαγραφή των εγγραφών που έχουν name = 'Alice Foobar':

```
DELETE
FROM Users
WHERE name = 'Alice Foobar';
```

Παρουσίαση της γενικής μορφής του παραδείγματος:

```
DELETE
FROM όνομα_πίνακα
WHERE συνθήκη_αναζήτησης;
```

Επιλογή Δεδομένων

Ερώτημα 1

Στο ερώτημα 1 θέλουμε να εμφανίσουμε όλες τις στήλες του πίνακα Users,:

```
SELECT *
FROM Users;
```

Ο πίνακας που προκύπτει από το ερώτημα:

| user_id | name |
|---------|-----------|
| 0 | John Doe |
| 1 | Alice Bar |
| 2 | Bob Foo |

Ερώτημα 2

Στο ερώτημα 2 θέλουμε να εμφανίσουμε όλες τις στήλες του πίνακα Orders και το αποτέλεσμα θέλουμε να εμφανιστεί ταξινομημένο ως προς τη στήλη user_id κατά αύξουσα σειρά:

```
SELECT *  
FROM Orders  
ORDER BY user_id ASC;
```

Ο πίνακας που προκύπτει από το ερώτημα:

| order_id | order_date | user_id |
|----------|------------|---------|
| 0 | 01-02-2016 | 0 |
| 2 | 04-02-2016 | 0 |
| 1 | 02-02-2016 | 1 |
| 3 | 10-02-2016 | 2 |

Ερώτημα 3

Στο ερώτημα 3 θέλουμε να εμφανίσουμε τις στήλες order_id και user_id από τον πίνακα Orders για εκείνες τις εγγραφές που έχουν user_id διαφορετικό του 0:

```
SELECT order_id, user_id  
FROM Orders  
WHERE user_id <> 0;
```

Ο πίνακας που προκύπτει από το ερώτημα:

| order_id | user_id |
|----------|---------|
| 1 | 1 |
| 3 | 2 |

Παρουσίαση της γενικής μορφής των παραδειγμάτων:

```
SELECT όνομα_στήλης1, όνομα_στήλης2,...όνομα_στήληςN
FROM όνομα_πίνακα
[WHERE συνθήκης_αναζήτησης];
```

Εσωτερική σύζευξη (Inner Join)

Με την εσωτερική σύζευξη στο παρακάτω ερώτημα θα εμφανιστούν οι εγγραφές που στην κοινή στήλη των πινάκων Users και Orders (user_id) έχουν την ίδια τιμή:

```
SELECT u.user_id, u.name, o.order_id, o.order_date
FROM Users AS u
     INNER JOIN
     Orders AS o
     ON u.user_id = o.user_id;
```

Ο πίνακας που προκύπτει από το ερώτημα:

| user_id | name | order_id | order_date |
|---------|-----------|----------|------------|
| 0 | John Doe | 0 | 01-02-2016 |
| 1 | Alice Bar | 1 | 04-02-2016 |
| 0 | John Doe | 2 | 02-02-2016 |
| 2 | Bob Foo | 3 | 10-02-2016 |

Παρουσίαση της γενικής μορφής του παραδείγματος:

```
SELECT όνομα_στήλης1, όνομα_στήλης2,...όνομα_στήληςN
FROM όνομα_πίνακα1 [[AS] ψευδώνυμο1]
     τύπος_σύζευξης
     όνομα_πίνακα2 [[AS] ψευδώνυμο2]
     ON ψευδώνυμο1.στήληX = ψευδώνυμο2.στήληX;
```

2.4 PostgreSQL

Η επιλογή του κατάλληλου Συστήματος Διαχείρισης Βάσης Δεδομένων είναι μια δύσκολη διαδικασία λόγω της πληθώρας επιλογών. Ανάλογα με τις ανάγκες κάποιος μπορεί να επιλέξει ένα εμπορικό σύστημα διαχείρισης βάσης δεδομένων ή ένα ανοικτού λογισμικού με προαιρετική εμπορική υποστήριξη. Όταν έρχεται η ώρα της επιλογής ενός Σχεσιακού Συστήματος Διαχείρισης Βάσης Δεδομένων (RDMS -

Relational Database Management System) η PostgreSQL είναι μια από τις κορυφαίες επιλογές. Το σλόγκαν της PostgreSQL “The world's most advanced open source database” («Η πιο προχωρημένη βάση δεδομένων ανοικτού λογισμικού στον κόσμο») δείχνει τον εκλεπτυσμό και την αυτοπεποίθηση της κοινότητάς της (Salahaldin et al., 2015).

Η PostgreSQL δίνει έμφαση στην επεκτασιμότητα και ανταγωνίζεται με τα κύρια συστήματα σχεσιακών βάσεων δεδομένων όπως η Oracle, ο SQL Server και η MySQL. Λόγω των πολλών επεκτάσεών της και της άδειας χρήσης ανοικτού λογισμικού χρησιμοποιείται πολλές φορές για ερευνητικούς σκοπούς. Ακόμα, πολλές νεοφυείς επιχειρήσεις συχνά προτιμούν την PostgreSQL λόγω του κόστους των αδειών χρήσης και της ύπαρξης πολλών επιχειρήσεων που προσφέρουν εμπορική υποστήριξη. Η PostgreSQL τρέχει στα περισσότερα Λειτουργικά Συστήματα συμπεριλαμβανομένων των Linux, Windows και Mac (Salahaldin et al., 2015).

Η PostgreSQL χρησιμοποιείται σε πολλά πεδία εφαρμογών συμπεριλαμβανομένων των τηλεπικοινωνιών, του κλάδου υγείας, των γεωγραφικών πληροφοριακών συστημάτων και των εφαρμογών ηλεκτρονικού εμπορίου (Salahaldin et al., 2015).

Ακολουθούν μερικές από τις εταιρίες που χρησιμοποιούν την PostgreSQL:

- Το Skype χρησιμοποιεί την PostgreSQL για την αποθήκευση των συζητήσεων και των δραστηριοτήτων των χρηστών του.
- Το Instagram που δίνει τη δυνατότητα στους χρήστες του να μοιράζονται εικόνες και φωτογραφίες.
- Ο Σύλλογος Χημικών της Αμερικής χρησιμοποιεί την PostgreSQL για να αποθηκεύσει περισσότερο από 1TB δεδομένων για το αρχείο των περιοδικών του.

Εκτός αυτών, η PostgreSQL χρησιμοποιείται από τις HP, VMware και Heroku. Επίσης, χρησιμοποιείται από πολλές επιστημονικές κοινότητες και οργανισμούς όπως η NASA (Salahaldin et al., 2015).

2.5 Η Object-Relational Mapping (ORM) εργαλειοθήκη SQLAlchemy

Η SQLAlchemy είναι μια εργαλειοθήκη η οποία χρησιμοποιείται για την αλληλεπίδραση με μια μεγάλη ποικιλία συστημάτων βάσεων δεδομένων. Επιτρέπει τη δημιουργία μοντέλων δεδομένων και επερωτήσεων με ένα τρόπο που μοιάζει με τις κανονικές κλάσεις και εντολές της Python. Δημιουργήθηκε από τον Mike Bayer το 2005, χρησιμοποιείται από πολλές μικρές και μεγάλες εταιρίες και θεωρείται ένας από τους εκ των πραγμάτων τρόπους να δουλεύει κανείς με τις σχεσιακές βάσεις στην Python (Myers & Copeland, 2016).

Μπορεί να χρησιμοποιηθεί για να συνδεθεί με τις πιο κοινές βάσεις όπως η PostgreSQL, η MySQL, η SQLite, ο SQL Server, η Oracle και πολλές άλλες. Επίσης,

προσφέρει ένα τρόπο για να υποστηριχθούν και άλλες σχεσιακές βάσεις. Η Amazon Redshift η οποία χρησιμοποιεί μια ειδική διάλεκτο της PostgreSQL είναι ένα παράδειγμα υποστήριξης βάσης που προήλθε από την κοινότητα της SQLAlchemy (Myers & Copeland, 2016).

Ο βασικότερος λόγος για να χρησιμοποιήσει κανείς την SQLAlchemy είναι ότι δημιουργεί αφαιρέσεις στον πηγαίο κώδικα για τη βάση δεδομένων που χρησιμοποιείται και για τις αντίστοιχες ιδιομορφίες της SQL. Η SQLAlchemy προσφέρει πολυδύναμες συνήθεις εντολές και τύπους για να βεβαιώσει ότι οι SQL εντολές της κατασκευάζονται αποδοτικά και αξιόπιστα για κάθε τύπο βάσης δεδομένων. Αυτό κάνει εύκολο το να μεταφέρει κανείς τη λογική από την Oracle για παράδειγμα στην PostgreSQL ή από μια εφαρμογή βάσης δεδομένων σε μια αποθήκη δεδομένων (data warehouse). Επίσης, βοηθάει στο να εξασφαλίσει ότι οι εισοδοί στη βάση δεδομένων ελέγχονται με ασφαλή τρόπο πριν να εκτελεστούν στη βάση. Αυτό εξαλείφει από συχνά θέματα ασφαλείας, όπως οι επιθέσεις SQL injection (Myers & Copeland, 2016).

Τέλος, η SQLAlchemy προσφέρει αρκετή ευελιξία μέσω της υποστήριξης δύο καταστάσεων λειτουργίας: της SQL Expression Language (γνωστή και ως SQLAlchemy Core) και της ORM. Αυτές οι καταστάσεις μπορούν να χρησιμοποιηθούν ξεχωριστά ή και μαζί ανάλογα με τις προτιμήσεις και τις ανάγκες της εφαρμογής (Myers & Copeland, 2016).

2.6 Nginx

Ο Nginx (προφέρεται engine X) είναι ένας HTTP διακομιστής. Η ανάπτυξή του ξεκίνησε το 2002 από τον Igor Sysoen με σκοπό να εξυπηρετήσει τις ανάγκες μιας Ρωσικής ιστοσελίδας η οποία είχε 500 εκατομμύρια HTTP αιτήματα την ημέρα το 2008. Ο Nginx χρησιμοποιείται από δημοφιλείς ιστοσελίδες όπως το Facebook, το Netflix και άλλες. Θεωρείται ένας διακομιστής πολύ αποδοτικός, ελαφρύς, με υψηλή απόδοση και διατίθεται με την άδεια 2-clause BSD.

2.7 Gunicorn

Ο Gunicorn είναι ένας Python Web Server Gateway Interface (WSGI) HTTP διακομιστής για τα λειτουργικά συστήματα UNIX. Υποστηρίζει εγγενώς την προδιαγραφή Web Server Gateway Interface (WSGI) και τα frameworks Django και web2py. Επίσης, έχει αυτόματη διαχείριση των διεργασιών των workers, για την ακρίβεια υπάρχει μια κεντρική κύρια διεργασία η οποία διαχειρίζεται τους workers. Είναι συμβατός με τις εκδόσεις της Python 2.6+ και 3.2+. Τέλος, διατίθεται με την άδεια MIT.

2.8 Η γλώσσα προγραμματισμού JavaScript

2.8.1 Εισαγωγή

Η JavaScript αναπτύχθηκε από τον Brendan Eich, ένα μηχανικό λογισμικού στη Netscape Communications Corporation το 1995. Η αρχική της ανάπτυξη ήταν πολύ γρήγορη και ένα μεγάλο μέρος της κριτικής που δέχεται η JavaScript προέρχεται από την έλλειψη διορατικότητας κατά τον αρχικό της σχεδιασμό. Ωστόσο, ο Brendan Eich είχε ένα ισχυρό υπόβαθρο στην επιστήμη των υπολογιστών και συμπεριέλαβε εκλεπτυσμένες ιδέες στην JavaScript. Η JavaScript ξεκίνησε με το όνομα Mocha και γρήγορα άλλαξε σε LiveScript πριν μετονομαστεί επίσημα σε JavaScript κατά την κυκλοφορία του Netscape Navigator το 1995. Η λέξη Java στο JavaScript δεν είναι συμπωματική, αν και μπερδεύει, εκτός από τη συντακτική συγγένεια με τη Java, η JavaScript έχει πιο πολλά κοινά με τη Self και τη Scheme. Το όνομα JavaScript ήταν μερικώς μια προσπάθεια του μάρκετινγκ να τραβήξει την προσοχή λόγω της επιτυχίας που είχε η Java εκείνη την περίοδο (Brown, 2016).

Το Νοέμβριο του 1996 η Netscape ανακοίνωσε ότι υπέβαλλε τη JavaScript στην ECMA, ένας ιδιωτικός, διεθνής μη κερδοσκοπικός οργανισμός προτύπων που έχει μεγάλη επιρροή στο χώρο των επιχειρήσεων τεχνολογίας και τηλεπικοινωνιών. Η ECMA δημοσίευσε την πρώτη έκδοση της προδιαγραφής ECMA-26, η οποία ήταν ουσιαστικά η JavaScript. Τεχνικά η JavaScript είναι μια εφαρμογή της ECMAScript, αλλά για πρακτικούς λόγους οι όροι JavaScript και ECMAScript μπορούν να χρησιμοποιηθούν για να περιγράψουν την ίδια γλώσσα. Η τελευταία έκδοση της ECMAScript είναι η 6 (ES6) η οποία εκδόθηκε τον Ιούνιο του 2015 (Brown, 2016).

2.8.2 Συντακτικό

2.8.2.1 Μεταβλητές

Στην ES6 μπορούμε να δηλώσουμε τις μεταβλητές με δύο τρόπους, ο ένας είναι χρησιμοποιώντας το *let* πριν από το όνομα της μεταβλητής και ο άλλος χρησιμοποιώντας το *var*. Η διαφορά μεταξύ αυτών των δύο λύσεων βρίσκεται στην εμβέλειά (scope) τους. Στην περίπτωση του *let* η ισχύς της εμβέλειας βρίσκεται μέσα σε ένα μπλοκ κώδικα που περικλείεται από άγκιστρα (block scope), ενώ στην περίπτωση του *var* η ισχύς βρίσκεται μέσα στην καθολική εμβέλεια (global scope) ή στην εμβέλεια μιας συνάρτησης (function scope). Μια συνάρτηση, όπως θα δούμε αναλυτικότερα παρακάτω, τη δηλώνουμε ως εξής: `function όνομα_συνάρτησης (παράμετροι){...}`. Επίσης, τα σχόλια στη JavaScript μπορούν να είναι μιας γραμμής (`//`) ή πολλαπλών (`/* ... */`).

Παραδείγματα δήλωσης μεταβλητών:

let

```
function letTest(){
  let x = 1;
  if (true){
    let x = 2;
    console.log(x); // εκτυπώνει 2
  }
  console.log(x); // εκτυπώνει 1
}
```

var

```
var x = 1;
```

```
function foo(){
  var x = 2;
  var y = 3;
  console.log(x); // εκτυπώνει 2
  console.log(y); // εκτυπώνει 3
}
```

```
foo();
console.log(x); // εκτυπώνει 1
console.log(y); // ReferenceError: y is not defined
```

Επίσης, στην ES6 μπορούμε να δηλώσουμε σταθερές χρησιμοποιώντας το *const* κατά τη δήλωση της σταθεράς. Η εμβέλεια μιας *const* είναι ίδια με αυτή της *let*.

Παράδειγμα δήλωσης σταθεράς:

```
const y = 1;
if (true){
  const z = 2;
  console.log(y); // εκτυπώνει 1
  console.log(z); // εκτυπώνει 2
}

console.log(y); // εκτυπώνει 1
```



```
console.log(z); // ReferenceError: z is not defined
y = 3; //TypeError: invalid assigment to const 'y'
```

2.8.2.2 Τύποι Δεδομένων

Οι τύποι δεδομένων της ECMAScript είναι οι εξής:

- Undefined
- Null
- Boolean
- Number
- String
- Object

Για να βρούμε τον τύπο μιας μεταβλητής μπορούμε να χρησιμοποιήσουμε τον τελεστή *typeof*, για παράδειγμα:

```
const s = 'asdf';
const u = undefined;
const b = true;
const i = 1;
const n = null;
const o = new Object();

console.log(typeof s); // εκτυπώνει string
console.log(typeof u); // εκτυπώνει undefined
console.log(typeof b); // εκτυπώνει boolean
console.log(typeof i); // εκτυπώνει number
console.log(typeof n); // εκτυπώνει object
console.log(typeof o); // εκτυπώνει object
```

Ο τύπος Null (μεταβλητή *n*) εκτυπώνει “object”, επειδή η τιμή *null* είναι ένας κενός δείκτης αντικειμένου.

Ο τελεστής *typeof* ελέγχει για τον τύπο της τιμής που υπάρχει εκείνη τη στιγμή, για παράδειγμα:

```
let a = 'one';
console.log(typeof a); //εκτυπώνει string
a = 1;
console.log(typeof a); // εκτυπώνει number
```

Όπως φαίνεται και στο παράδειγμα αν ο τύπος της τιμής αλλάξει, τότε αλλάζει και το αποτέλεσμα που μας δίνει ο τελεστής *typeof*. Από αυτό το παράδειγμα φαίνεται και μια βασική ιδιότητα που υποστηρίζει η JavaScript, το *dynamic typing*. Η δυνατότητα δηλαδή που έχουμε να εκχωρούμε στην ίδια μεταβλητή τιμές με διαφορετικούς τύπους δεδομένων. Στις γλώσσες που υποστηρίζουν το *dynamic typing* (Python, JavaScript, Ruby) ο έλεγχος του τύπου γίνεται κατά το *run-time* και όχι κατά το *compile-time*, όπως συμβαίνει στις γλώσσες που υποστηρίζουν το *static typing* (C, C++, Java, Go).

2.8.2.3 Συμβολοσειρές

Ο τύπος *String* αναπαριστά μια ακολουθία από μηδέν ή περισσότερους 16-bit Unicode χαρακτήρες. Οι τιμές των συμβολοσειρών μπορούν να γραφούν μέσα σε διπλά εισαγωγικά (“...”) ή σε μονά (‘...’). Μάλιστα λόγω της υποστήριξης του Unicode μπορούμε να γράψουμε απευθείας Ελληνικούς χαρακτήρες, χωρίς να χρειαστεί να κάνουμε κάποια επιπλέον ενέργεια.

Παράδειγμα

```
const a = "Ακρόπολη";  
const b = 'Ακρόπολη';
```

Αν θέλουμε μια συμβολοσειρά να τη διανείμουμε σε πολλαπλές γραμμές τότε θα πρέπει να χρησιμοποιήσουμε την ανάστροφη κάθετο (\) στο τέλος της γραμμής

```
const s = "1η γραμμή,\n2η γραμμή";  
console.log(s); // εκτυπώνει 1η γραμμή,2η γραμμή
```

Στην ES6 με τη χρήση του *backtick* (‘) είναι εφικτό να γράψουμε τιμές συμβολοσειρών σε πολλαπλές γραμμές και να κάνουμε μορφοποίηση χρησιμοποιώντας τις τιμές μεταβλητών σε όποιο σημείο της συμβολοσειράς επιλέξουμε.

Παράδειγμα συμβολοσειράς πολλαπλών γραμμών

```
const b = `1η γραμμή,  
2η γραμμή`;  
console.log(b); // εκτυπώνει 1η γραμμή,  
                //                2η γραμμή
```

Μια διαφορά που έχει η χρήση του *backtick* είναι ότι σε αυτή τη περίπτωση λαμβάνεται υπόψιν όλο το κενό διάστημα, για αυτό και το “2η γραμμή” εκτυπώνεται σε άλλη σειρά.

Παράδειγμα μορφοποίησης συμβολοσειράς

```
let name = 'world';
    msg = `Hello ${name}`;
console.log(msg); // εκτυπώνει Hello world
```

Ακόμα, είναι εφικτό να μετατρέψουμε μια τιμή σε συμβολοσειρά, για παράδειγμα:

```
let i = 1;
let iStr = i.toString(); // συμβολοσειρά "1"
```

2.8.2.4 Αντικείμενα

Ο τύπος αντικείμενο είναι ένας από τους πιο χρησιμοποιούμενους τύπους στη JavaScript. Τα αντικείμενα μπορούν να αναπαραστήσουν πολλαπλές ή σύνθετες τιμές και να τις αλλάζουν κατά τη διάρκεια της χρήσης τους. Κατά κάποιο τρόπο ένα αντικείμενο μπορεί να χαρακτηριστεί ως μια συλλογή και τα περιεχόμενα του μπορούν να αλλάζουν με τον καιρό.

Παράδειγμα δήλωσης ενός κενού αντικειμένου:

```
let o = {};
```

Παράδειγμα δήλωσης ενός αντικειμένου με τις ιδιότητές του:

```
let vehicle = {
    type: 'truck',
    wheels: 6
};
```

Στο παραπάνω παράδειγμα έχουμε το αντικείμενο vehicle, με τις ιδιότητες type και wheels. Αν θέλουμε πρόσβαση σε μια ιδιότητα του αντικειμένου μπορούμε να χρησιμοποιήσουμε το συμβολισμό τελείας ., όπως στο παρακάτω παράδειγμα που αλλάζουμε την τιμή της ιδιότητας wheels.

```
vehicle.wheels = 8;
```

2.8.2.5 Πίνακες

Μετά τον τύπο αντικείμενο, ο τύπος πίνακα (array) είναι ο πιο διαδεδομένος. Σε αντίθεση με άλλες γλώσσες που μπορούν σε κάθε πίνακα να έχουν μόνο ένα τύπο δεδομένου, στη JavaScript οι πίνακες μπορούν να δεχθούν οποιοδήποτε τύπο. Στο επόμενο παράδειγμα εμφανίζεται ο τρόπος δημιουργίας ενός πίνακα με διαφορετικούς τύπους δεδομένων.

```
let arr = [1, 'one', true];
```

Ένας πίνακας μπορεί να δημιουργηθεί δηλώνοντας το αρχικό του μήκος και ως εξής:

```
let a = new Array(10);  
console.log(a.length); // εκτυπώνει 10, το μήκος του a
```

Ένας πίνακας στη JavaScript έχει και κάποιες μεθόδους οι οποίες εφαρμόζονται σε αυτή τη δομή δεδομένων.

Στη συνέχεια εμφανίζονται κάποιες μέθοδοι που χρησιμοποιούνται συχνά, κυρίως όταν χρησιμοποιούνται ως δομές δεδομένων οι σιόιβες και οι ουρές.

```
let n = ['One', 'Two', 'Three'];  
  
// προσθήκη ενός στοιχείου στο τέλος  
n.push('Four');  
  
// αφαίρεση ενός στοιχείου από το τέλος ('Four')  
n.pop();  
  
// εισαγωγή στοιχείου στην αρχή  
n.unshift('Zero');  
  
// αφαίρεση του 1ου στοιχείου ('Four')  
n.shift();  
  
/* αφαίρεση 2 στοιχείων ξεκινώντας από το δείκτη 1  
και μεταφορά τους στον πίνακα subAr, έτσι έχουμε:  
n = [ 'One' ]  
subAr = [ 'Two', 'Three' ]*/  
let subAr = n.splice(1, 2);
```

2.8.2.6 Έλεγχος ροής

Στη συνέχεια παρατίθενται οι βασικές δομές των εντολών ελέγχου της ροής της JavaScript.

Εντολή if

Η εντολή if έχει την παρακάτω γενική δομή:

```
if (condition){
    statement(s);
} else if (condition){
    statement(s);
} else {
    statement(s);
}
```

Εντολή switch

Η γενική δομή της switch:

```
switch (expression) {
    case value: statement(s)
    break;
    case value: statement(s)
    break;
    case value: statement(s)
    break;
    case value: statement(s)
    break;
    default: statement(s)
}
```

Εντολή for

Η εντολή for έχει την παρακάτω γενική δομή:

```
for (initialization; expression; post-loop-expression){
    statement(s)
}
```

Εντολή while

Η εντολή while έχει την παρακάτω γενική δομή:

```
while (expression){  
    statements(s)  
}
```

Εντολή do-while

Η εντολή do-while έχει την παρακάτω γενική δομή:

```
do {  
    statement(s)  
} while (expression);
```

2.8.2.7 Συναρτήσεις

Οι συναρτήσεις στην JavaScript είναι ένα από τα πιο ισχυρά χαρακτηριστικά της, πολλές φορές όμως προκαλούν σύγχυση σε προγραμματιστές που έρχονται από άλλες γλώσσες. Αυτό συμβαίνει επειδή, σε αντίθεση με άλλες γλώσσες, οι συναρτήσεις στη JavaScript είναι στην πραγματικότητα αντικείμενα. Κάθε συνάρτηση είναι ένα στιγμιότυπο του τύπου Function το οποίο έχει ιδιότητες και μεθόδους όπως κάθε άλλος τύπος αναφοράς. Λόγω του ότι οι συναρτήσεις είναι αντικείμενα, τα ονόματα των συναρτήσεων είναι δείκτες σε αντικείμενα συναρτήσεων (Zakas, 2012).

Μια συνάρτηση μπορεί να οριστεί ως εξής:

```
function functionName (arguments){  
    statement(s)  
}
```

Για παράδειγμα:

```
function sum(a, b){  
    return a + b;  
}
```

Η συνάρτηση αυτή επιστρέφει το άθροισμα των δύο ορισμάτων της. Εξαιτίας όμως της φύσης των συναρτήσεων στη JavaScript, η παραπάνω σύνταξη είναι σχεδόν ισοδύναμη με το να χρησιμοποιήσουμε μια έκφραση συνάρτησης, όπως παρακάτω:

```
let sum = function (a, b){  
    return a + b;  
};
```

Στη συνέχεια επειδή η μεταβλητή *sum* ορίστηκε ως μια συνάρτηση μπορούμε να την χρησιμοποιήσουμε ως τέτοια:

```
console.log(sum(1, 2));
```

2.8.2.8 Κλάσεις

Σε αντίθεση, με τις περισσότερες καθαρά αντικειμενοστρεφείς γλώσσες, η JavaScript μέχρι την προδιαγραφή ES5 δεν υποστήριζε τη δημιουργία κλάσεων ως τον πρωτεύοντα τρόπο με τον οποίο ορίζονται παρόμοια αντικείμενα. Αυτό άλλαξε στην ES6, όπου μπορεί να οριστεί συντακτικά το μοτίβο μιας κλάσης. Για παράδειγμα:

```
class Point {  
  
    constructor(x, y){  
        this.x = x;  
        this.y = y;  
    }  
  
    calculateDistance(other){  
        return Math.sqrt(Math.pow(this.x - other.x, 2) +  
                           Math.pow(this.y - other.y, 2));  
    }  
}  
  
const p1 = new Point(1,2);  
const p2 = new Point(2,4);  
console.log(p1.calculateDistance(p2)); // εκτυπώνει 2.23606797749979
```

Στο παραπάνω παράδειγμα βλέπουμε τη δημιουργία της κλάσης *Point* (ταυτόσημη με αυτή που ορίστηκε στο αντίστοιχο παράδειγμα της Python), η οποία δέχεται στον κατασκευαστή της (*constructor(x, y)*) τις δύο μεταβλητές ενός σημείου. Με βάση αυτές τις δύο μεταβλητές στιγμιότυπου (*this.x* και *this.y*) υπολογίζεται η απόσταση μεταξύ δύο αντικειμένων τύπου *Point* με τη χρήση της μεθόδου *calculateDistance()*. Στη συνέχεια δημιουργούμε δύο αντικείμενα τύπου *Point* (*p1* και *p2*) και καλούμε την *calculateDistance()* για να εκτυπώσουμε την απόσταση των *p1* και *p2*.

2.9 HyperText Markup Language (HTML)

Η HTML είναι η γλώσσα σήμανσης που χρησιμοποιούν οι φυλλομετρητές για να αναπαραστήσουν κείμενα και πολυμέσα. Η HTML αποτελείται από μια σειρά από στοιχεία. Για παράδειγμα έχουμε το παρακάτω στοιχείο:

```
<p> The quick brown fox jumps over the lazy dog </p>
```

Το <p> είναι η ετικέτα (tag) από την οποία ξεκινάει το στοιχείο και η ετικέτα </p> είναι η ετικέτα που το κλείνει. Το κείμενο μεταξύ των ετικετών είναι το περιεχόμενο του στοιχείου.

2.10 Cascading Style Sheets (CSS)

Η CSS χρησιμοποιείται για την παρουσίαση του περιεχομένου που έχει γραφτεί στην HTML. Μέσω της CSS μπορούμε να ελέγξουμε μεταξύ άλλων το χρώμα, τις διαστάσεις και τις αποστάσεις των στοιχείων ή ακόμα και να προσθέσουμε οπτικά εφέ.

Ένα αρχείο της CSS αποτελείται από κάποιους κανόνες, η βασική δομή ενός κανόνα είναι η παρακάτω:

```
selector-name {  
    property-name: property-value;  
}
```

Ένας κανόνας ελέγχει την εμφάνιση ενός στοιχείου της HTML, το ποιο θα είναι αυτό το στοιχείο επιλέγεται στην αρχή του κανόνα (selector-name), το πώς θα εμφανιστεί αυτό το στοιχείο ελέγχεται από μια ακολουθία ιδιοτήτων με τις αντίστοιχες τιμές τους (property-name: property-value).

2.11 Bootstrap

Το Bootstrap είναι ένα framework για την πλευρά του πελάτη (client-side) το οποίο βοηθά στη γρήγορη ανάπτυξη του σχεδιασμού μιας ιστοσελίδας. Η ανάπτυξή του ξεκίνησε από τους προγραμματιστές του Twitter, Mark Otto και Jacob Thornton. Κατά την ανάπτυξή του δόθηκε προτεραιότητα, όπως και στα περισσότερα frameworks, στη συνέπεια και στη συντηρησιμότητα του κώδικα. Στην αρχή, στην έκδοση 1.0.0 υπήρχαν μόνο στοιχεία για CSS και HTML. Όμως με την έκδοση 1.3.0 ήρθε και η παρουσία plugins της JavaScript.

Το Bootstrap έχει ένα δικό του σύστημα για την τοποθέτηση των στοιχείων (Grid system) το οποίο κάνει το σχεδιασμό της ιστοσελίδας να ανταποκρίνεται σε οθόνες όλων των διαστάσεων. Με άλλα λόγια, μέσω του Grid system είναι εφικτή η προσαρμογή του σχεδιασμού ανάλογα με τις διαστάσεις της οθόνης, για αυτό το

λόγο μπορούμε να έχουμε ένα σχεδιασμό που προσαρμόζεται στις διαστάσεις της μικρής οθόνης ενός κινητού τηλεφώνου και ταυτόχρονα εκμεταλλεύεται τον παραπάνω χώρο που υπάρχει σε μια οθόνη ενός σταθερού υπολογιστή.

Εκτός από την υποστήριξη των χαρακτηριστικών του Responsive Web Design, το Bootstrap έρχεται με πολλές κλάσεις οι οποίες διευκολύνουν το σχεδιασμό των πινάκων, των φορμών, των κουμπιών, των στοιχείων πλοήγησης και των στοιχείων ενημέρωσης του χρήστη. Τέλος, με σκοπό τη βελτίωση της διαδραστικότητας του σχεδιασμού, το Bootstrap περιέχει και πολλές συναρτήσεις της JavaScript, στις οποίες περιλαμβάνονται μεταξύ άλλων λειτουργίες εμφάνισης ενημερωτικών μηνυμάτων και ειδοποιήσεων στο χρήστη.

2.12 jQuery

Το jQuery είναι μια δωρεάν βιβλιοθήκη της JavaScript η οποία δημιουργήθηκε το 2006 από τον John Resig και διατίθεται υπό την άδεια MIT. Το jQuery δημιουργήθηκε με σκοπό την απλοποίηση της συγγραφής του κώδικα στην πλευρά του πελάτη. Σύμφωνα με την ιστοσελίδα trends.builtwith.com το jQuery χρησιμοποιείται από το 86.4% των ιστοσελίδων που ανήκουν στο 1^ο εκατομμύριο παγκοσμίως. Το Μάρτιο του 2017 κυκλοφόρησε η πιο νέα του έκδοση μέχρι τώρα, η 3.2.1.

Με τη χρήση του jQuery είναι δυνατή η πρόσβαση στα στοιχεία του Document Object Model (DOM), η αλλαγή της εμφάνισης μιας σελίδας με τις δυναμικές προσθαφαιρέσεις κλάσεων της CSS, η αλλαγή του περιεχομένου του DOM, η ανταπόκριση σε συμβάντα του φυλλομετρητή, η κίνηση των περιεχομένων μιας σελίδας και η επικοινωνία με το διακομιστή μέσω της τεχνικής AJAX.

2.13 Asynchronous JavaScript και XML (AJAX)

Το 2005 ο Jesse James Garrett δημοσίευσε ένα άρθρο με τίτλο “Ajax: A New Approach to Web Applications” (www.adaptivepath.com/ideas/essays/archives/000385.php). Σε αυτό το άρθρο αναφερόταν η τεχνική AJAX, με αυτήν την τεχνική μπορούσαν να γίνουν αιτήματα στο διακομιστή για επιπλέον δεδομένα χωρίς να επαναφορτωθεί η ιστοσελίδα, με αυτό τον τρόπο βελτιωνόταν η εμπειρία χρήσης. Στο άρθρο ο συγγραφέας εξηγούσε πως αυτή η τεχνική θα μπορούσε να αλλάξει τον παραδοσιακό τρόπο με τον οποίο κάποιος θα έπρεπε να περιμένει να φορτωθεί μια ολόκληρη καινούργια σελίδα για να δει νέο περιεχόμενο (Zakas, 2012).

Αυτή η καινοτομία έγινε εφικτή λόγω της εισαγωγής του *XMLHttpRequest* αντικειμένου στους φυλλομετρητές στις αρχές τις δεκαετίας του 2000 και προκάλεσε αυτό που μετά έγινε γνωστό ως Web 2.0.

Κατά βάση αυτό που κάνει το AJAX είναι απλό, μέσω της JavaScript αποστέλλεται ένα HTTP αίτημα (HTTP request) στο διακομιστή, ο οποίος με τη σειρά του επιστρέφει δεδομένα, συνήθως σε JSON μορφή. Τα δεδομένα αυτά χρησιμοποιούνται για επιπλέον λειτουργικότητα στο φυλλομετρητή. Λόγω του AJAX ο φόρτος εργασίας

που προκαλείται από τη μεταφορά και την παρουσίαση μια σελίδας είναι σημαντικά μειωμένος, επιτρέποντας έτσι στις εφαρμογές ιστού να τρέχουν γρηγορότερα (Brown, 2016).

2.14 JavaScript Object Notation (JSON)

Ο Douglas Crockford πρώτος μέσω του κειμένου IETF RFC 4627 το 2006 προσδιόρισε τη μορφή δεδομένων JSON. Το JSON είναι ένα αυστηρά ορισμένο υποσύνολο της JavaScript, χρησιμοποιώντας πολλά από τα μοτίβα που χρησιμοποιούνται στη JavaScript για την αναπαράσταση δεδομένων. Ο Crockford παρουσίασε το JSON ως μια καλύτερη εναλλακτική σε σχέση με το XML για την πρόσβαση σε δεδομένα στη JavaScript. Το πιο σημαντικό όσον αφορά στην κατανόηση του JSON είναι ότι είναι μια μορφή δεδομένων και όχι μια γλώσσα προγραμματισμού. Ακόμα, το JSON δεν είναι μέρος της JavaScript, παρόλο που μοιράζονται κάποιο κοινό συντακτικό. Επιπλέον, το JSON δεν χρησιμοποιείται μόνο με τη JavaScript, αφού πρόκειται για μια μορφή δεδομένων. Υπάρχουν εργαλεία για τη χρήση του JSON σε πολλές γλώσσες προγραμματισμού (Zakas, 2012).

Το συντακτικό του JSON επιτρέπει την αναπαράσταση τριών τύπων τιμών (Zakas, 2012):

- **Απλές τιμές**, όπως συμβολοσειρές, αριθμοί, τιμές δεδομένων Αληθείας (Booleans) και το null.
- **Αντικείμενα**, αντικείμενα που αναπαριστούν διατεταγμένα ζεύγη κλειδιών-τιμών.
- **Πίνακες**, πίνακες που αναπαριστούν μια διατεταγμένη λίστα από τιμές στις οποίες υπάρχει πρόσβαση μέσω αριθμητικών δεικτών.

Παράδειγμα ενός απλού JSON αντικειμένου:

```
{  
  "name": "Ferrari F40",  
  "horsepower": 478,  
}
```

Στα αριστερά έχουμε τις ιδιότητες (name και yearManufactured) και στα δεξιά τις τιμές των ιδιοτήτων ("Ferrari F40" και 1990 αντίστοιχα).

2.15 GeoJSON

Το GeoJSON είναι ένας μορφότυπος για την ενσωμάτωση διαφόρων γεωγραφικών δομών δεδομένων και βασίζεται στο μορφότυπο JSON. Το 2015 η IETF (Internet Engineering Task Force) δημιούργησε το πρότυπο GeoJSON. Το πρότυπο καθορίζεται στο έγγραφο RFC 7946, το οποίο δημοσιεύτηκε τον Αύγουστο του 2016. Οι γεωμετρικοί τύποι δεδομένων που διαθέτει είναι: *Point*, *LineString*, *Polygon*,

MultiPoint, *MultiLineString* και *MultiPolygon*. Επίσης, υποστηρίζει τη δημιουργία γεωμετρικών αντικειμένων με επιπρόσθετες ιδιότητες, τα οποία ονομάζονται *Feature*. Σύνολα από *Feature* αντικείμενα μπορούν να ενσωματωθούν σε αντικείμενα τύπου *FeatureCollection*.

Ακολουθεί ένα παράδειγμα ενός *FeatureCollection*, το οποίο περιέχει δύο *Feature* αντικείμενα τύπου *Point*:

```
{
  "type": "FeatureCollection",
  "features": [{
    "type": "Feature",
    "properties": {
      "name": "Somewhere 0"
    },
    "geometry": {
      "type": "Point",
      "coordinates": [
        25.6640625,
        37.7185903
      ]
    }
  },
  {
    "type": "Feature",
    "properties": {
      "name": "Somewhere 1"
    },
    "geometry": {
      "type": "Point",
      "coordinates": [
        23.203125,
        36.315125
      ]
    }
  }
]
```

Το σύστημα αναφοράς συντεταγμένων του GeoJSON είναι το World Geodetic System 1984 (WGS 84) και οι συντεταγμένες ορίζονται με τη σειρά γεωγραφικό μήκος (longitude) και γεωγραφικό πλάτος (latitude).

2.16 Geolocation API

2.16.1 Γενικά

Το Geolocation API είναι το αποτέλεσμα των ενεργειών του οργανισμού W3C (World Wide Web Consortium) για τη δημιουργία ενός προτύπου για τις υπηρεσίες εύρεσης

τοποθεσίας που χρησιμοποιούνται στην πλευρά του πελάτη (client-side). Με την αύξηση της διείσδυσης των κινητών συσκευών στην καθημερινότητα ήρθε και η αύξηση των εφαρμογών που κάνουν χρήση των πληροφοριών της τοποθεσίας. Οι αισθητήρες GPS (Global Positioning System) πλέον βρίσκονται σε όλα τα έξυπνα κινητά τηλέφωνα, μέσω αυτών των αισθητήρων είναι δυνατή η εύρεση της θέσης του χρήστη με σχετικά υψηλή ακρίβεια. Όταν δεν είναι εφικτή η εύρεση της θέσης μέσω του GPS τότε οι συσκευές βρίσκουν την τοποθεσία χρησιμοποιώντας πληροφορίες από το δίκτυο της κινητής, το δίκτυο Wi-Fi και τη διεύθυνση IP (Crowther et al., 2014).

Μέσω της Geolocation API ορίζονται μέθοδοι οι οποίες επιτρέπουν στις εφαρμογές ιστού την εύρεση της τοποθεσίας του χρήστη. Όταν κάποια από αυτές τις μεθόδους καλείται ο χρήστης ειδοποιείται για το αίτημα εύρεσης της θέσης του από την εφαρμογή, τότε ο χρήστης μπορεί να δεχθεί ή να απορρίψει αυτό το αίτημα (Crowther et al., 2014). Το σύστημα συντεταγμένων που χρησιμοποιείται είναι το WG84.

Τα δεδομένα που δίνει τη δυνατότητα να βρεθούν η Geolocation API είναι: το γεωγραφικό μήκος και πλάτος, το υψόμετρο, η κατεύθυνση, η ταχύτητα και η ακρίβεια αυτών των τιμών (Crowther et al., 2014). Επίσης, σε περίπτωση που συμβεί κάποιο σφάλμα υπάρχουν οι αντίστοιχοι κωδικοί τους οποίους μπορεί να χρησιμοποιήσει ο προγραμματιστής της εφαρμογής για την παροχή πληροφοριών στο χρήστη σχετικά με τη μη λειτουργία της εφαρμογής και για πιθανούς τρόπους επίλυσης.

2.16.2 Χρήση του Geolocation API

Στη συνέχεια ακολουθεί μια σύντομη περιγραφή των αντικειμένων, των ιδιοτήτων και των μεθόδων του Geolocation API που χρησιμοποιούνται.

navigator.geolocation

Το Geolocation API χρησιμοποιείται μέσω του αντικειμένου `navigator.geolocation`. Αν το συγκεκριμένο αντικείμενο υπάρχει στο φυλλομετρητή (browser), τότε υπάρχει πρόσβαση στις υπηρεσίες του Geolocation.

navigator.geolocation.getCurrentPosition(successCallback, errorCallback, [options])

Η μέθοδος `getCurrentPosition()` χρησιμοποιείται με σκοπό να βρεθεί η θέση της συσκευής εκείνη τη στιγμή. Σε περίπτωση που βρεθεί επιτυχώς, τότε επιστρέφει ένα αντικείμενο `Position` και καλείται η συνάρτηση `successCallback`. Σε περίπτωση που δεν μπορέσει να βρεθεί η θέση επιτυχώς, καλείται η `errorCallback`. Η παράμετρος `options` ορίζει κάποιες επιλογές που μπορούν να γίνουν σχετικά με τον εντοπισμό της θέσης.

navigator.geolocation.watchPosition(successCallback [, errorCallback [, options]])

Η μέθοδος `watchPosition()` χρησιμοποιείται με σκοπό να οριστεί μια συνάρτηση η οποία θα καλείται κάθε φορά που η θέση της συσκευής αλλάζει (`successCallback`). Σε περίπτωση που υπάρξει σφάλμα τότε μπορεί να οριστεί μια `errorCallback` συνάρτηση για το χειρισμό αυτού του ενδεχομένου. Όπως και με την `getCurrentPosition()`, η παράμετρος `options` ορίζει κάποιες επιλογές που μπορούν να γίνουν σχετικά με τον εντοπισμό της θέσης.

Ιδιότητες της διεπαφής Coordinates του αντικειμένου Position

latitude

Μια τιμή τύπου `double` που αναπαριστά το γεωγραφικό πλάτος σε μοίρες σε δεκαδική μορφή.

longitude

Μια τιμή τύπου `double` που αναπαριστά το γεωγραφικό μήκος σε μοίρες δεκαδικής μορφής.

speed

Μια τιμή τύπου `double` που αναπαριστά την ταχύτητα κίνησης σε μέτρα ανά δευτερόλεπτο.

accuracy

Μια τιμή τύπου `double` που αναπαριστά την ακρίβεια των τιμών του γεωγραφικού πλάτους και μήκους σε μέτρα.

altitude

Μια τιμή τύπου `double` που αναπαριστά το ύψος σε μέτρα σε σχέση με το επίπεδο της θάλασσας.

altitudeAccuracy

Μια τιμή τύπου `double` που αναπαριστά την ακρίβεια της τιμής του ύψους σε μέτρα.

heading

Μια τιμή τύπου `double` που αναπαριστά την κατεύθυνση κατά την οποία η συσκευή κινείται.

2.17 Google Maps API

2.17.1 Γενικά

Η διεπαφή Google Maps βγήκε σε κυκλοφορία το έτος 2005 από την Google με στόχο να γίνει εύκολη η εισαγωγή των Google Maps στις ιστοσελίδες. Αρχικά οι Google Maps ξεκίνησαν από δύο αδέρφια από τη Δανία, τους Lars και Jens Eilstrup Rasmussen, στη συνέχεια το 2004 η εταιρία των δύο αδερφιών εξαγοράστηκε από τη Google. Εκτός από το JavaScript API που χρησιμοποιείται στην παρούσα εργασία υπάρχουν εκδόσεις των Google Maps και για κινητά τηλέφωνα με Λειτουργικό Σύστημα Android και iOS. Η Google Maps JavaScript API δεν είναι δωρεάν, έχει όμως ένα όριο 25,000 φορτώσεων του χάρτη ανά ημέρα και ανά ιστοσελίδα, το οποίο επαρκεί σε πολλές περιπτώσεις. Οι Google Maps χρησιμοποιούν το WGS 84 Web Mercator ως το προβολικό τους σύστημα.

2.17.2 Χρήση του Google Maps API

Στη συνέχεια ακολουθεί μια σύντομη περιγραφή των μεθόδων του Google Maps API που χρησιμοποιούνται.

Συμβάντα κατά το σχεδιασμό γεωμετρικών αντικειμένων πάνω στο χάρτη

- `markercomplete`

Στο ακόλουθο τμήμα πηγαίου κώδικα παρουσιάζεται ένας διαχειριστής συμβάντος (event handler) που ενεργοποιείται κατά την επιλογή ενός σημείου πάνω στο χάρτη (*markercomplete*).

```
google.maps.event.addListener(drawingManager, 'markercomplete',  
(marker) => {  
    ...  
});
```

- `polylinecomplete`

Στο ακόλουθο τμήμα πηγαίου κώδικα παρουσιάζεται ένας διαχειριστής συμβάντος που ενεργοποιείται κατά σχεδιασμό ενός `polyline` πάνω στο χάρτη (*polylinecomplete*).

```
google.maps.event.addListener(drawingManager, 'polylinecomplete',  
(polyline) => {  
    ...
```

```
});
```

- `polygoncomplete`

Στο ακόλουθο τμήμα πηγαίου κώδικα παρουσιάζεται ένας διαχειριστής συμβάντος που ενεργοποιείται κατά σχεδιασμό ενός πολυγώνου πάνω στο χάρτη (*polygoncomplete*).

```
google.maps.event.addListener(drawingManager, 'polygoncomplete',  
(polygon) => {  
  ...  
});
```

Μέσα στους παραπάνω διαχειριστές συμβάντων (*{...}*) μπορούν να χρησιμοποιηθούν τα γεωμετρικά αντικείμενα που σχεδιάστηκαν για την εύρεση και τον υπολογισμό ιδιοτήτων και τιμών αντίστοιχα, όπως θα παρουσιαστεί ακολούθως.

Εύρεση των συντεταγμένων ενός σημείου

Στο ακόλουθο τμήμα πηγαίου κώδικα ορίζεται ένας διαχειριστής συμβάντος μέσα στον οποίο εκχωρούνται σε δύο μεταβλητές (*pLat*, *pLng*) οι τιμές του γεωγραφικού πλάτους και μήκους αντίστοιχα, για ένα σημείο που επιλέχθηκε πάνω στο χάρτη (*marker*).

```
google.maps.event.addListener(drawingManager, 'markercomplete',  
(marker) => {  
  let pLat = marker.getPosition().lat();  
  let pLng = marker.getPosition().lng();  
});
```

Υπολογισμός της απόστασης μεταξύ δύο σημείων

Στο ακόλουθο τμήμα πηγαίου κώδικα ορίζεται ένας διαχειριστής συμβάντος μέσα στον οποίο εκχωρούνται σε δύο μεταβλητές (*gmPoint1*, *gmPoint2*) τα σημεία με τις συντεταγμένες τους όπως επιλέχθηκαν πάνω σε ένα Google Maps χάρτη. Στη συνέχεια υπολογίζεται η μεταξύ τους απόσταση σε μέτρα χρησιμοποιώντας τη μέθοδο *computeDistanceBetween()*.

```
google.maps.event.addListener(drawingManager, 'markercomplete',  
(marker) => {  
  ...  
  let gmPoint1 = new google.maps.LatLng(marker1.getPosition().lat(),  
marker1.getPosition().lng());
```

```
    let gmPoint2 = new google.maps.LatLng(marker2.getPosition().lat(),
marker2.getPosition().lng());
    let distance =
google.maps.geometry.spherical.computeDistanceBetween(gmPoint1,
gmPoint2);
}
```

Υπολογισμός του εμβαδού ενός πολυγώνου

Στο ακόλουθο τμήμα πηγαίου κώδικα ορίζεται ένας διαχειριστής συμβάντος μέσα στον οποίο εκχωρείται σε μια μεταβλητή (*areaPlgn*) το εμβαδόν ενός πολυγώνου (*polygon*) που σχεδιάστηκε πάνω σε ένα Google Maps χάρτη. Για τον υπολογισμό του εμβαδού χρησιμοποιείται η μέθοδος *computeArea()*, η οποία επιστρέφει την τιμή του εμβαδού σε τετραγωνικά μέτρα.

```
google.maps.event.addListener(drawingManager, 'polygoncomplete',
(polygon) => {
    let areaPlgn =
google.maps.geometry.spherical.computeArea(polygon.getPath());
});
```

2.18 Turf.js

2.18.1 Γενικά

Η Turf.js είναι μια βιβλιοθήκη της JavaScript για γεωχωρικές αναλύσεις. Είναι ένα project ανοιχτού λογισμικού το οποίο συντηρείται από τη Mapbox. Με τη χρήση της Turf.js είναι εφικτή η εφαρμογή προχωρημένων χωρικών διεργασιών στον φυλλομετρητή. Μεταξύ άλλων, υποστηρίζει πολλές χωρικές διεργασίες, όπως ο υπολογισμός αποστάσεων και εμβαδού. Ακόμα, υποστηρίζει την ενσωμάτωση σημειακών δεδομένων μέσα σε πολύγωνα και τη χωρική παρεμβολή. Τέλος, τα γεωχωρικά δεδομένα στη Turf.js είναι όλα σε μορφή GeoJSON και οι συντεταγμένες θα πρέπει να είναι στη στάνταρ μορφή του WGS84 γεωδαιτικού συστήματος.

2.18.2 Χρήση της βιβλιοθήκης Turf.js

Στη συνέχεια παρουσιάζονται οι μέθοδοι της βιβλιοθήκης Turf.js που χρησιμοποιούνται.

turf.point()

Παίρνει ως όρισμα τις συντεταγμένες ενός σημείου και επιστρέφει ένα σημείο τύπου Point Feature.

Παράδειγμα:

```
let point = turf.point([25.6640625, 37.7185903]);
```

turf.midpoint()

Παίρνει ως ορίσματα δύο σημεία και επιστρέφει ένα σημείο που βρίσκεται στο μέσο της απόστασης των δύο. Κατά τον υπολογισμό λαμβάνει υπόψιν την καμπυλότητα της Γης.

Παράδειγμα:

```
let point1 = turf.point([25.6640625, 37.7185903]);
let point2 = turf.point([145.14244, -37.830937]);

let midpoint = turf.midpoint(point1, point2);
```

turf.buffer()

Παίρνει ως ορίσματα ένα GeoJSON αντικείμενο, την ακτίνα και ένα αντικείμενο που ορίζει 1) τις μονάδες που χρησιμοποιούνται (χιλιόμετρα, μίλια κ.ά) και 2) τα βήματα που λαμβάνονται υπόψιν κατά τον υπολογισμό του buffer zone. Μπορεί να επιστρέφει ένα πολύγωνο (τύπου FeatureCollection | Feature < (Polygon | MultiPolygon)) το οποίο αντιστοιχεί στη buffer zone.

Παράδειγμα:

```
let point = turf.point([25.6640625, 37.7185903]);
let buffered = turf.buffer(point, 500, {units: 'kilometers', steps: 256});
```

turf.polygon()

Παίρνει ως όρισμα μια λίστα με τις συντεταγμένες ενός πολυγώνου και επιστρέφει ένα πολύγωνο τύπου Feature <Polygon>.

Παράδειγμα:

```
let polygon = turf.polygon([[[[5, 52], [4, 56], [2, 51], [7, 54], [5, 52]]], { name: 'plgn0' }]);
```

turf.centroid()

Παίρνει ως όρισμα ένα GeoJSON αντικείμενο και επιστρέφει ένα σημείο που αντιστοιχεί στο κεντροειδές.

Παράδειγμα:

```
let polygon = turf.polygon([[-81, 41], [-88, 36], [-84, 31], [-80, 33], [-77, 39], [-81, 41]]);  
let centroid = turf.centroid(polygon);
```

2.19 Google Charts

Τα Google Charts αναπτύχθηκαν από την ομώνυμη εταιρία και είναι μια υπηρεσία ιστού που σκοπό έχει την παραγωγή γραφημάτων στην πλευρά του πελάτη (client-side). Υποστηρίζουν πολλά είδη γραφημάτων, μεταξύ αυτών είναι: γράφημα στηλών, γράφημα γραμμών, γράφημα πίτας, γράφημα ράβδων, γράφημα φυσαλίδας κ.ά.

2.20 Chart.js

Η Chart.js είναι μια βιβλιοθήκη της JavaScript για τη δημιουργία γραφημάτων στην πλευρά του πελάτη που προσαρμόζουν την εμφάνισή τους ανάλογα με την οθόνη στην οποία βρίσκονται, με στόχο τη βέλτιστη εμπειρία χρήσης. Όπως και τα Google Charts υποστηρίζουν γραφήματα στηλών, γραμμών πίτας, ράβδων, φυσαλίδας κ.ά.

2.21 Linux

Το λειτουργικό σύστημα Linux είναι μια δημοφιλής παραλλαγή του UNIX. Είναι το πιο δημοφιλές λειτουργικό σύστημα για διακομιστές, συγκεκριμένα σύμφωνα με την ιστοσελίδα w3techs.com το Linux έχει ποσοστό 36.8% σε σχέση με το 33.2% των Windows στο μερίδιο αγοράς μεταξύ των λειτουργικών συστημάτων διακομιστών. Επίσης, όλη η οικογένεια των λειτουργικών UNIX έχει ποσοστό χρήσης στο 66.8%.

Το Linux αναπτύχθηκε από τον Linux Torvalds όταν εκείνος αποφάσισε να γράψει ένα κλώνο του UNIX ως φοιτητής. Η 1^η του έκδοση κυκλοφόρησε το 1991. Το Linux διατίθεται δωρεάν υπό την άδεια GPL (GNU Public License).

Ως λειτουργικό σύστημα υποστηρίζει την ταυτόχρονη εκτέλεση πολλών διεργασιών, έχει υψηλό επίπεδο ασφαλείας, κάτι πολύ σημαντικό στους διακομιστές, είναι πολύ αποδοτικό στη διαχείριση των πόρων, χαρακτηρίζεται από μεγάλη σταθερότητα και αξιοπιστία και παρουσιάζει μεγάλη ευελιξία όσον αφορά την εξατομίκευση ανάλογα με τις ανάγκες του χρήστη του.

3 Ανάλυση Απαιτήσεων και Σχεδιασμός της εφαρμογής

3.1 Απαιτήσεις

Λειτουργικές Απαιτήσεις (Functional Requirements)

Ένας ανώνυμος χρήστης να μπορεί να:

- Δημιουργήσει ένα νέο λογαριασμό καταχωρώντας τα στοιχεία που θα του ζητηθούν.
- Συνδεθεί στην εφαρμογή χρησιμοποιώντας τα στοιχεία εισόδου του.

Ένας εγγεγραμμένος χρήστης να μπορεί να:

- Ψηφιοποιήσει τα όρια ενός αγροτεμαχίου και να βρει το συνολικό του εμβαδόν.
- Καταγράψει τα δεδομένα από το GPS της συσκευής του κατά τη διάρκεια μιας επέμβασης ψεκασμού.
- Καταχωρήσει τα στοιχεία μιας επέμβασης ψεκασμού.
- Ανεβάσει στο διακομιστή της εφαρμογής ένα αρχείο με τα δεδομένα κάποιας επέμβασης ψεκασμού σε περίπτωση που στο αγροτεμάχιο του δεν έχει πρόσβαση στο διαδίκτυο.
- Δει το συνολικό αριθμό αγροτεμαχίων και επεμβάσεων του.
- Τη συνολική έκταση που καταλαμβάνουν τα αγροτεμάχιά του.
- Επιλέξει από μια λίστα πρώτα ένα αγροτεμάχιο και έπειτα την επέμβαση ψεκασμού για την οποία θέλει να ενημερωθεί.
- Δει ένα χάρτη με τη διαδρομή που ακολούθησε και τα σημεία έναρξης και λήξης της.
- Επιλέξει ένα σημείο πάνω στη διαδρομή και να δει τις συντεταγμένες του, το υψόμετρο και την ταχύτητα κίνησης.
- Δει ένα γράφημα με τη μεταβολή του υψομέτρου κατά τη διάρκεια της διαδρομής.
- Δει ένα πίνακα με τα ακόλουθα δεδομένα: απόσταση που διένυσε, βέλτιστη ταχύτητα, μέση ταχύτητα, δόση ψεκασμού, παροχή ακροφυσίων, διάρκεια ψεκασμού και όνομα του φυτοφαρμάκου.
- Δει ένα γράφημα με τις ταχύτητες που καταγράφηκαν και τη διαφορά τους σε σχέση με τη βέλτιστη.
- Χρησιμοποιήσει εφτά εργαλεία βασικών χωρικών υπολογισμών: εύρεση συντεταγμένων σημείου, υπολογισμός απόστασης μεταξύ δύο σημείων, εύρεση συντεταγμένων του μέσου δύο σημείων, εμφάνιση buffer zone με βάση μια ακτίνα οριζόμενη από το χρήστη, υπολογισμός μήκους γραμμής, υπολογισμός εμβαδού πολυγώνου και εύρεση του κεντροειδούς ενός πολυγώνου.

Μη Λειτουργικές Απαιτήσεις (Non-functional Requirements)

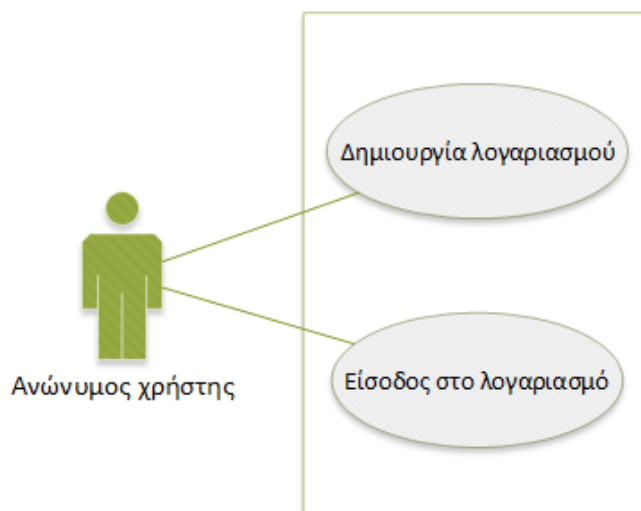
- Η εφαρμογή να μπορεί να τρέξει σε σταθερό/φορητό υπολογιστή, κινητό τηλέφωνο και tablet.
- Το υπόβαθρο των χαρτών να προέρχεται από τα Google Maps.

3.2 Χαρακτήρες (Actors)

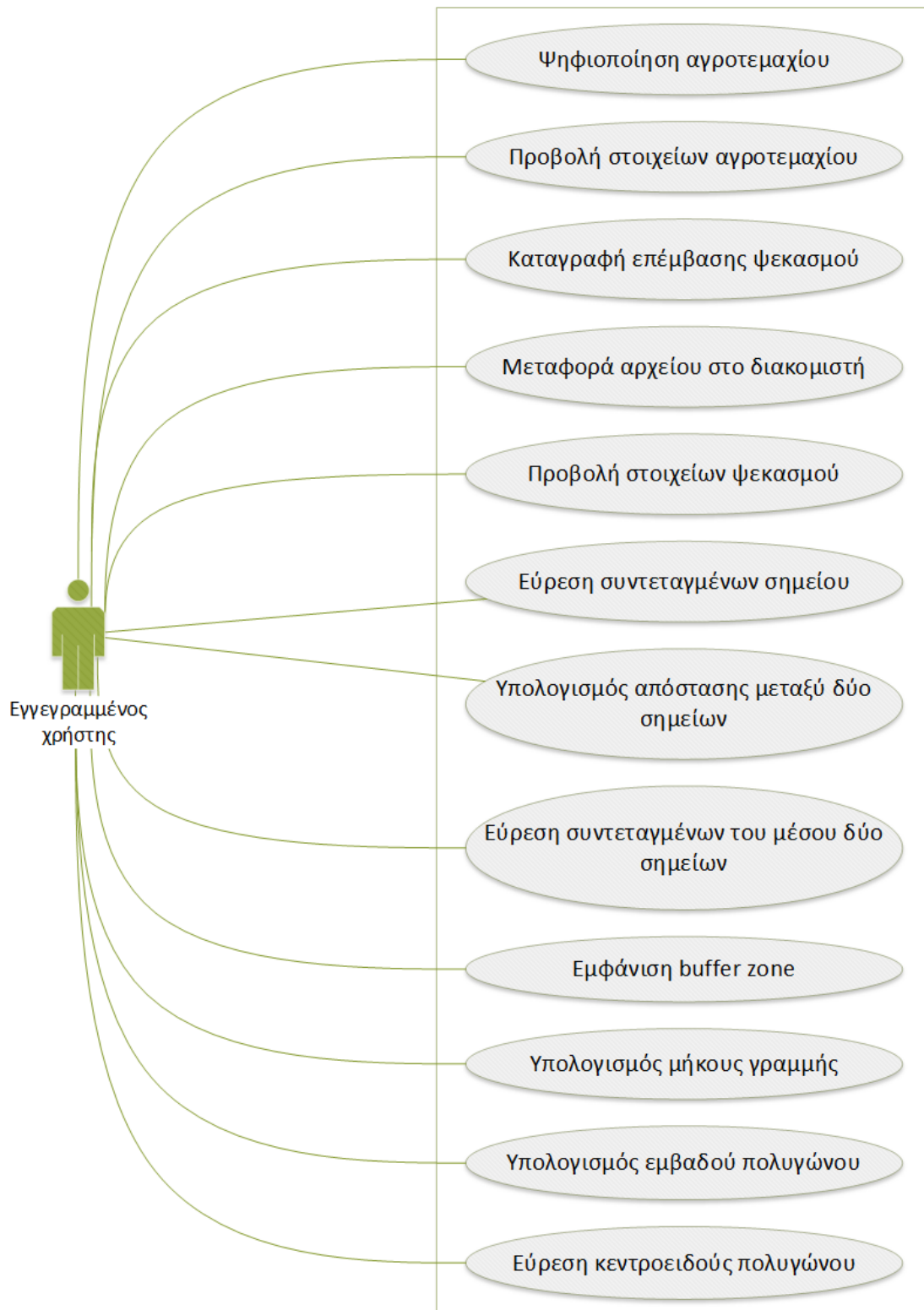
Οι χαρακτήρες οι οποίοι αλληλοεπιδρούν με την εφαρμογή είναι δύο. Από τη μια έχουμε τον χαρακτήρα του ανώνυμου χρήστη και από την άλλη τον χαρακτήρα του εγγεγραμμένου χρήστη. Ο κύριος ρόλος της εφαρμογής είναι αυτός του εγγεγραμμένου χρήστη. Ουσιαστικά πρόκειται για αυτόν που θα επιδρά μέσω ενός γραφικού περιβάλλοντος με την εφαρμογή και θα λαμβάνει πληροφορίες από αυτή. Ο ρόλος του ανώνυμου χρήστη είναι η περίπτωση ενός χρήστη που δεν έχει κάνει την εγγραφή του στην εφαρμογή.

3.3 Περιπτώσεις χρήσης (Use Cases)

Στις εικόνες 3.1 και 3.2 παρουσιάζονται τα διαγράμματα των περιπτώσεων χρήσης για τον ανώνυμο και τον εγγεγραμμένο χρήστη αντίστοιχα.



Εικόνα 3.1: Διάγραμμα περιπτώσεων χρήσης για τον ανώνυμο χρήστη



Εικόνα 3.2: Διάγραμμα περιπτώσεων χρήσης για τον εγγεγραμμένο χρήστη

3.3.1 Δημιουργία λογαριασμού

Σύντομη περιγραφή

Ένας χειριστής μέσω αυτής της περίπτωσης χρήσης μπορεί να δημιουργήσει ένα προσωπικό λογαριασμό με τα στοιχεία του.

Χειριστές

Ανώνυμος χρήστης.

Βασική ροή

- Ο χρήστης επιλέγει το κουμπί “Register”.
- Εμφανίζεται μια φόρμα με τα στοιχεία χρήστη που απαιτούνται.
- Ο χρήστης εισάγει τις απαιτούμενες πληροφορίες και τις στέλνει στο σύστημα για επικύρωση.
- Το σύστημα επικυρώνει τα στοιχεία χρήστη και τον ανακατευθύνει στη φόρμα της εισόδου στο λογαριασμό.

Εναλλακτική ροή 1: Ακύρωση δημιουργίας λογαριασμού

Ο ίδιος ο χρήστης διακόπτει τη δημιουργία του λογαριασμού.

Εναλλακτική ροή 2: Εισαγωγή στοιχείων χρήστη που υπάρχει ήδη

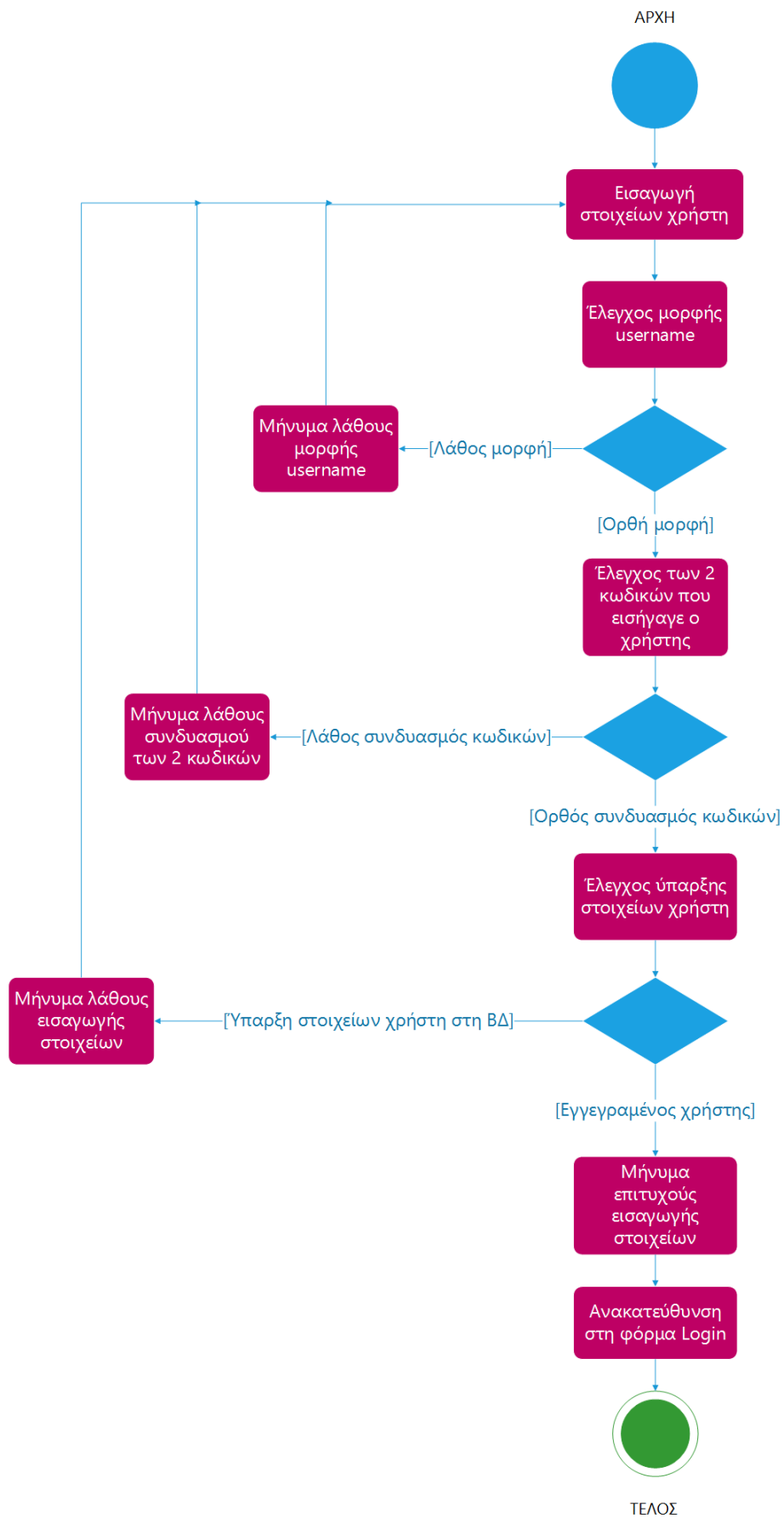
Ο χειριστής εισάγει τα στοιχεία ενός χρήστη που υπάρχει ήδη στο σύστημα. Για λόγους ασφαλείας δεν ενημερώνεται συγκεκριμένα ότι εισήγαγε τα στοιχεία χρήστη που υπάρχει ήδη, αλλά η ειδοποίηση που λαμβάνει είναι ότι εισήγαγε λάθος στοιχεία. Στο σύστημα όμως γίνεται καταγραφή του περιστατικού.

Εναλλακτική ροή 3: Δεν ταιριάζουν μεταξύ τους οι δύο κωδικοί που εισήγαγε

Στη φόρμα εισαγωγής στοιχείων ο κωδικός χρήστη ζητείται να περαστεί δύο φορές, στη συγκεκριμένη ροή οι δύο κωδικοί που δήλωσε δεν ταιριάζουν μεταξύ τους.

Εναλλακτική ροή 4: Λανθασμένη μορφή στοιχείων

Το όνομα χρήστη που ζητείται είναι ένας λογαριασμός e-mail, όταν όμως ο χρήστης εισάγει όνομα χρήστη που δεν ικανοποιεί τα κριτήρια της μορφής του e-mail, τότε ενημερώνετε σχετικά με το σφάλμα.



Εικόνα 3.3: Το διάγραμμα δραστηριότητας για τη δημιουργία λογαριασμού

3.3.2 Είσοδος στο λογαριασμό

Σύντομη περιγραφή

Ο χειριστής μέσω αυτής της περίπτωσης χρήσης μπορεί να εισάγει τα προσωπικά του στοιχεία και να εισέλθει στο σύστημα.

Χειριστές

Ανώνυμος χρήστης.

Βασική ροή

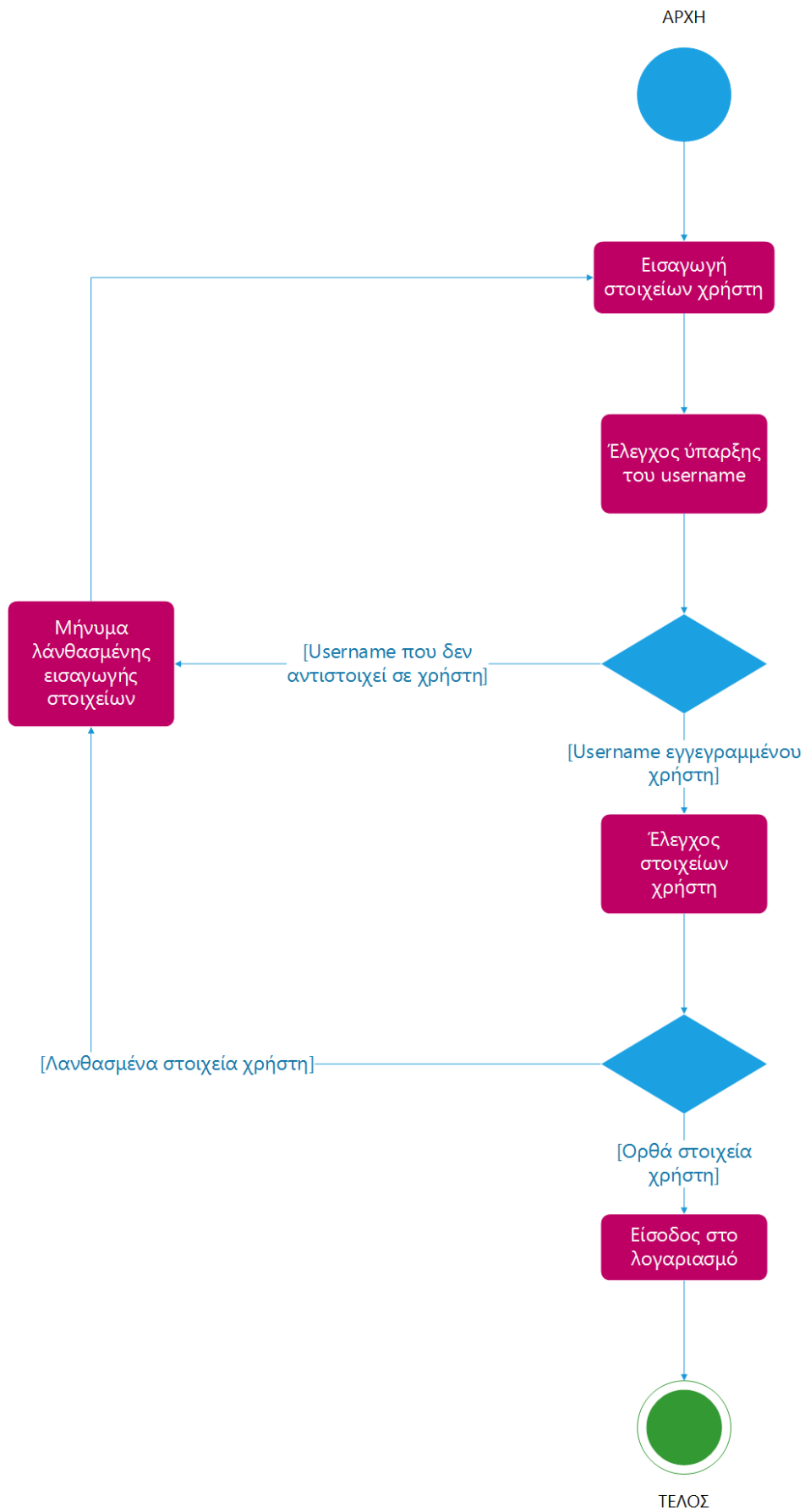
- Ο χρήστης επιλέγει το κουμπί “Log in”.
- Εμφανίζεται μια φόρμα εισαγωγής των στοιχείων χρήστη.
- Ο χρήστης εισάγει τις απαιτούμενες πληροφορίες και τις στέλνει στο σύστημα για επικύρωση.
- Το σύστημα επικυρώνει τα στοιχεία χρήστη και του εμφανίζει την αρχική σελίδα του λογαριασμού.

Εναλλακτική ροή 1: Εισαγωγή στοιχείων χρήστη που δεν υπάρχει στο σύστημα

Ο χειριστής εισήγαγε όνομα χρήστη που δεν υπάρχει στο σύστημα, για αυτό του επιστρέφεται ένα γενικό μήνυμα λάθους εισαγωγής στοιχείων.

Εναλλακτική ροή 2: Εισαγωγή λανθασμένων στοιχείων χρήστη

Ο χειριστής εισήγαγε όνομα χρήστη που υπάρχει στο σύστημα, αλλά με λάθος κωδικό χρήστη. Στο χειριστή επιστρέφει ένα γενικό μήνυμα λάθους εισαγωγής στοιχείων.



Εικόνα 3.4: Το διάγραμμα δραστηριότητας για την είσοδο στο λογαριασμό

3.3.3 Ψηφιοποίηση αγροτεμαχίου

Σύντομη περιγραφή

Ο χειριστής καταχωρεί στη βάση δεδομένων τα όρια του αγροτεμαχίου, το όνομά του και προαιρετικά το είδος των δένδρων που περιέχονται σε αυτό.

Χειριστές

Εγγεγραμμένος χρήστης.

Βασική ροή

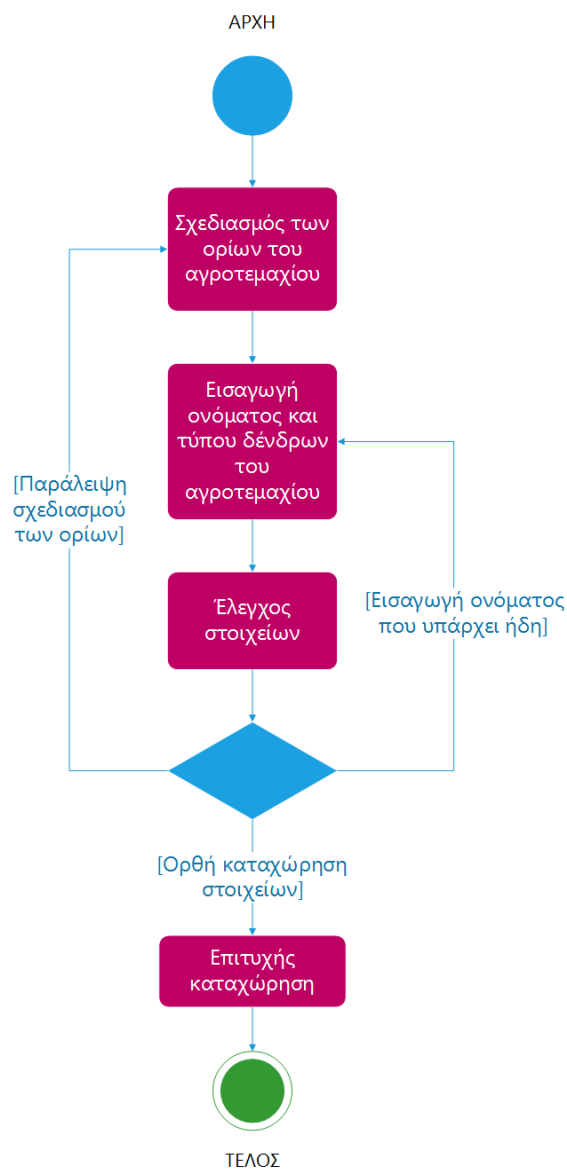
- Ο χρήστης βρίσκει την τοποθεσία του αγροτεμαχίου στο χάρτη.
- Σχεδιάζει τα όρια της έκτασης πάνω στο χάρτη.
- Εισάγει το όνομα του αγροτεμαχίου και προαιρετικά τον τύπο των δένδρων στα αντίστοιχα πεδία της φόρμας.
- Το σύστημα ελέγχει τα στοιχεία και εμφανίζει το μήνυμα της επιτυχούς καταχώρησης.

Ενναλακτική ροή 1: Εισαγωγή ονόματος αγροτεμαχίου που υπάρχει ήδη στο σύστημα

Ο χειριστής εισήγαγε όνομα αγροτεμαχίου που υπάρχει ήδη στο σύστημα, για αυτό του επιστρέφεται το ανάλογο μήνυμα σφάλματος.

Ενναλακτική ροή 2: Παράλειψη σχεδιασμού των ορίων του αγροτεμαχίου

Ο χειριστής παρέλειψε να σχεδιάσει τα όρια του αγροτεμαχίου και για αυτό το σύστημα του επιστρέφει το ανάλογο μήνυμα σφάλματος.



Εικόνα 3.5: Το διάγραμμα δραστηριότητας για τη ψηφιοποίηση ενός αγροτεμαχίου

3.3.4 Προβολή στοιχείων αγροτεμαχίου

Σύντομη περιγραφή

Ο χειριστής επιλέγει ένα αγροτεμάχιο για να προβληθούν τα στοιχεία του που είναι καταχωρημένα στο σύστημα.

Χειριστές

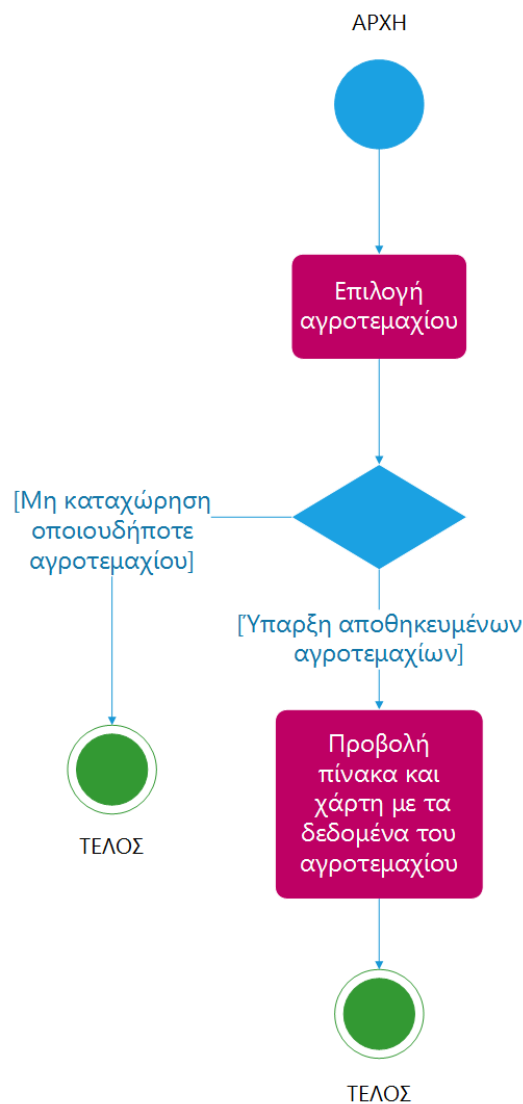
Εγγεγραμμένος χρήστης.

Βασική ροή

- Ο χρήστης επιλέγει ένα αγροτεμάχιο.
- Το σύστημα ελέγχει το αναγνωριστικό του αγροτεμαχίου και επιστρέφει τα δεδομένα που χρειάζονται για την προβολή του.
- Προβάλλεται στη σελίδα ένας πίνακας με το όνομα, το είδος των δένδρων και την έκταση του αγροτεμαχίου. Κάτω από τον πίνακα προβάλλεται ένας χάρτης με σχεδιασμένα τα όρια του αγροτεμαχίου.

Εναλλακτική ροή 1: Δεν έχει καταχωρηθεί κανένα αγροτεμάχιο στο σύστημα

Ο χειριστής δεν μπορεί να επιλέξει κανένα αγροτεμάχιο.



Εικόνα 3.6: Το διάγραμμα δραστηριότητας για την προβολή των στοιχείων ενός αγροτεμαχίου

3.3.5 Καταγραφή επέμβασης ψεκασμού

Σύντομη περιγραφή

Ο χειριστής επιχειρεί να καταγράψει τα δεδομένα μιας επέμβασης ψεκασμού.

Χειριστές

Εγγεγραμμένος χρήστης.

Βασική ροή

- Ο χρήστης καταχωρεί στη φόρμα τα εξής στοιχεία: δόση ψεκασμού, συνολική παροχή ακροφυσίων, απόσταση μεταξύ των δένδρων και όνομα φυτοφαρμάκου.
- Ο χρήστης επιλέγει το αγροτεμάχιο στο οποίο αντιστοιχεί η επέμβαση ψεκασμού.
- Το σύστημα υπολογίζει τη βέλτιστη ταχύτητα κίνησης και προβάλλει τη λειτουργία καταγραφής των χωρικών δεδομένων.
- Ο χειριστής ελέγχει πρώτα την ικανότητα εύρεσης της τοποθεσίας από τη συσκευή του και στη συνέχεια ξεκινάει την καταγραφή. Κατά τη διάρκεια, αν χρειαστεί, μπορεί να σταματήσει προσωρινά την καταγραφή και να τη συνεχίσει αργότερα.
- Ο χειριστής σταματάει την καταγραφή και προχωράει στην αποθήκευση των δεδομένων στη βάση.

Ενναλακτική ροή 1: Λανθασμένη εισαγωγή στοιχείων στη φόρμα

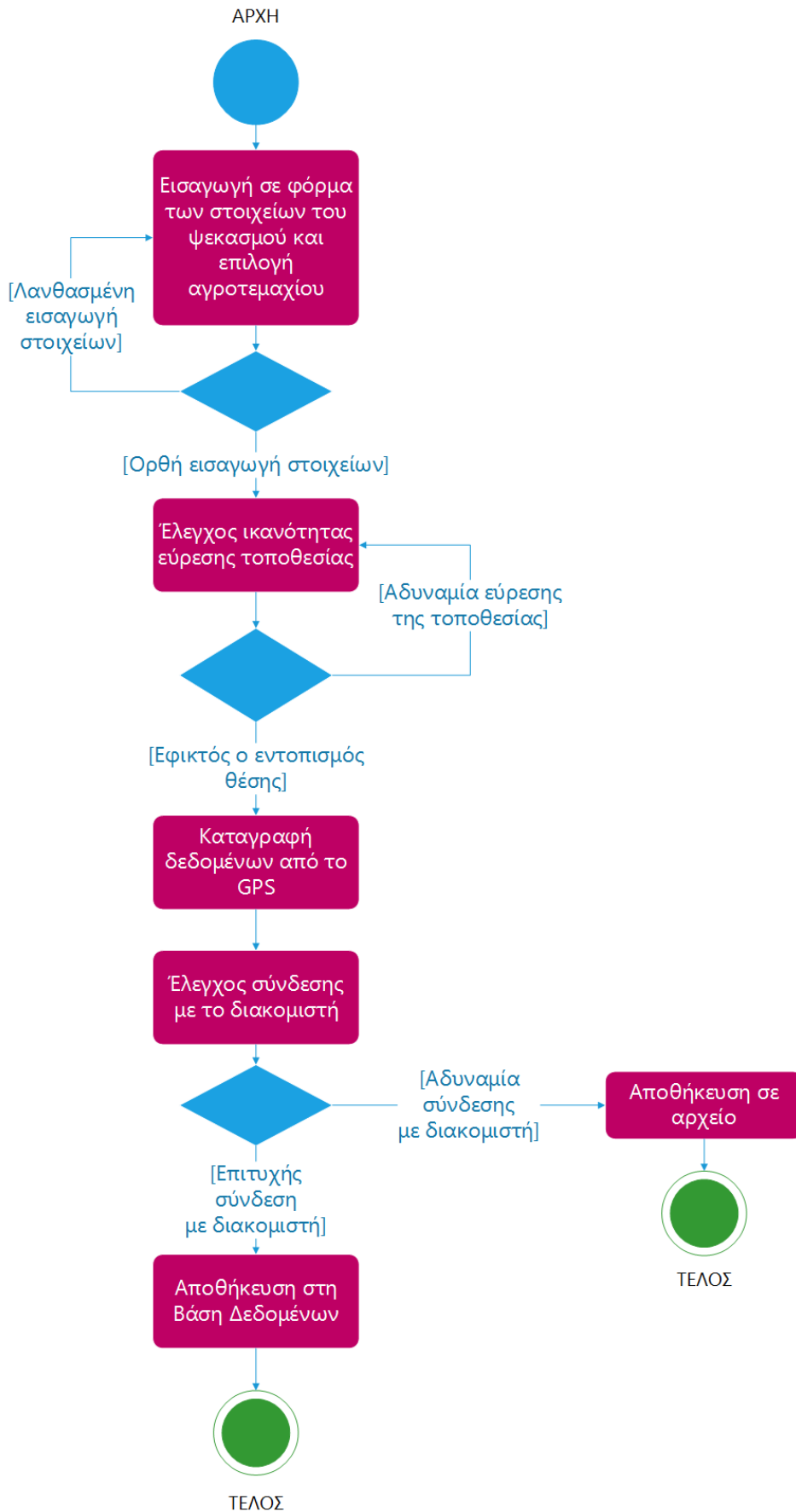
Ο χειριστής θα πρέπει να επαναλάβει την εισαγωγή στοιχείων.

Ενναλακτική ροή 2: Υπάρχει αδυναμία σύνδεσης με το διακομιστή λόγω μη ύπαρξης ενεργούς σύνδεσης στο διαδίκτυο

Ο χειριστής θα πρέπει να αποθηκεύσει τα δεδομένα του ψεκασμού σε ένα αρχείο για να τα φορτώσει όταν θα έχει πρόσβαση στο διαδίκτυο.

Ενναλακτική ροή 3: Υπάρχει αδυναμία εύρεσης της τοποθεσίας της συσκευής

Ο χειριστής δεν μπορεί να προχωρήσει στην καταγραφή των δεδομένων ψεκασμού



Εικόνα 3.7: Το διάγραμμα δραστηριότητας για την καταγραφή μιας επέμβασης ψεκασμού

3.3.6 Μεταφορά αρχείου στο διακομιστή

Σύντομη περιγραφή

Ο χειριστής φορτώνει στο διακομιστή ένα αρχείο με τα δεδομένα της επέμβασης ψεκασμού που έχει καταγράψει.

Χειριστές

Εγγεγραμμένος χρήστης.

Βασική ροή

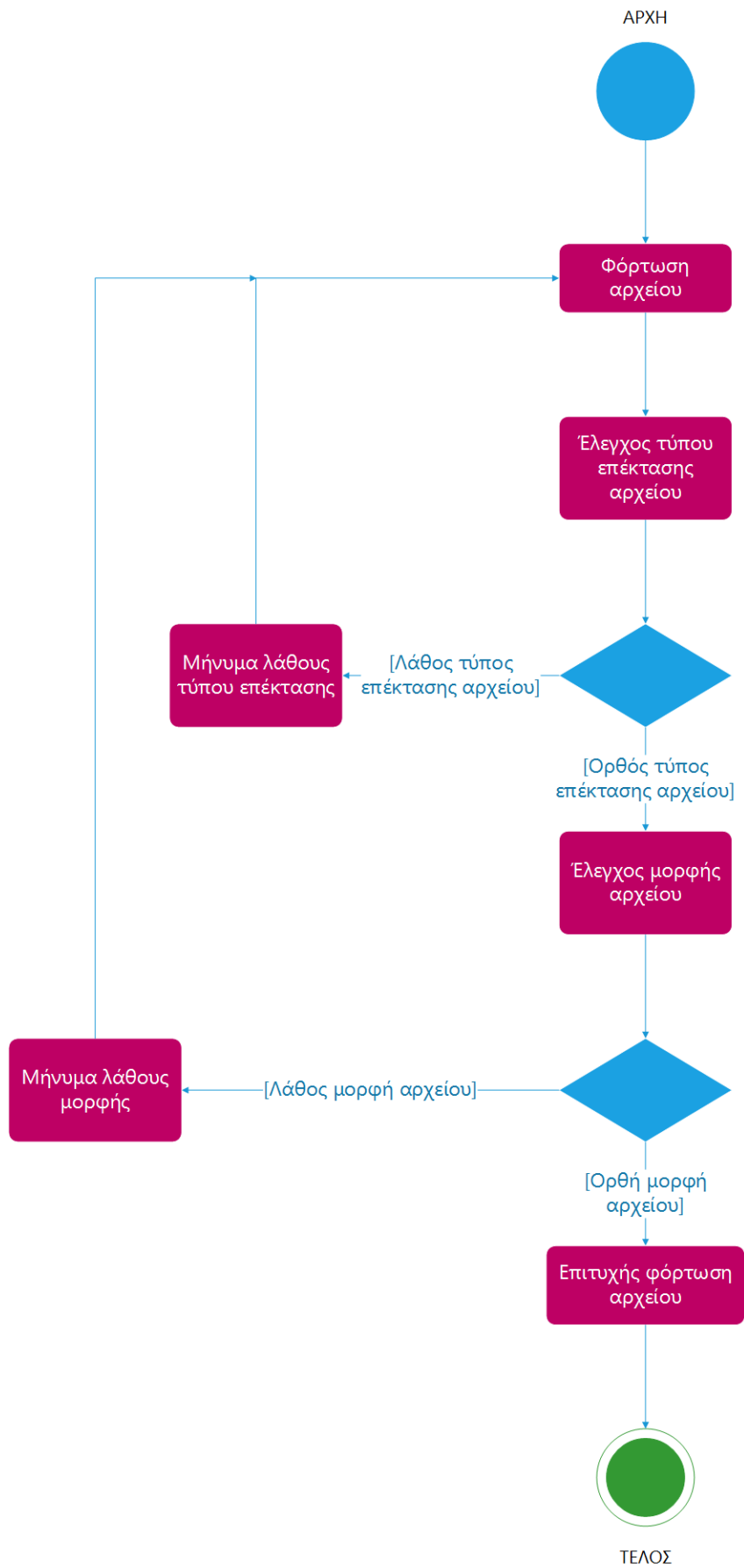
- Ο χρήστης επιλέγει από το μενού την επιλογή “Upload data”.
- Κατευθύνεται στη φόρμα μεταφοράς αρχείου.
- Επιλέγει 1) ένα αρχείο μέσω του παραθύρου διαχείρισης αρχείων του λειτουργικού συστήματος και 2) το αγροτεμάχιο στο οποίο αντιστοιχεί η επέμβαση ψεκασμού και στέλνει τα δεδομένα στο διακομιστή.
- Ο διακομιστής ελέγχει το αρχείο, μεταφέρει τα δεδομένα του μετά από επεξεργασία στη Βάση Δεδομένων και στέλνει μήνυμα επιτυχούς καταχώρησης στο χρήστη.

Ενναλακτική ροή 1: Λάθος επέκταση αρχείου

Το αρχείο που επέλεξε ο χρήστης έχει λάθος τύπο επέκτασης αρχείου. Στο χρήστη επιστρέφεται μήνυμα που τον ενημερώνει για το συγκεκριμένο σφάλμα.

Ενναλακτική ροή 2: Λάθος μορφή αρχείου

Η μορφή που έχει το αρχείο δεν είναι η κατάλληλη έτσι ώστε να διαβαστεί επιτυχημένα από το διακομιστή. Στο χρήστη επιστρέφεται μήνυμα που τον ενημερώνει για το συγκεκριμένο σφάλμα.



Εικόνα 3.8: Το διάγραμμα δραστηριότητας για τη μεταφορά του αρχείου

3.3.7 Προβολή των στοιχείων ψεκασμού

Σύντομη περιγραφή

Ο χειριστής θέλει να προβάλλει τα δεδομένα του ψεκασμού, πιο συγκεκριμένα ένα χάρτη με τη διαδρομή, ένα γράφημα με τη μεταβολή του υψομέτρου κατά τη διάρκεια της διαδρομής και ένα γράφημα με τις ταχύτητες που έχουν καταγραφεί.

Χειριστές

Εγγεγραμμένος χρήστης.

Βασική ροή

- Ο χρήστης επιλέγει πρώτα το αγρόκτημα που επιθυμεί και στη συνέχεια την επέμβαση ψεκασμού με βάση την ημερομηνία και την ώρα έναρξης.
- Ο διακομιστής επιστρέφει στο φυλλομετρητή τα δεδομένα που αντιστοιχούν στη συγκεκριμένη επέμβαση ψεκασμού.
- Μέσω τριών κουμπιών ο χρήστης μπορεί να επιλέξει 1) την προβολή χάρτη, 2) την προβολή των δεδομένων και 3) την προβολή του γραφήματος της ταχύτητας.
 - ◆ Επιλέγοντας την προβολή χάρτη εμφανίζεται ο χάρτης σε μορφή ορθοφωτογραφιών από δορυφόρο. Πάνω στο χάρτη καταγράφεται η διαδρομή που πραγματοποιήθηκε κατά την επέμβαση ψεκασμού, σημειώνονται με τα αντίστοιχα σύμβολα τα σημεία έναρξης και διακοπής. Ο χρήστης μπορεί να επιλέξει ένα σημείο πάνω στη διαδρομή και να δει τις συντεταγμένες, το υψόμετρο και την ταχύτητα διέλευσης.
 - ◆ Επιλέγοντας την προβολή των δεδομένων εμφανίζεται ο πίνακας με τα εξής δεδομένα:
 - Απόσταση που διανύθηκε.
 - Βέλτιστη ταχύτητα.
 - Μέση ταχύτητα.
 - Δόση ψεκασμού.
 - Παροχή ακροφυσίων.
 - Διάρκεια ψεκασμού.
 - Όνομα του φυτοφαρμάκου.
 - ◆ Επιλέγοντας την προβολή του γραφήματος με τις ταχύτητες εμφανίζεται ένα γράφημα σε μορφή μπάρας. Το γράφημα έχει τρεις διαφορετικές μπάρες που εμφανίζουν τον αριθμό των εγγραφών της ιδιότητας ταχύτητα, οι οποίες διακρίνονται στις:
 - Από -10% έως +10% της βέλτιστης.

- Μεγαλύτερη του +10% της βέλτιστης.
- Μικρότερη του -10% της βέλτιστης.

Εναλλακτική ροή 1: Δεν υπάρχει κανένα καταχωρημένο αγρόκτημα στο σύστημα

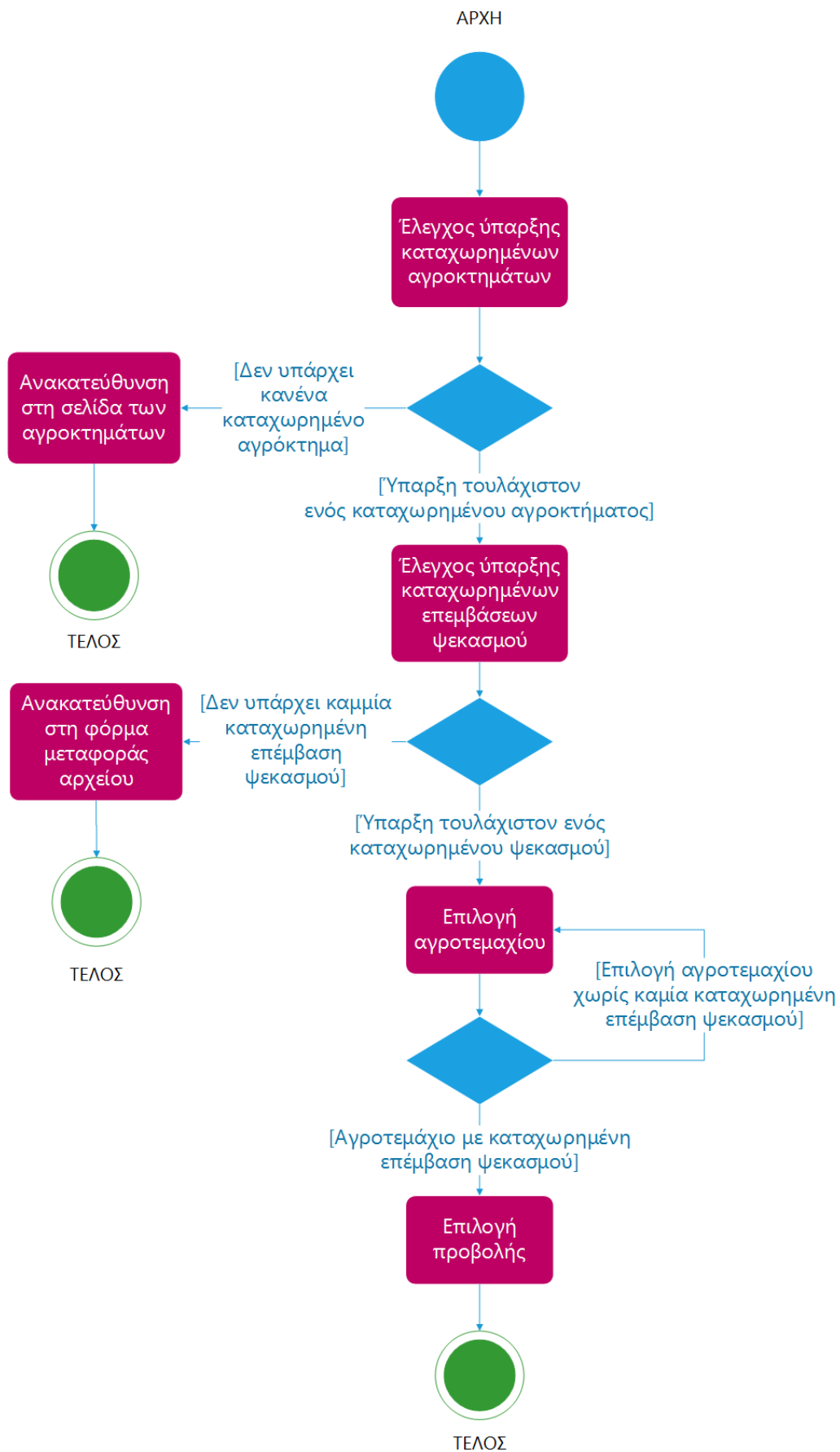
Αν ο χρήστης δεν έχει ψηφιοποιήσει τουλάχιστον ένα αγροτεμάχιο, τότε ανακατευθύνεται στην επιλογή αγροκτήματα με σκοπό να ψηφιοποιήσει τουλάχιστον ένα.

Εναλλακτική ροή 2: Δεν υπάρχει καμία καταχωρημένη επέμβαση ψεκασμού στο σύστημα

Αν ο χρήστης έχει ψηφιοποιήσει τουλάχιστον ένα αγρόκτημα, αλλά δεν έχει καταχωρήσει κάποια επέμβαση ψεκασμού τότε ανακατευθύνεται στη φόρμα μεταφοράς αρχείου.

Εναλλακτική ροή 3: Ο χρήστης επιλέγει ένα αγροτεμάχιο στο οποίο δεν υπάρχει καμία καταχωρημένη επέμβαση ψεκασμού

Ο χρήστης ειδοποιείται μέσω του ανάλογου μηνύματος για τη μη ύπαρξη επέμβασης ψεκασμού για αυτό το αγρόκτημα και μπορεί να επιλέξει κάποιο άλλο αγρόκτημα με καταχωρημένη επέμβαση ψεκασμού.



Εικόνα 3.9: Το διάγραμμα δραστηριότητας για την προβολή των στοιχείων ενός ψεκασμού

3.3.8 Εύρεση συντεταγμένων σημείου

Σύντομη περιγραφή

Ο χειριστής επιλέγοντας ένα σημείο πάνω στο χάρτη βρίσκει τις συντεταγμένες του.

Χειριστές

Εγγεγραμμένος χρήστης.

Βασική ροή

- Ο χρήστης εμφανίζει το χάρτη είτε μέσω της επιλογής των αποθηκευμένων αγροκτημάτων, είτε μέσω της επιλογής της θέσης του εκείνη τη στιγμή.
- Ο χρήστης κάνει αριστερό click σε ένα σημείο και εμφανίζονται οι συντεταγμένες του.



Εικόνα 3.10: Το διάγραμμα δραστηριότητας για την εύρεση των συντεταγμένων ενός σημείου

3.3.9 Υπολογισμός απόστασης μεταξύ δύο σημείων

Σύντομη περιγραφή

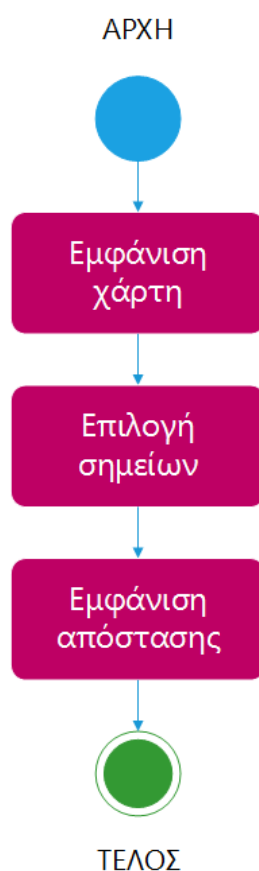
Ο χειριστής επιλέγοντας δύο σημεία πάνω στο χάρτη βρίσκει τη μεταξύ τους απόσταση.

Χειριστές

Εγγεγραμμένος χρήστης.

Βασική ροή

- Ο χρήστης εμφανίζει το χάρτη είτε μέσω της επιλογής των αποθηκευμένων αγροκτημάτων, είτε μέσω της επιλογής της θέσης του εκείνη τη στιγμή.
- Ο χρήστης κάνει αριστερό click σε δύο σημεία πάνω στο χάρτη και εμφανίζεται η μεταξύ τους απόσταση.



Εικόνα 3.11: Το διάγραμμα δραστηριότητας για τον υπολογισμό της απόστασης μεταξύ δύο σημείων

3.3.10 Εύρεση συντεταγμένων του μέσου δύο σημείων

Σύντομη περιγραφή

Ο χειριστής επιλέγοντας δύο σημεία πάνω στο χάρτη βρίσκει τις συντεταγμένες του σημείου που βρίσκεται στο μέσο της απόστασής τους.

Χειριστές

Εγγεγραμμένος χρήστης.

Βασική ροή

- Ο χρήστης εμφανίζει το χάρτη είτε μέσω της επιλογής των αποθηκευμένων αγροκτημάτων, είτε μέσω της επιλογής της θέσης του εκείνη τη στιγμή.
- Ο χρήστης κάνει αριστερό click σε δύο σημεία πάνω στο χάρτη και εμφανίζονται οι συντεταγμένες του σημείου που βρίσκεται στο μέσο της απόστασής τους.



Εικόνα 3.12: Το διάγραμμα δραστηριότητας για την εύρεση των συντεταγμένων του μέσου δύο σημείων

3.3.11 Εμφάνιση buffer zone

Σύντομη περιγραφή

Ο χειριστής επιλέγοντας ένα σημείο πάνω στο χάρτη εμφανίζει μια buffer zone γύρω από αυτό.

Χειριστές

Εγγεγραμμένος χρήστης.

Βασική ροή

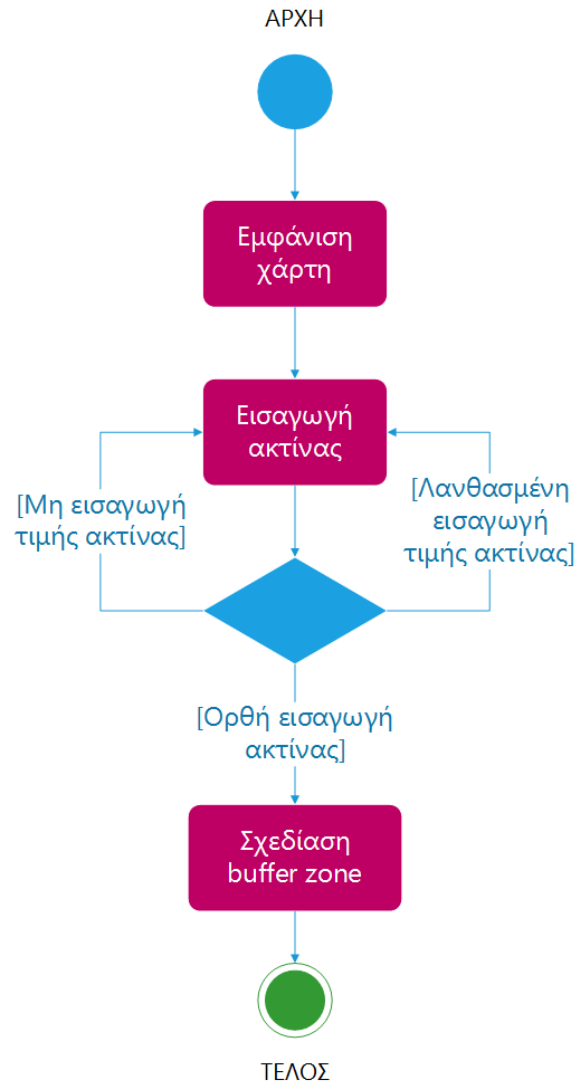
- Ο χρήστης εμφανίζει το χάρτη είτε μέσω της επιλογής των αποθηκευμένων αγροκτημάτων, είτε μέσω της επιλογής της θέσης του εκείνη τη στιγμή.
- Ο χρήστης εισάγει την ακτίνα της buffer zone, κάνει αριστερό click σε ένα σημείο και εμφανίζεται μια buffer zone γύρω από αυτό.

Ενναλακτική ροή 1: Λανθασμένη εισαγωγή της ακτίνας στο πεδίο της φόρμας

Ο χρήστης εισάγει, αντί για ένα αριθμό, γράμματα ή/και σύμβολα, τότε εμφανίζεται το ανάλογο μήνυμα σφάλματος.

Ενναλακτική ροή 2: Μη εισαγωγή της ακτίνας στο πεδίο της φόρμας

Ο χρήστης παραλείπει να εισάγει τιμή ακτίνας και τότε εμφανίζεται το ανάλογο μήνυμα σφάλματος.



Εικόνα 3.13: Το διάγραμμα δραστηριότητας για την εμφάνιση buffer zone γύρω από ένα σημείο

3.3.12 Υπολογισμός μήκους γραμμής

Σύντομη περιγραφή

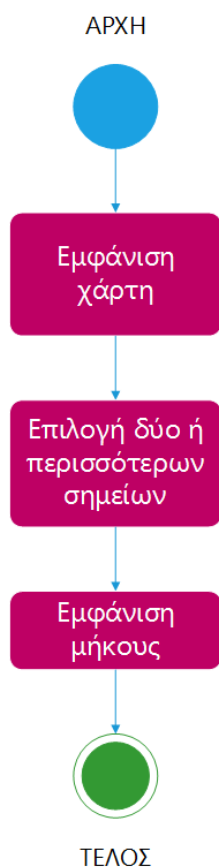
Ο χειριστής επιλέγοντας δύο ή περισσότερα σημεία πάνω στο χάρτη βρίσκει το μήκος της γραμμής που σχηματίζεται.

Χειριστές

Εγγεγραμμένος χρήστης.

Βασική ροή

- Ο χρήστης εμφανίζει το χάρτη είτε μέσω της επιλογής των αποθηκευμένων αγροκτημάτων, είτε μέσω της επιλογής της θέσης του εκείνη τη στιγμή.
- Ο χρήστης κάνει αριστερό click σε δύο ή περισσότερα σημεία πάνω στο χάρτη και εμφανίζεται το μήκος της γραμμής που σχηματίζεται.



Εικόνα 3.14: Το διάγραμμα δραστηριότητας για τον υπολογισμό του μήκους μιας γραμμής

3.3.13 Υπολογισμός εμβαδού πολυγώνου

Σύντομη περιγραφή

Ο χειριστής επιλέγοντας τρία ή περισσότερα σημεία πάνω στο χάρτη εμφανίζει το εμβαδόν του σχήματος.

Χειριστές

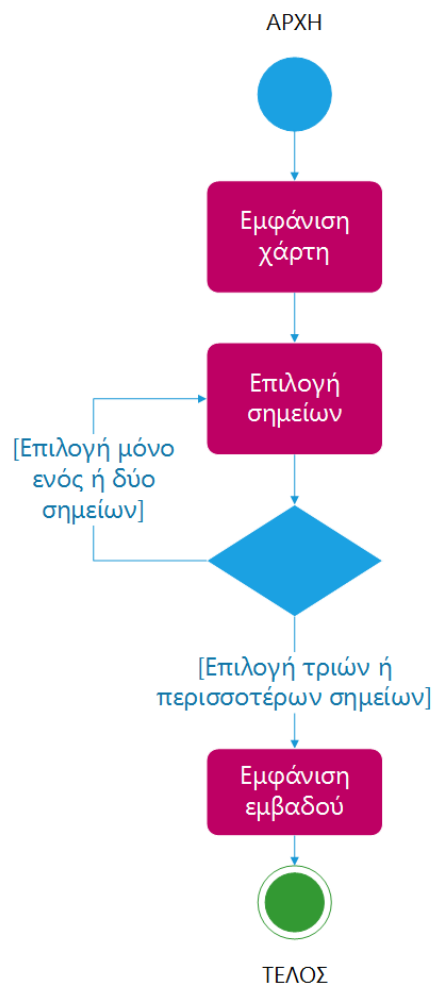
Εγγεγραμμένος χρήστης.

Βασική ροή

- Ο χρήστης εμφανίζει το χάρτη είτε μέσω της επιλογής των αποθηκευμένων αγροκτημάτων, είτε μέσω της επιλογής της θέσης του εκείνη τη στιγμή.
- Ο χρήστης κάνει αριστερό click σε τρία ή περισσότερα σημεία πάνω στο χάρτη και εμφανίζεται το εμβαδόν του σχήματος.

Εναλλακτική ροή 1: Επιλογή ενός ή δύο σημείων πάνω στο χάρτη

Δεν εμφανίζεται το εμβαδόν του σχήματος.



Εικόνα 3.15: Το διάγραμμα δραστηριότητας για τον υπολογισμό του εμβαδού

3.3.14 Εύρεση κεντροειδούς πολυγώνου

Σύντομη περιγραφή

Ο χειριστής επιλέγοντας τρία ή περισσότερα σημεία πάνω στο χάρτη εμφανίζει το κεντροειδές του σχήματος.

Χειριστές

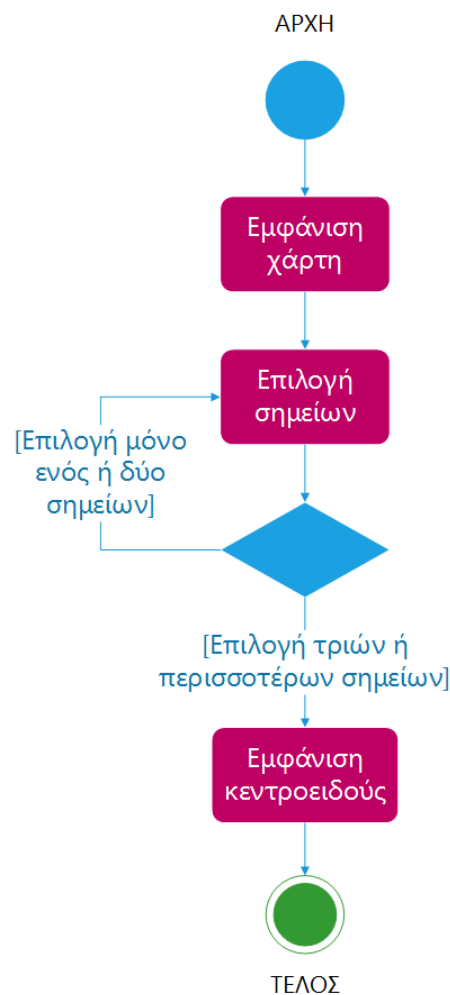
Εγγεγραμμένος χρήστης.

Βασική ροή

- Ο χρήστης εμφανίζει το χάρτη είτε μέσω της επιλογής των αποθηκευμένων αγροκτημάτων, είτε μέσω της επιλογής της θέσης του εκείνη τη στιγμή.
- Ο χρήστης κάνει αριστερό click σε τρία ή περισσότερα σημεία πάνω στο χάρτη και εμφανίζονται οι συντεταγμένες του κεντροειδούς του σχήματος.

Εναλλακτική ροή 1: Επιλογή ενός ή δύο σημείων πάνω στο χάρτη

Δεν εμφανίζεται το κεντροειδές του σχήματος.



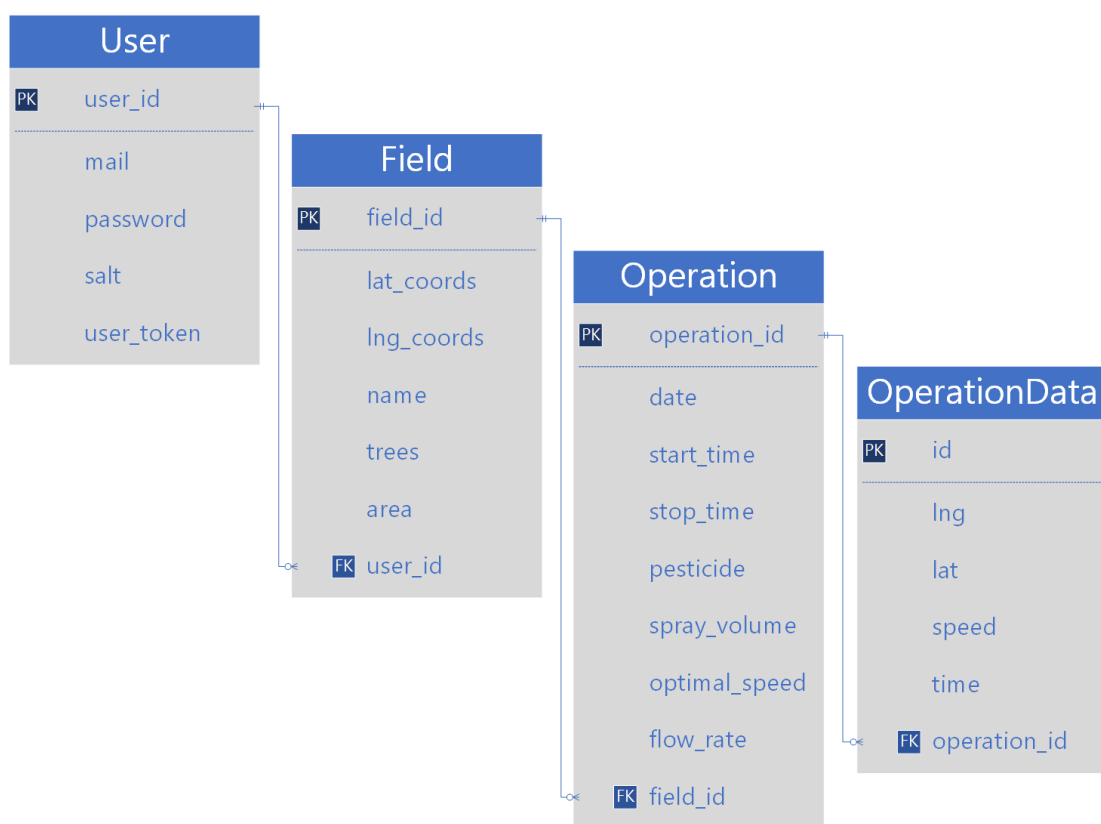
Εικόνα 3.16: Το διάγραμμα δραστηριότητας για την εμφάνιση του κεντροειδούς ενός πολυγώνου

3.4 Σχεδιασμός της Βάσης Δεδομένων

Με βάση τις πληροφορίες που συλλέχθηκαν από τα προηγούμενα στάδια γίνεται ο σχεδιασμός της Βάσης Δεδομένων, όταν τα βήματα γίνονται με αυτή τη σειρά εξασφαλίζεται ότι δεν θα υπάρχουν στη βάση περιττά πεδία.

Το τελικό σχήμα της βάσης αποτελείται από τους παρακάτω πίνακες:

- User
- Field
- Operation
- OperationData



Εικόνα 3.17: Το σχήμα της Βάσης Δεδομένων

3.4.1 Πίνακας User

| Ιδιότητα/Στήλη | Τύπος δεδομένων | Κενό | Κλειδί |
|----------------|-----------------|------|----------|
| user_id | Ακέραιος | Όχι | Πρωτεύον |
| mail | Συμβολοσειρά | Όχι | - |
| password | Συμβολοσειρά | Όχι | - |

| | | | |
|------------|--------------|-----|---|
| salt | Συμβολοσειρά | Όχι | - |
| user_token | Συμβολοσειρά | Όχι | - |

user_id

Πρόκειται για το πρωτεύον κλειδί του πίνακα, για κάθε εγγραφή που εισάγεται η τιμή του αυξάνεται κατά ένα.

mail

Είναι το στοιχείο το οποίο χρησιμοποιείται για όνομα χρήστη στο σύστημα. Επιλέχθηκε γιατί κάθε διεύθυνση e-mail είναι μοναδική και μπορεί να χρησιμοποιηθεί και για επικοινωνία με το χρήστη.

password

Το στοιχείο που χρησιμοποιείται ως κωδικός εισόδου στην εφαρμογή και έχει μήκος 64 χαρακτήρες.

salt

Για λόγους ασφαλείας η συμβολοσειρά του κωδικού ακολουθείται από μια τυχαία συμβολοσειρά μήκους 8 χαρακτήρων. Στη συνέχεια μέσω μιας συνάρτησης κατακερματισμού μετατρέπεται σε δεκαεξαδική (hex) μορφή, στην οποία και αποθηκεύεται ο κωδικός (στήλη password).

user_token

Υπάρχει στη βάση για την παραγωγή ενός μοναδικού token, με σκοπό τη διαχείριση του λογαριασμού.

3.4.2 Πίνακας Field

| Ιδιότητα/Στήλη | Τύπος δεδομένων | Κενό | Κλειδί |
|----------------|-----------------|------|----------|
| field_id | Ακέραιος | Όχι | Πρωτεύον |
| lat_coords | Συμβολοσειρά | Όχι | - |

| | | | |
|------------|--------------|-----|-------------|
| lng_coords | Συμβολοσειρά | Όχι | - |
| name | Συμβολοσειρά | Όχι | - |
| trees | Συμβολοσειρά | Ναι | - |
| area | Ακέραιος | - | - |
| user_id | Ακέραιος | Όχι | Ξένο (User) |

field_id

Πρόκειται για το πρωτεύον κλειδί του πίνακα, για κάθε εγγραφή που εισάγεται η τιμή του αυξάνεται κατά ένα.

lat_coords, lng_coords

Είναι τα δύο στοιχεία στα οποία αποθηκεύονται οι συντεταγμένες των πολυγώνων. Οι συντεταγμένες προστίθενται σε ένα μονοδιάστατο πίνακα και πριν την αποθήκευσή τους μετατρέπονται σε συμβολοσειρές.

name

Το μοναδικό για κάθε χρήστη όνομα ενός αγροτεμαχίου.

trees

Ο τύπος των δένδρων ενός αγροτεμαχίου, η εισαγωγή του οποίου από το χρήστη είναι προαιρετική.

area

Το συνολικό εμβαδόν ενός αγροτεμαχίου σε m².

user_id

Το ξένο κλειδί από τον πίνακα User. Χρησιμοποιείται για την αντιστοίχιση του κάθε αγροτεμαχίου με το χρήστη.

3.4.3 Πίνακας Operation

| Ιδιότητα/Στήλη | Τύπος δεδομένων | Κενό | Κλειδί |
|----------------|----------------------|------|--------------|
| operation_id | Συμβολοσειρά | Όχι | Πρωτεύον |
| date | Ημερομηνία | Όχι | - |
| start_time | Ώρα | Όχι | - |
| stop_time | Ώρα | Όχι | - |
| pesticide | Συμβολοσειρά | Όχι | - |
| spray_volume | Κινητής-υποδιαστολής | Όχι | - |
| optimal_speed | Κινητής-υποδιαστολής | Όχι | - |
| flow_rate | Κινητής-υποδιαστολής | Όχι | - |
| field_id | Ακέραιος | Όχι | Ξένο (Field) |

operation_id

Το μοναδικό αναγνωριστικό για κάθε επέμβαση ψεκασμού, έχει μήκος 30 χαρακτήρων και ένα παράδειγμα της μορφής του: 05e14af46744f137-000011, όπου το 000011 προκύπτει από το user_id.

date

Η ημερομηνία που πραγματοποιήθηκε η έναρξη της επέμβασης ψεκασμού.

start_time

Ο χρόνος έναρξης της επέμβασης ψεκασμού.

stop_time

Ο χρόνος διακοπής της επέμβασης ψεκασμού

pesticide

Το όνομα του φυτοφαρμάκου που χρησιμοποιήθηκε.

spray_volume

Η δόση ψεκασμού σε λίτρα ανά στρέμμα.

optimal_speed

Η βέλτιστη ταχύτητα κίνησης σε χιλιόμετρα ανά ώρα.

flow_rate

Η συνολική παροχή των ακροφυσίων σε λίτρα ανά λεπτό.

field_id

Το ξένο κλειδί από τον πίνακα Field. Χρησιμοποιείται για την αντιστοίχιση της κάθε επέμβασης ψεκασμού με το αγροτεμάχιο στο οποίο ανήκει.

3.4.4 Πίνακας OperationData

| Ιδιότητα/Στήλη | Τύπος δεδομένων | Κενό | Κλειδί |
|----------------|----------------------|------|------------------|
| id | Ακέραιος | Όχι | Πρωτεύον |
| lng | Κινητής-υποδιαστολής | - | - |
| lat | Κινητής-υποδιαστολής | - | - |
| speed | Κινητής-υποδιαστολής | - | - |
| time | Ώρα | - | - |
| operation_id | Συμβολοσειρά | Όχι | Ξένο (Operation) |

id

Το πρωτεύον κλειδί του πίνακα, χρησιμοποιείται για τη μοναδική ταυτοποίηση κάθε εγγραφής με δεδομένα ψεκασμού.

lng

Το γεωγραφικό μήκος (longitude) του σημείου στο σύστημα WGS84.

lat

Το γεωγραφικό πλάτος (latitude) του σημείου στο σύστημα WGS84.

speed

Η ταχύτητα διέλευσης από το σημείο σε χιλιόμετρα ανά ώρα.

time

Η χρονική στιγμή που έγινε η καταγραφή, σε μορφή HH:MM:SS.

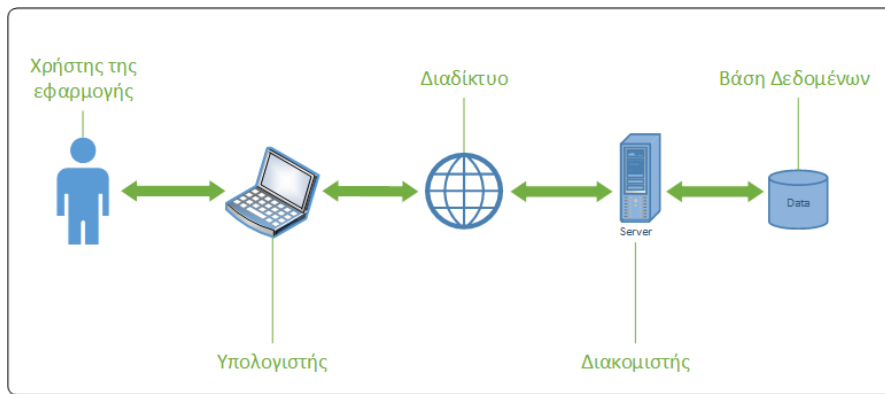
operation_id

Το ξένο κλειδί από τον πίνακα Operation, χρησιμοποιείται για την αντιστοίχιση κάθε εγγραφής δεδομένων του πίνακα OperationData με τα στοιχεία της επέμβασης ψεκασμού στην οποία ανήκει.

3.5 Αρχιτεκτονική και Δομή της εφαρμογής

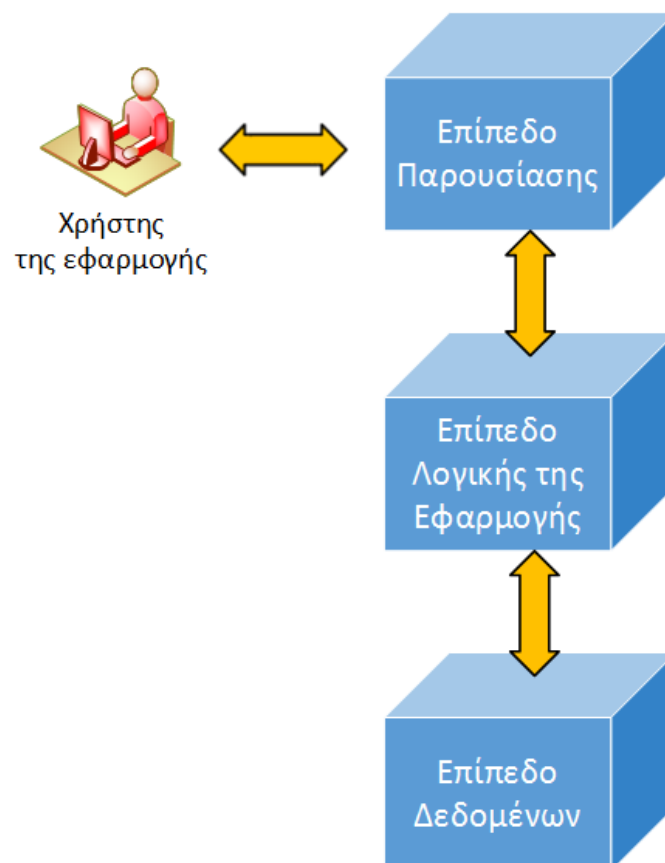
Στην εικόνα 3.18 παρουσιάζονται σε γενικό πλαίσιο τα μέρη του συστήματος. Η εφαρμογή αποτελείται κατά βάση από το τμήμα του διακομιστή, στο οποίο βρίσκεται ένα σημαντικό μέρος της λογικής του συστήματος. Τα δεδομένα πάνω στα οποία κάνει την επεξεργασία ο διακομιστής αντλούνται από μια Βάση Δεδομένων, η οποία μπορεί να βρίσκεται στο ίδιο μηχάνημα και χώρο με το διακομιστή, αλλά μπορεί και να βρίσκεται απομακρυσμένη σε ένα άλλο μηχάνημα.

Στη συνέχεια, μέσω του Διαδικτύου, μεταφέρονται στον υπολογιστή του χρήστη τα αποτελέσματα της επεξεργασίας των δεδομένων. Από εκεί μέσω του φυλλομετρητή της επιλογής του ο χρήστης μπορεί να τα δει και να αλληλοεπιδράσει με το τμήμα της λογικής της εφαρμογής που τρέχει στην πλευρά του πελάτη (client-side).



Εικόνα 3.18: Μια γενική εικόνα του συστήματος

Στην εικόνα 3.19 παρουσιάζεται η αρχιτεκτονική του συστήματος. Το σύστημα χωρίζεται σε τρία νοητά επίπεδα. Ο χρήστης αλληλοεπιδρά με το σύστημα μέσω του Επιπέδου της Παρουσίασης. Η λειτουργικότητα της εφαρμογής αναπτύσσεται στο Επίπεδο Λογικής και τα δεδομένα που χρησιμοποιούνται από την εφαρμογή βρίσκονται στο δικό τους ξεχωριστό επίπεδο, το Επίπεδο Δεδομένων.

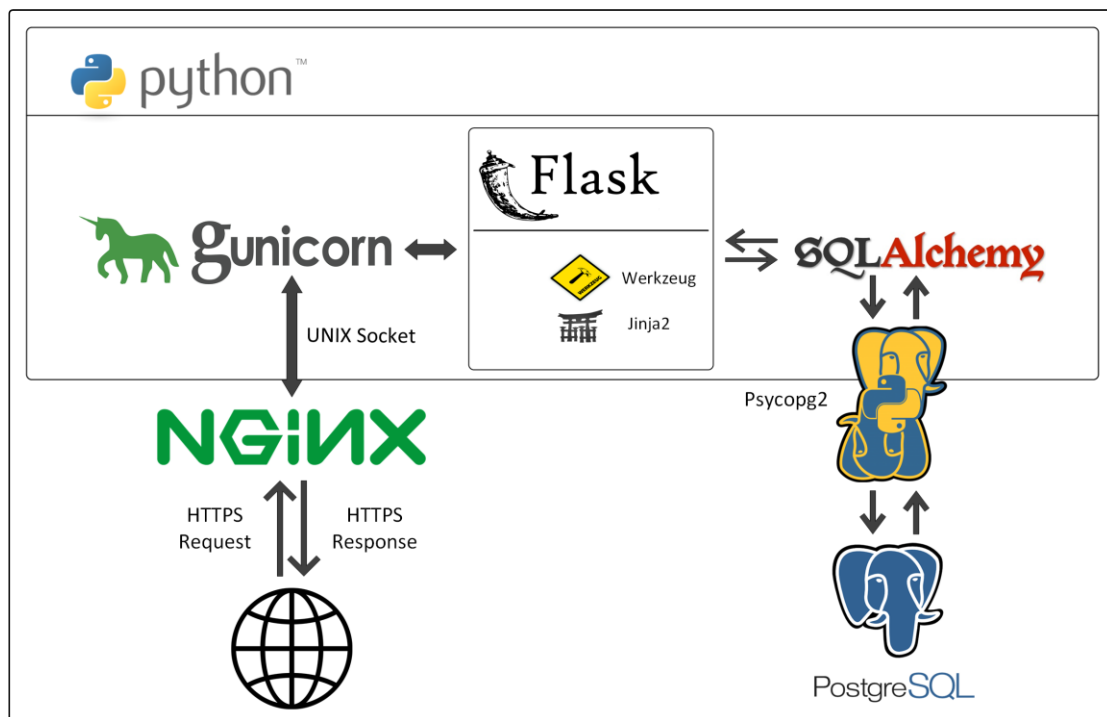


Εικόνα 3.19: Η αρχιτεκτονική της εφαρμογής

Στην εικόνα 3.20 παρουσιάζονται οι τεχνολογίες που χρησιμοποιούνται στην πλευρά του διακομιστή. Αρχικά, τα HTTP requests και responses τα χειρίζεται ο Nginx ο

οποίος λειτουργεί ως reverse proxy και κατευθύνει την κίνηση στο Gunicorn μέσω ενός Unix socket. Ουσιαστικά ο Nginx λειτουργεί ως ο διακομιστής ιστού (web server) που κατευθύνει την κίνηση στο διακομιστή εφαρμογής (application server). Η αρχιτεκτονική την οποία ακολουθεί η εφαρμογή είναι αυτή των τριών επιπέδων (three tier architecture) όπου στο 1^ο επίπεδο έχουμε το διακομιστή ιστού, στο 2^ο το διακομιστή εφαρμογής και στο 3^ο τη βάση δεδομένων. Στην αρχιτεκτονική αυτή ιστορικά το κάθε ένα επίπεδο βρίσκεται σε ξεχωριστό μηχάνημα, αλλά οι ανάγκες της παρούσας εφαρμογής δεν είναι τέτοιες που να απαιτείται η λειτουργία αυτών των επιπέδων σε τρία διαφορετικά μηχανήματα. Επομένως και τα τρία αυτά επίπεδα της εφαρμογής βρίσκονται στο ίδιο μηχάνημα.

Η γλώσσα προγραμματισμού που χρησιμοποιείται είναι η Python, της οποίας η πλούσια κοινή βιβλιοθήκη (standard library) κατά τη διάρκεια της ανάπτυξης της εφαρμογής διευκόλυνε σε αρκετές περιπτώσεις την γρήγορη επίτευξη λειτουργικότητας. Σε αυτό τον τομέα συνέβαλλε και το χαρακτηριστικό της ως γλώσσα, να επιτυγχάνει δηλαδή κανείς μέσα σε λίγες γραμμές κώδικα υψηλή λειτουργικότητα.



Εικόνα 3.20: Η δομή της εφαρμογής στην πλευρά του διακομιστή

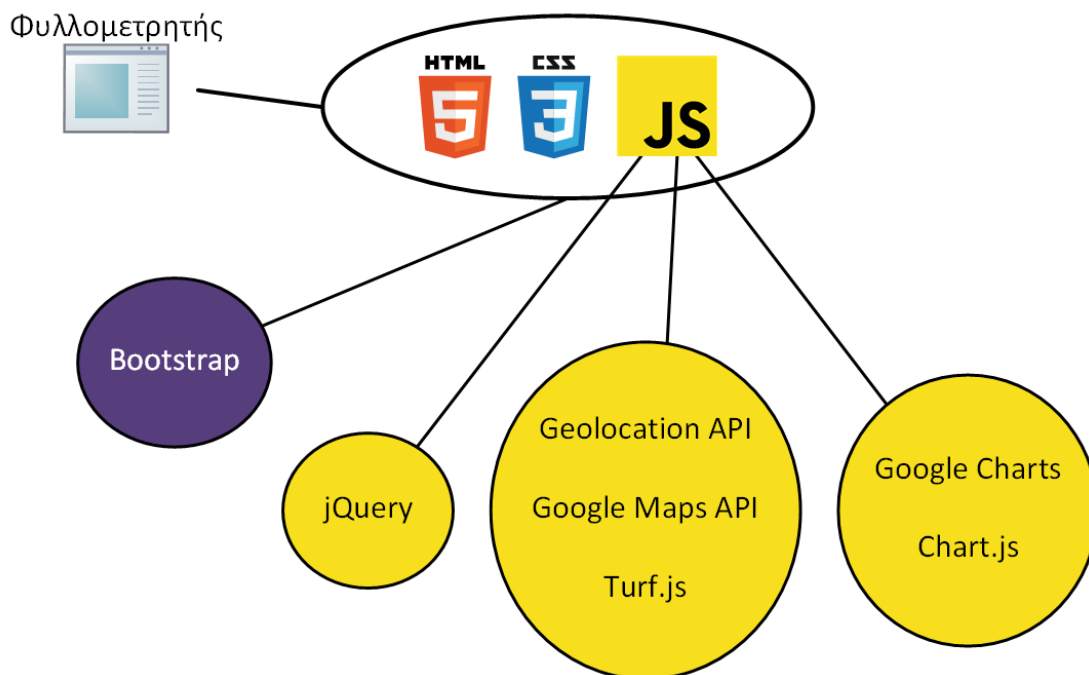
Για τον τομέα της διαχείρισης της εφαρμογής ιστού (web application) χρησιμοποιήθηκε το Flask Web Framework, το οποίο στο μεγαλύτερο βαθμό αποτελείται από δύο υποσυστήματα, το Werkzeug και το Jinja2. Για τη διαχείριση της Βάσης Δεδομένων χρησιμοποιήθηκε η Object Relational Mapping (ORM) εργαλειοθήκη SQLAlchemy, χάρη στην οποία γίνεται πιο εύκολη η αντιστοίχιση μεταξύ δεδομένων της βάσης και των αντικειμένων της Python. Ακόμα, μέσω της

ενδιάμεσης αφαίρεσης που προσφέρει ένα ORM εργαλείο είναι δυνατή η λειτουργία της εφαρμογής και σε διαφορετικά Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ). Δηλαδή, η εφαρμογή μπορεί να δουλέψει χωρίς τροποποιήσεις στον πηγαίο κώδικα, εκτός του PostgreSQL που χρησιμοποιείται στην εφαρμογή και με άλλα ευρέως χρησιμοποιούμενα ΣΔΒΔ, όπως τα MySQL, Microsoft SQL Server και SQLite. Η μοναδική αλλαγή που χρειάζεται είναι στα στοιχεία σύνδεσης του χρήστη στο ΣΔΒΔ και η εγκατάσταση του κατάλληλου προσαρμογέα του ΣΔΒΔ. Η επικοινωνία με τη PostgreSQL επιτυγχάνεται με τον προσαρμογέα για Python-PostgreSQL, Psycopg2. Ο Psycopg2 είναι πλήρως συμβατός με την προδιαγραφή Python DB API 2.0 και επιτυγχάνει ασφαλή διαχείριση των νημάτων.

Στην εικόνα 3.21 παρουσιάζονται οι τεχνολογίες που χρησιμοποιούνται στην πλευρά του πελάτη (client-side). Εκεί αρχικά χρησιμοποιείται η HTML5 για το περιεχόμενο, η CSS3 για την παρουσίαση του περιεχομένου και η JavaScript για τη λογική της εφαρμογής στο επίπεδο του φυλλομετρητή και για την αποστολή και διαχείριση των HTTP Requests μέσω της AJAX μεθόδου.

Για την καλύτερη λειτουργία του Responsive Web Design (RWD) και γενικότερα για τη βελτίωση της παρουσίασης του περιεχομένου στην πλευρά του πελάτη χρησιμοποιήθηκε το Bootstrap 3.

Στον τομέα της JavaScript χρησιμοποιήθηκε η βιβλιοθήκη jQuery για την απλούστευση του πηγαίου κώδικα. Για τον εντοπισμό της τοποθεσίας χρησιμοποιείται το Geolocation API. Στον τομέα της διαχείρισης των χωρικών δεδομένων χρησιμοποιείται το Google Maps API για την εμφάνιση των χαρτών και η βιβλιοθήκη Turf.js για τους χωρικούς υπολογισμούς. Τα γραφήματα χρησιμοποιούν τις βιβλιοθήκες Google Charts και Chart.js.



Εικόνα 3.21: Οι τεχνολογίες που χρησιμοποιούνται στην πλευρά του πελάτη (client-side)

Στην εικόνα 3.22 παρουσιάζεται η δομή της εφαρμογής στην πλευρά του διακομιστή (server-side). Η λογική της εφαρμογής οργανώνεται σε 6 αρθρώματα (modules), από τα οποία τα 5 είναι τα βασικά και το ένα (config.py) χρησιμοποιείται για τις ρυθμίσεις του Flask Web Framework και τη σύνδεση στη PostgreSQL μέσω του προσαρμογέα Psycopg2.

Το άρθρωμα wsgi.py είναι αυτό από το οποίο ξεκινάει η εκτέλεση της εφαρμογής. Πιο συγκεκριμένα το Gunicorn έχει την εντολή να εκτελεί το wsgi.py για να ξεκινήσει η εφαρμογή, αυτή είναι και η μοναδική του λειτουργία.

Το άρθρωμα routes_views.py είναι υπεύθυνο για τις δρομολογήσεις των URLs και την επιστροφή ενός HTTP response στον πελάτη. Το response μπορεί να έχει τη μορφή ενός HTML template ή ενός JSON. Για την εύρεση των κατάλληλων δεδομένων καλεί μεθόδους που βρίσκονται στο άρθρωμα handler.py.

Το handler.py λειτουργεί ως ένας ενδιάμεσος μεταξύ του routes_views.py και του db_manipulation.py. Ο ρόλος του είναι μέσω των μεθόδων του να αντλεί δεδομένα καλώντας τις κατάλληλες μεθόδους από το db_manipulation.py και να τα επεξεργάζεται κατάλληλα, έτσι ώστε να τα επιστρέφει στο routes_views.py στη μορφή που απαιτείται.

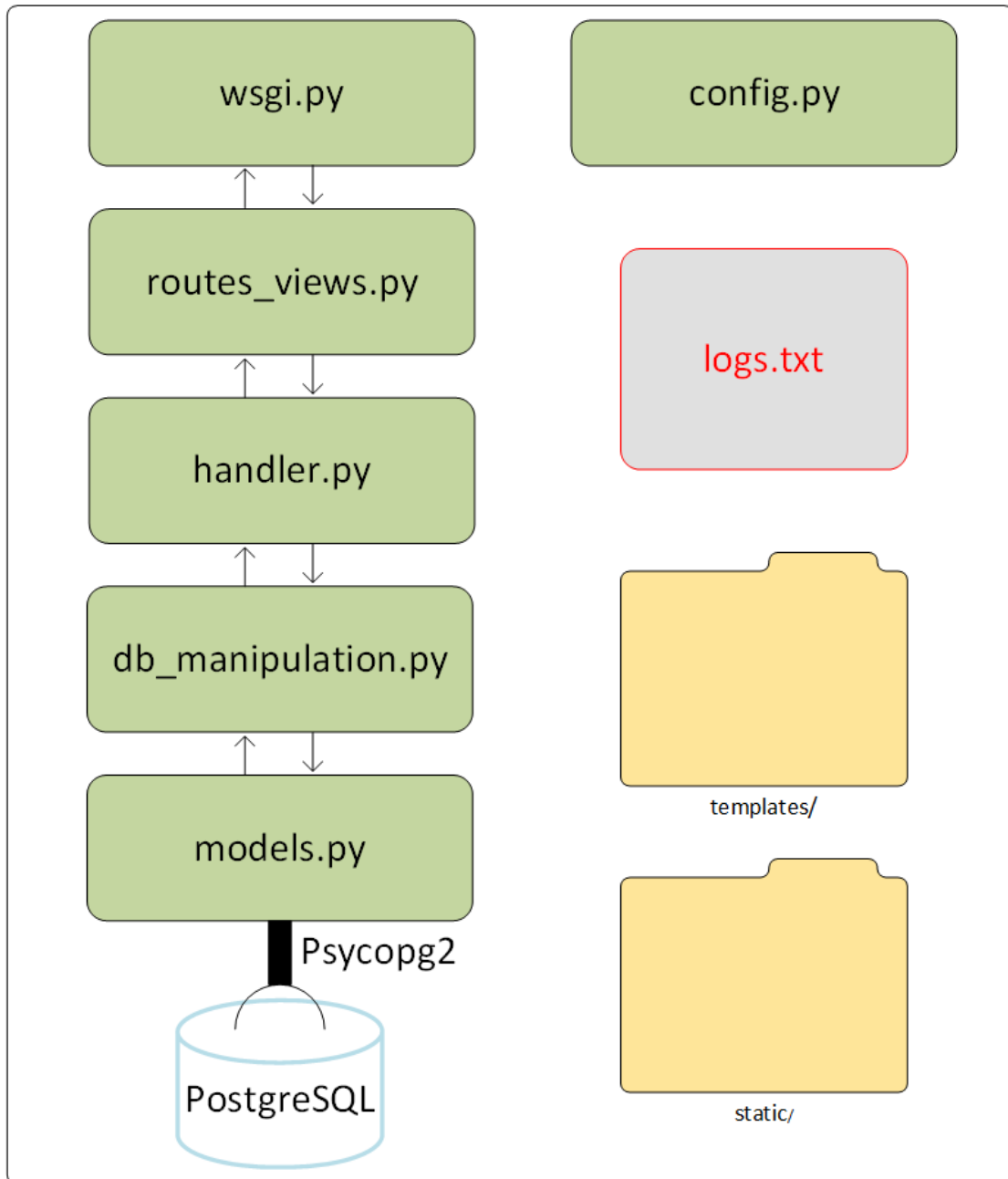
Το db_manipulation.py διαχειρίζεται τα δεδομένα της βάσης δεδομένων. Το άρθρωμα αυτό περιέχει μεθόδους για 1) την είσοδο ελεγμένων και επεξεργασμένων δεδομένων στη βάση, 2) τον έλεγχο της ύπαρξης κάποιων συγκεκριμένων δεδομένων και 3) την πραγματοποίηση επερωτήσεων προς τη βάση για την άντληση δεδομένων.

Το models.py περιέχει 4 κλάσεις οι οποίες αντιστοιχούν η κάθε μια στις σχέσεις της βάσης. Δηλαδή, αποτελείται από τις κλάσεις User, Field, Operation και OperationData. Η ύπαρξη αυτής της δομής είναι απαραίτητη για τη χρησιμοποίηση του SQLAlchemy. Με αυτό τον τρόπο κάθε ιδιότητα της βάσης αντιστοιχεί σε μια μεταβλητή στιγμιότυπου της Python.

Για τη σύνδεση της εφαρμογής με τη PostgreSQL χρησιμοποιείται ο προσαρμογέας Psycopg2.

Για την καταγραφή πιθανών σφαλμάτων υπάρχει το αρχείο logs.txt στο οποίο καταγράφεται η ημερομηνία, η ώρα και ο τύπος σφάλματος.

Η εφαρμογή αποτελείται επίσης και από 2 καταλόγους με αρχεία. Στον κατάλογο templates υπάρχουν αρχεία html τα οποία χρησιμοποιούνται από τη μηχανή προτύπων Jinja2. Ο κατάλογος static υποδιαιρείται στους καταλόγους img, scripts και styles. Στον κατάλογο img υπάρχουν τα εικονίδια της εφαρμογής. Στον κατάλογο scripts υπάρχουν τα αρχεία της JavaScript. Τέλος, στον κατάλογο styles υπάρχει το αρχείο με τους κανόνες της CSS.



Εικόνα 3.22: Η δομή του πηγαίου κώδικα στην πλευρά του διακομιστή (server-side)

4 Η εφαρμογή

4.1 Εγκατάσταση και ρύθμιση

Η εφαρμογή για να είναι προσβάσιμη μέσω του διαδικτύου ανέβηκε σε ένα Virtual Private Server (VPS) με τα εξής χαρακτηριστικά:

| | |
|----------------------------|------------------------------|
| Virtualization | OpenVZ |
| Λειτουργικό Σύστημα | Debian 8.0 x86_64 ("jessie") |
| Bandwidth | 500GB |
| Μνήμη | 1GB |
| VSwap μνήμη | 512MB |
| Χωρητικότητα δίσκου | 10GB |

Το domain name της εφαρμογής: agrisopt.top

Τα προγράμματα και πακέτα που εγκαταστάθηκαν στο διακομιστή (server) για να λειτουργήσει η εφαρμογή είναι:

| Προγράμματα | Πακέτα της Python |
|--------------------|--------------------------|
| Python 3.5.2 | Virtualenv 15.1.0 |
| Pip 9.0.1 | Gunicorn 19.7.1 |
| Nginx 1.6.2 | Flask 0.11.1 |
| Python-Psycopg2 | Flask-Login 0.4.0 |
| PostgreSQL 9.4.15 | Psycopg2 2.7.3.2 |
| | Flask-SQLAlchemy 2.3.2 |

Μετά την εγκατάσταση ακολούθησαν οι ρυθμίσεις για να λειτουργήσει η εφαρμογή. Αρχικά, δημιουργήθηκε ένας νέος λογαριασμός χρήστη στη PostgreSQL στον οποίο και εκχωρήθηκε η βάση δεδομένων της εφαρμογής. Στη συνέχεια δημιουργήθηκε ένα service στο σύστημα το οποίο έχει ως ρόλο να εκτελεί τον κώδικα στην πλευρά του εξυπηρετητή. Αυτόν τον ρόλο έχει το Gunicorn, το οποίο ρυθμίστηκε να τρέχει με 2 workers και να διοχετεύεται σε αυτό η κίνηση από τον Nginx διακομιστή μέσω ενός UNIX socket. Μετέπειτα ρυθμίστηκε ο Nginx λειτουργεί ως reverse proxy και να κατευθύνει την κίνηση στο Unix socket. Τέλος, έγιναν οι απαραίτητες ρυθμίσεις για τη χρήση της HTTPS (HTTP Secure) σύνδεσης αντί της απλής HTTP.

4.2 Περιγραφή

Στο κεφάλαιο αυτό παρουσιάζεται η εφαρμογή όπως είναι κατά τη λειτουργία της, για την παρουσίαση χρησιμοποιούνται εικόνες (screenshots) από την εφαρμογή και περιγράφονται οι αντίστοιχες ενέργειες που κάνει ο χρήστης για να αλληλοεπιδράσει με το σύστημα.

Το logo της εφαρμογής



Εικόνα 4.1: Το logo της εφαρμογής

Επιλέχθηκε για όνομα της εφαρμογής το “Agrisopt” που προκύπτει από το **Agricultural spraying optimization** και για εικονίδιο μια αναπαράσταση ενός γεωργικού ελκυστήρα με τα εικονίδια της τοποθεσίας και ενός δορυφόρου.

Η αρχική σελίδα

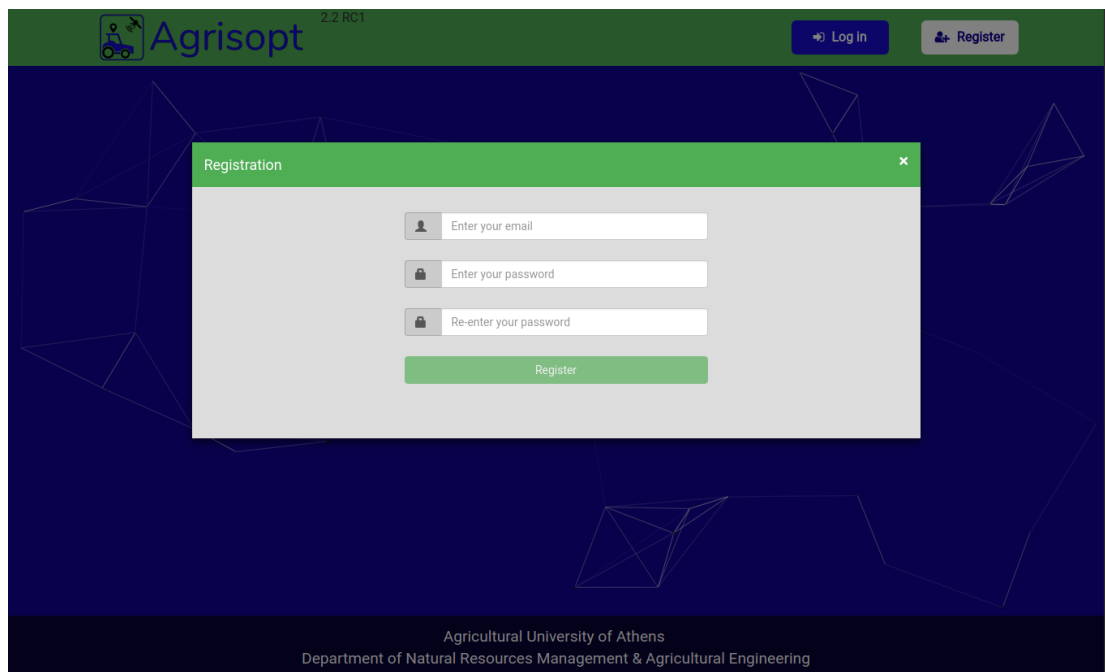


Εικόνα 4.2: Η αρχική σελίδα

Η αρχική σελίδα αποτελείται από τη μπάρα με το logo της εφαρμογής, το κουμπί για την επιλογή Register και το κουμπί για την επιλογή Log in. Στο κέντρο της οθόνης παρουσιάζεται ένα μήνυμα που περιγράφει με τρεις λέξεις τη βασική λειτουργία της

εφαρμογής. Στο κατώτερο σημείο υπάρχει ένα υποσέλιδο με τα ονόματα του πανεπιστημίου και του τμήματος.

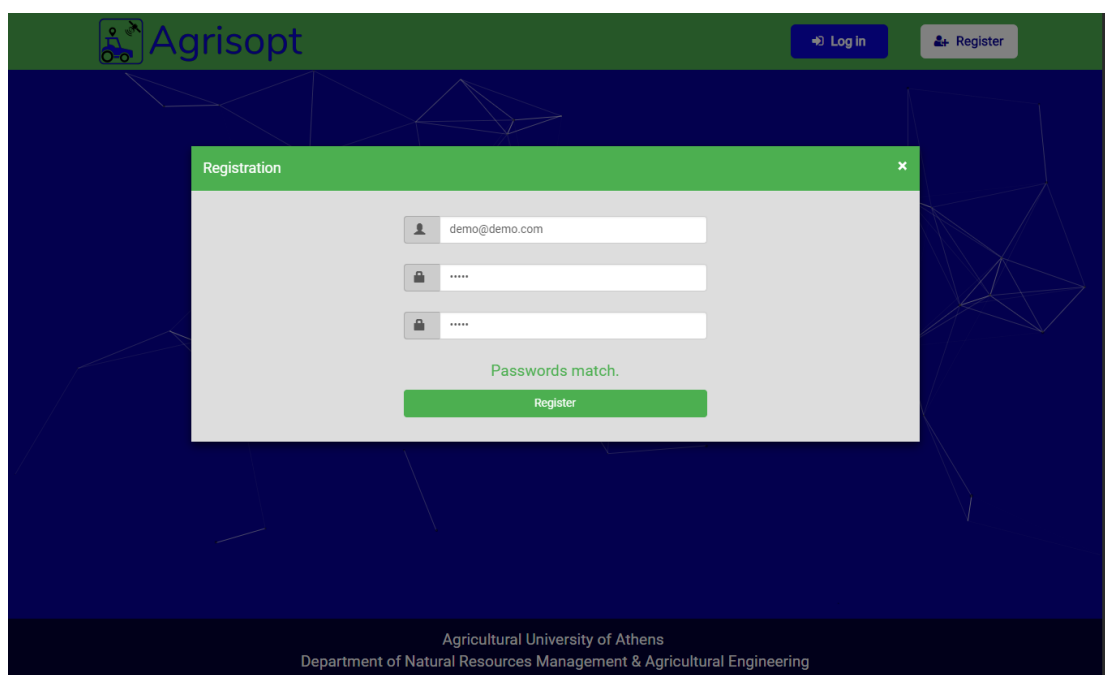
Δημιουργία λογαριασμού



The screenshot shows the Agrisopt website's registration form. The form is titled "Registration" and is displayed in a modal window. It contains three input fields: "Enter your email", "Enter your password", and "Re-enter your password". A green "Register" button is located below the fields. The background of the website is dark blue with a geometric pattern. The top navigation bar includes the Agrisopt logo, the version number "2.2 RC1", and "Log In" and "Register" buttons. The footer text reads "Agricultural University of Athens" and "Department of Natural Resources Management & Agricultural Engineering".

Εικόνα 4.3: Η φόρμα δημιουργίας λογαριασμού

Στην εικόνα 4.3 βλέπουμε τη φόρμα για τη δημιουργία ενός λογαριασμού χρήστη. Τα στοιχεία που ζητούνται από το χρήστη είναι ένας λογαριασμός email και να πληκτρολογήσει τον κωδικό της επιλογής του δύο φορές.

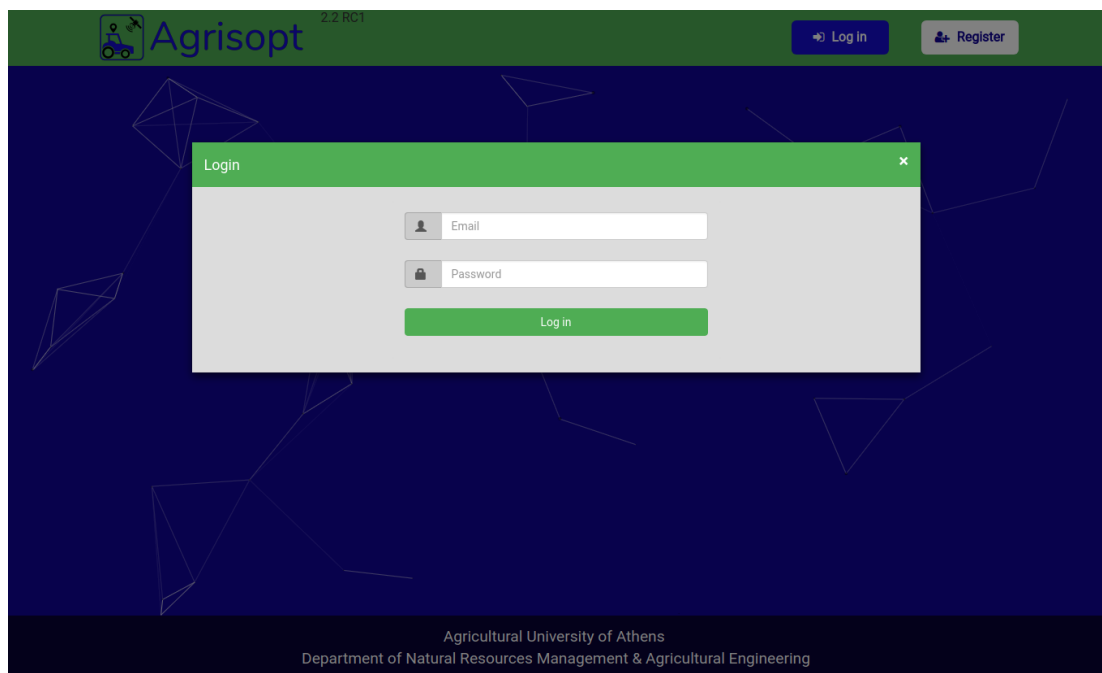


The screenshot shows the Agrisopt website's registration form with the fields filled. The "Enter your email" field contains "demo@demo.com". The "Enter your password" and "Re-enter your password" fields contain ".....". A green message "Passwords match." is displayed below the password fields. The "Register" button is still visible. The background and navigation elements are the same as in the previous screenshot.

Εικόνα 4.4: Η φόρμα δημιουργίας λογαριασμού συμπληρωμένη

Κατά τη συμπλήρωση των κωδικών του λογαριασμού, όπως φαίνεται στην εικόνα 4.4 γίνεται ένας έλεγχος για το αν ταιριάζουν οι δύο κωδικοί μεταξύ τους και εμφανίζεται το ανάλογο μήνυμα στον χρήστη.

Είσοδος στο λογαριασμό

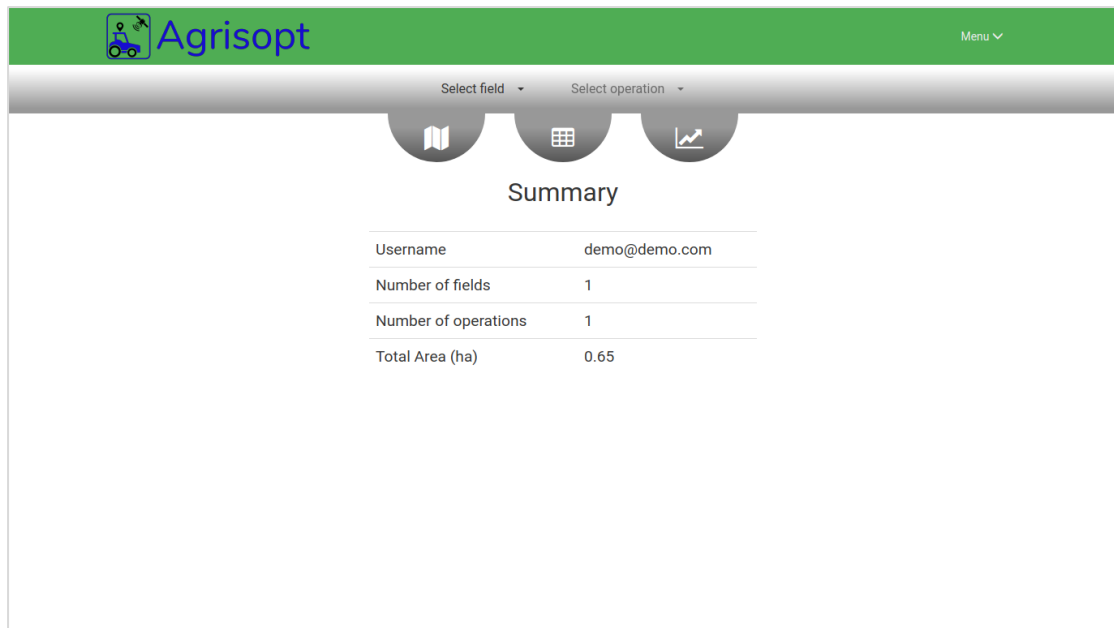


Εικόνα 4.5: Η φόρμα εισόδου στο λογαριασμό

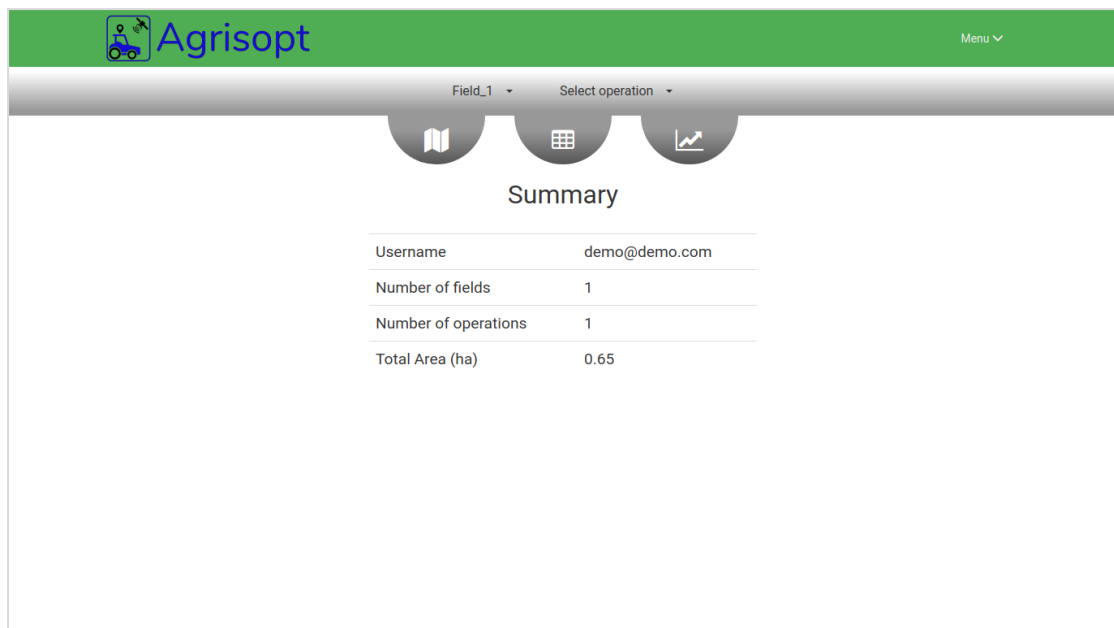
Στην εικόνα 4.5 εμφανίζεται η φόρμα εισόδου στο λογαριασμό. Από το χρήστη απαιτείται να πληκτρολογήσει το όνομα χρήστη και τον κωδικό.

Αρχική σελίδα χρήστη

Στην εικόνα 4.6 βλέπουμε την αρχική εικόνα που βλέπει ο χρήστης όταν συνδέεται στο λογαριασμό του (αφού έχει ψηφιοποιήσει ένα τουλάχιστον αγρόκτημα και αποθηκεύσει τουλάχιστον μία επέμβαση ψεκασμού). Στη σελίδα αυτή υπάρχουν οι επιλογές "Select field" και "Select operation", με τη 2^η να είναι μπλοκαρισμένη αρχικά και να ενεργοποιείται η δυνατότητα επιλογής αφού επιλεγθεί πρώτα ένα αγρόκτημα (εικόνα 4.7). Κάτω από τις δύο επιλογές υπάρχουν τρία κουμπιά τα οποία όταν επιλεγθούν εμφανίζουν τις αντίστοιχες όψεις με τα εικονίδια τους. Στα αριστερά είναι το εικονίδιο για την προβολή χάρτη, στη μέση είναι το εικονίδιο για την προβολή πίνακα με δεδομένα και στα δεξιά είναι το εικονίδιο με την προβολή του γραφήματος της ταχύτητας. Τέλος, πάνω δεξιά στην πράσινη μπάρα υπάρχει η επιλογή του μενού της εφαρμογής (εικόνα 4.8).

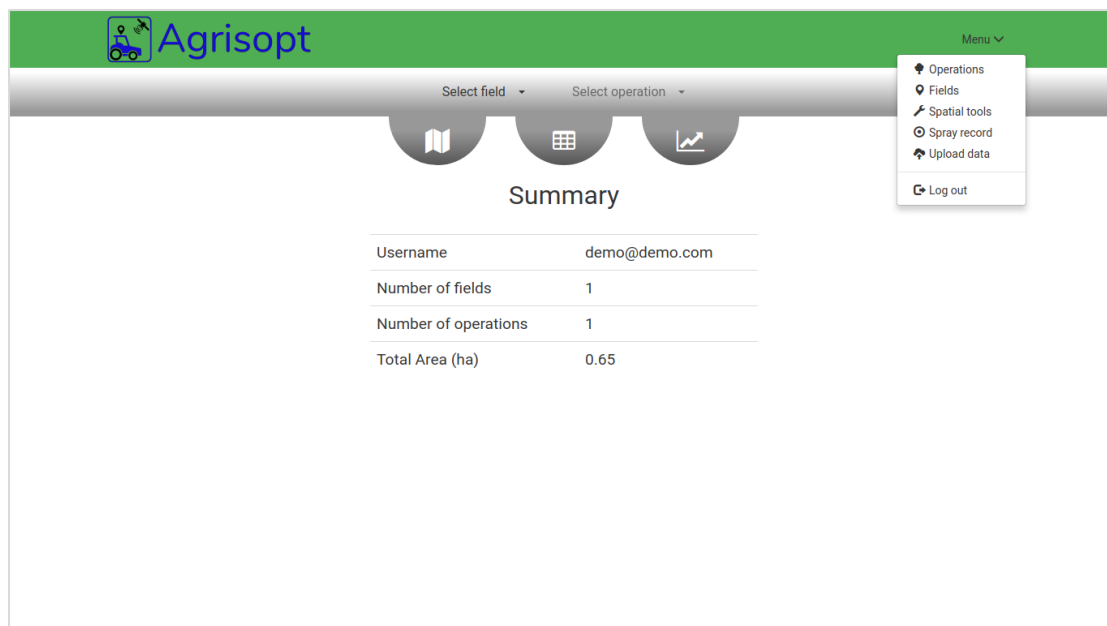


Εικόνα 4.6: Η αρχική εικόνα που βλέπει ο χρήστης όταν μπαίνει στο λογαριασμό του



Εικόνα 4.7: Όταν ο χρήστης επιλέγει ένα αγρόκτημα (“Select field”) ενεργοποιείται η δυνατότητα επιλογής επέμβασης ψεκασμού (“Select operation”)

Στην εικόνα 4.8 βλέπουμε το μενού περιήγησης της εφαρμογής. Η αρχική σελίδα που εμφανίζεται σε ένα χρήστη αντιστοιχεί στην επιλογή “Operations”.



Εικόνα 4.8: Πάνω δεξιά εμφανίζεται το μενού μέσω του οποίου μπορεί να περιηγηθεί ο χρήστης

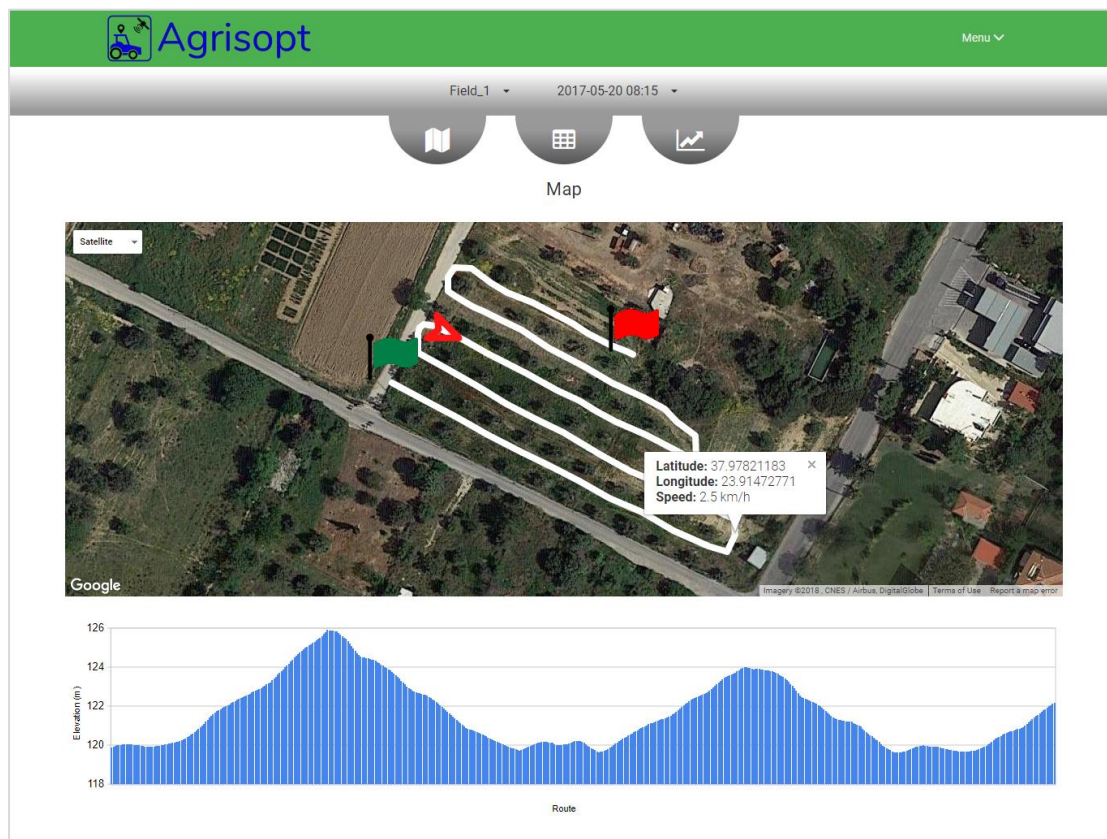
Όπως φαίνεται στην εικόνα 4.9 όταν γίνει η επιλογή μιας επέμβασης ψεκασμού εμφανίζεται ένας πίνακας με δεδομένα. Η εμφάνιση αυτού του πίνακα μπορεί να γίνει και με το μεσαίο κουμπί που έχει το εικονίδιο πίνακα.



Εικόνα 4.9: Η επιλογή του πίνακα με τα δεδομένα

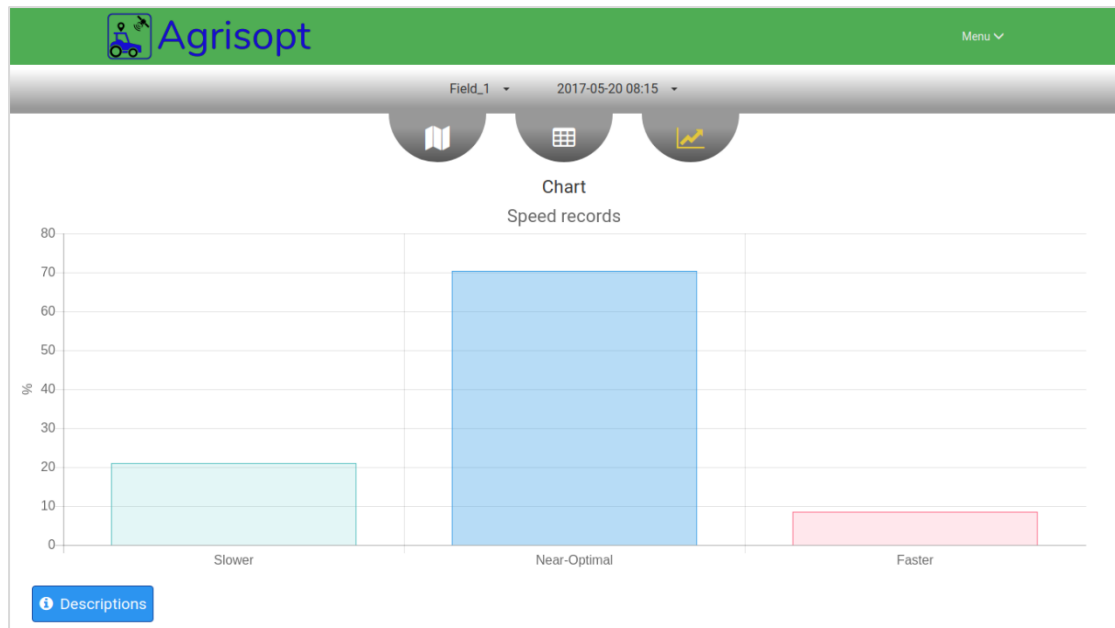
Στην εικόνα 4.10 βλέπουμε την όψη που βλέπει ο χρήστης στην προβολή χάρτη. Για να εμφανιστεί, ο χρήστης πρέπει να επιλέξει το αριστερό κουμπί με το εικονίδιο χάρτη. Στο πάνω μέρος εμφανίζεται ο χάρτης με τη διαδρομή, εκεί ο χρήστης

επιλέγοντας ένα σημείο πάνω στη διαδρομή, μπορεί να δει τα στοιχεία των συντεταγμένων του και την ταχύτητα διέλευσης. Στα δύο άκρα της διαδρομής εμφανίζονται δύο εικονίδια, το ένα είναι μια πράσινη σημαία που συμβολίζει την αρχή της διαδρομής και αντίστοιχα η κόκκινη σημαία, το τέλος της. Επίσης, κατά την αρχική φόρτωση του χάρτη τρέχει κατά μήκος της διαδρομής ένα κόκκινο τριγωνικό εικονίδιο. Τέλος, κάτω από το χάρτη υπάρχει το γράφημα με τη μεταβολή του υψομέτρου κατά τη διάρκεια της διαδρομής.



Εικόνα 4.10: Η επιλογή προβολής χάρτη

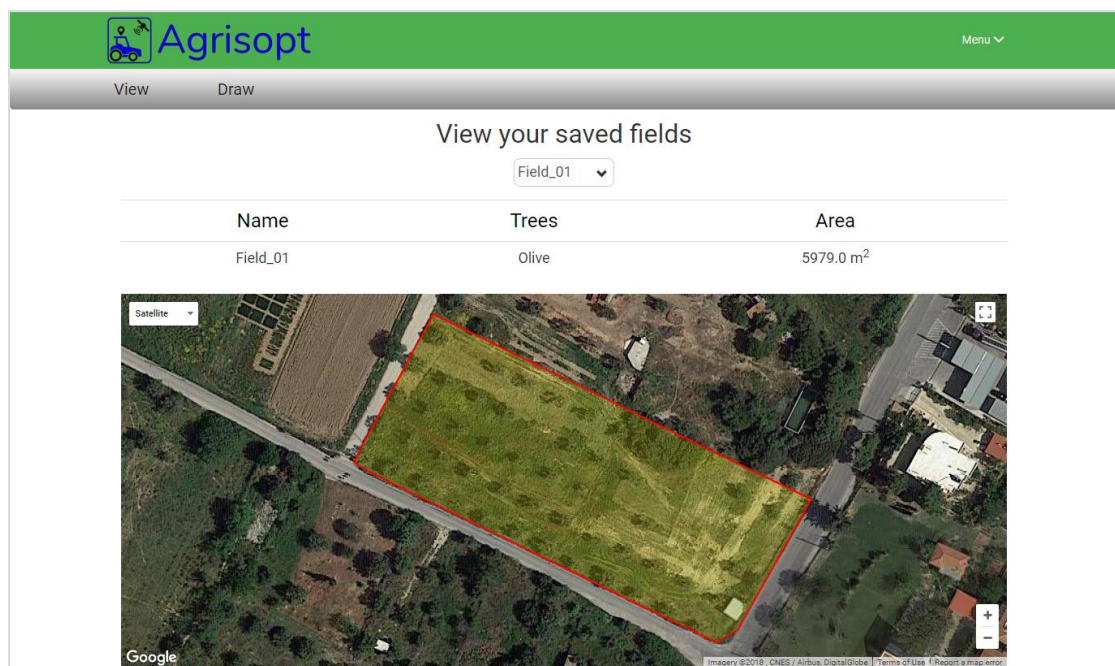
Στην εικόνα 4.11 βλέπουμε το γράφημα της ταχύτητας, όπου παρουσιάζεται το ποσοστό των εγγραφών των δεδομένων μιας επέμβασης ψεκασμού, με ταχύτητα κοντά στη βέλτιστη (μεσαία μπάρα – “Near-Optimal”), μικρότερη από το -10% της βέλτιστης (αριστερή μπάρα – “Slower”) και μεγαλύτερη από το +10% της βέλτιστης (δεξιά μπάρα – “Faster”). Υπάρχει επίσης, το κουμπί “Descriptions” από όπου ο χρήστης μπορεί να διαβάσει μια σύντομη περιγραφή των στοιχείων του γραφήματος.



Εικόνα 4.11: Η επιλογή προβολής του γραφήματος της ταχύτητας

Επιλογή από το Menu: Fields

Στην εικόνα 4.12 βλέπουμε την επιλογή “View”, όπου ο χρήστης μπορεί να δει τις εκτάσεις που έχει ψηφιοποιήσει και αποθηκεύσει. Στα στοιχεία του κάθε αγροτεμαχίου εμφανίζετε το όνομά του, ο τύπος της καλλιέργειας και το εμβαδόν του σε m² ή ha (ανάλογα με την τιμή).



Εικόνα 4.12: Ένα αποθηκευμένο αγροτεμάχιο

Στην εικόνα 4.13 εμφανίζεται η φόρμα μέσω της οποίας ένας χρήστης μπορεί να καταχωρήσει ένα αγροτεμάχιο. Αρχικά ο χρήστης πρέπει να βρει την τοποθεσία της επιλογής του. Αυτό μπορεί να γίνει μέσω:

1. Της εύρεσης της θέσης του εκείνη τη στιγμή, ή
2. καταχώρησης των συντεταγμένων στα αντίστοιχα πεδία, ή
3. του χάρτη

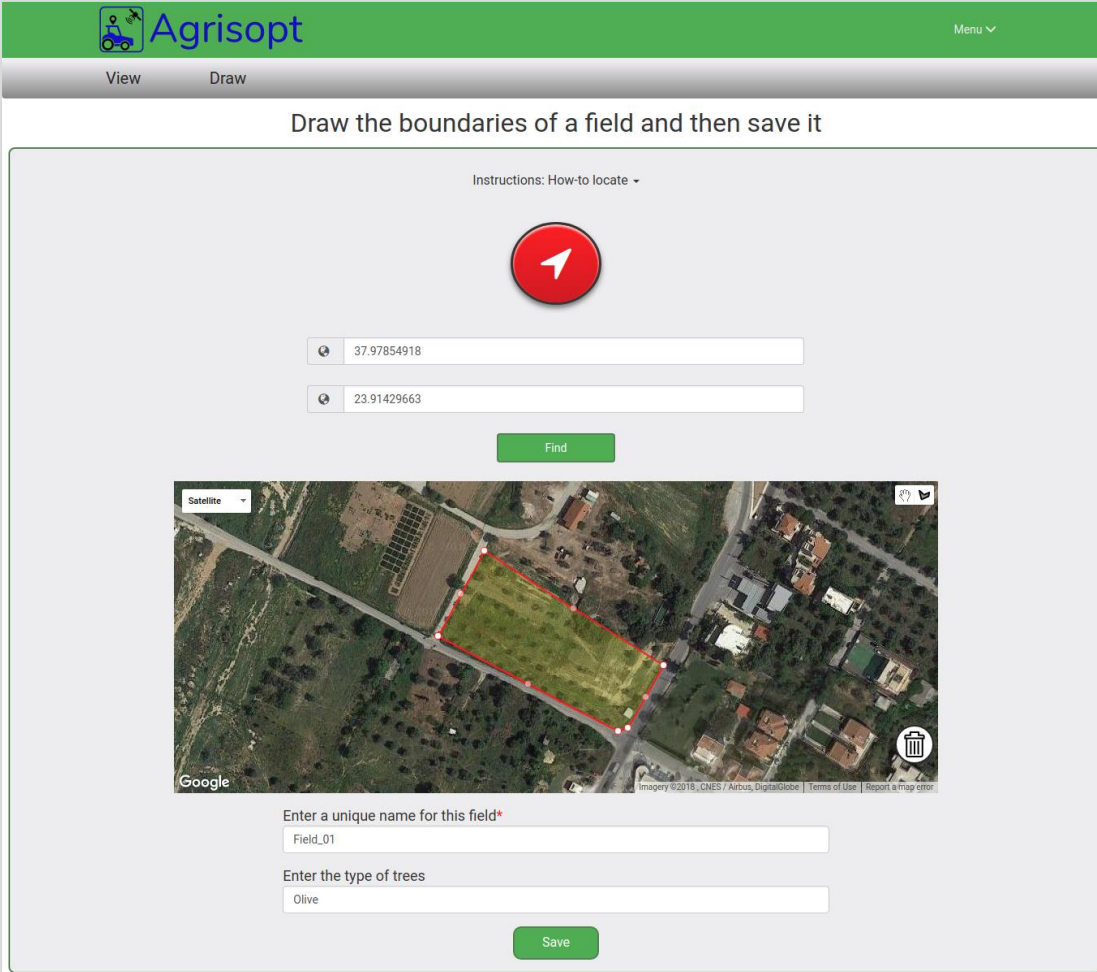
Για την 1^η περίπτωση θα πρέπει να έχει ενεργοποιημένο το GPS της συσκευής του ή/και ενεργή σύνδεση στο διαδίκτυο (χρήση Geolocation API). Στην περίπτωση χρήσης του GPS μπορεί προφανώς να πετύχει μεγαλύτερη ακρίβεια.

Για τη σχεδίαση των ορίων του αγροτεμαχίου μπορεί να χρησιμοποιήσει μέσα στο χάρτη το άνω δεξιά εικονίδιο που αναπαριστά ένα πολύγωνο. Επίσης, μπορεί να κάνει παύση της ψηφιοποίησης μέσω του εικονιδίου που αναπαριστά ένα χέρι. Τέλος, μπορεί να διαγράψει το σχήμα που σχεδίασε μέσω του αντίστοιχου κουμπιού κάτω δεξιά.

The screenshot shows the Agrisopt web application interface. At the top, there is a green header with the Agrisopt logo and a 'Menu' dropdown. Below the header, there are 'View' and 'Draw' buttons. The main content area is titled 'Draw the boundaries of a field and then save it'. It contains a section for 'Instructions: How-to locate' with a red location pin icon. Below this are input fields for 'Latitude' and 'Longitude', and a 'Find' button. A map of the Mediterranean region is displayed, showing countries like Italy, Greece, Turkey, and others. Below the map, there are two more input fields: 'Enter a unique name for this field*' (with the example 'e.g. Field_1') and 'Enter the type of trees' (with the example 'e.g. Apple'). A 'Save' button is located at the bottom of the form.

Εικόνα 4.13: Η φόρμα σχεδίασης και αποθήκευσης ενός αγροτεμαχίου

Στη συνέχεια θα πρέπει να εισάγει ένα όνομα για αυτό το αγροτεμάχιο (υποχρεωτικό πεδίο) και τον τύπο των δένδρων (προαιρετικό πεδίο). Τα δεδομένα αυτά θα σταλούν για αποθήκευση στη βάση δεδομένων με το πάτημα του κουμπιού “Save”.

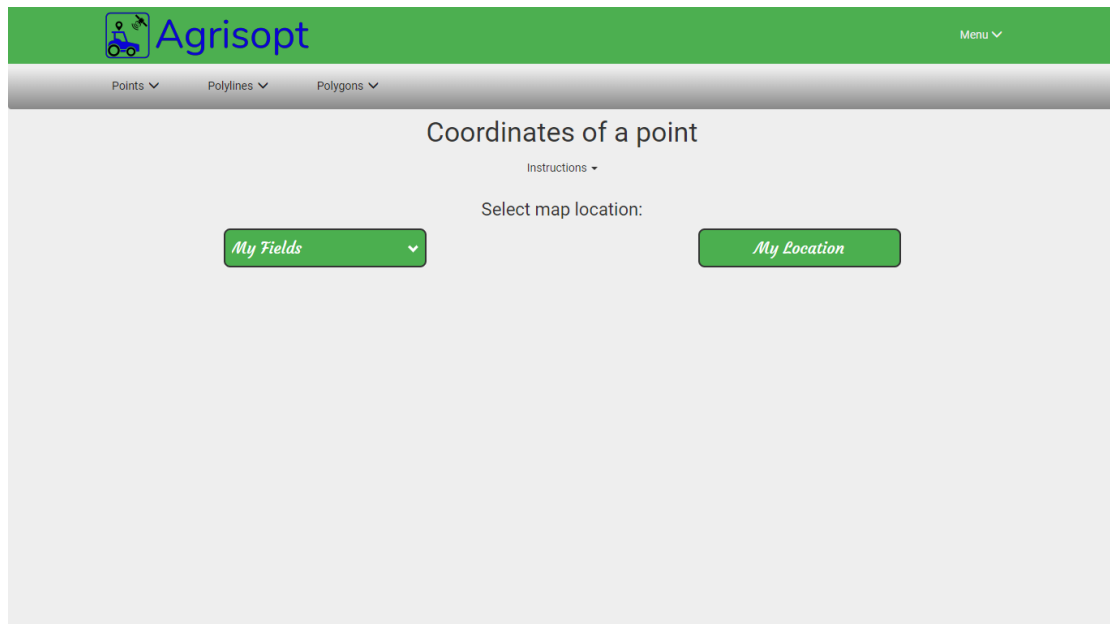


The screenshot shows the Agrisoft web application interface. At the top, there is a green header with the Agrisoft logo and a 'Menu' dropdown. Below the header, there are 'View' and 'Draw' buttons. The main content area has a title 'Draw the boundaries of a field and then save it' and a sub-header 'Instructions: How-to locate'. A large red circular button with a white arrow icon is centered. Below it are two input fields for coordinates: the first contains '37.97854918' and the second contains '23.91429663'. A green 'Find' button is positioned below the second input field. Below the 'Find' button is a satellite map showing a field outlined in red. The map includes a 'Satellite' dropdown menu, a 'Google' logo, and a 'Report a map error' link. Below the map are two input fields: the first is labeled 'Enter a unique name for this field*' and contains 'Field_01', and the second is labeled 'Enter the type of trees' and contains 'Olive'. A green 'Save' button is located at the bottom of the form.

Εικόνα 4.14: Ένα παράδειγμα συμπλήρωσης της φόρμας ψηφιοποίησης αγροτεμαχίου

Επιλογή από το Menu: Spatial tools

Στην σελίδα “Spatial tools” υπάρχουν επτά εργαλεία βασικών χωρικών υπολογισμών που ομαδοποιούνται σε τρεις κατηγορίες ανάλογα με το είδος του χωρικού δεδομένου που χρησιμοποιείται. Για κάθε επιλογή υπάρχουν ξεχωριστές οδηγίες χρήσης οι οποίες μπορούν να εμφανιστούν με αριστερό click στο “Instructions”. Η εμφάνιση του χάρτη μπορεί να γίνει μέσω δύο επιλογών, η 1^η είναι το “My Fields” και η 2^η το “My Location”. Το “My Fields” εμφανίζει μια λίστα με όλα τα αποθηκευμένα αγροτεμάχια, ενώ το “My Location” εμφανίζει τη θέση που βρίσκεται εκείνη τη στιγμή ο χρήστης. Στη 2^η περίπτωση θα πρέπει να είναι ενεργοποιημένο το GPS ή/και να υπάρχει ενεργή σύνδεση στο Internet (χρήση Geolocation API).

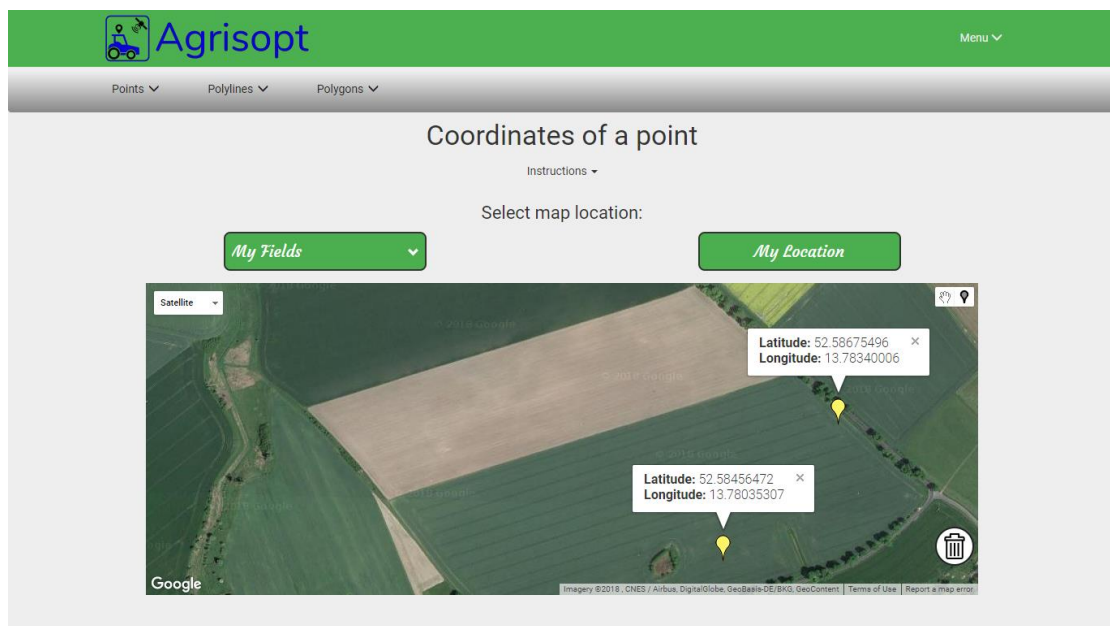


Εικόνα 4.15: Η 1^η επιλογή της κατηγορίας “Points” το “Coordinates of a point”

Οι επιλογές για το “Points”:

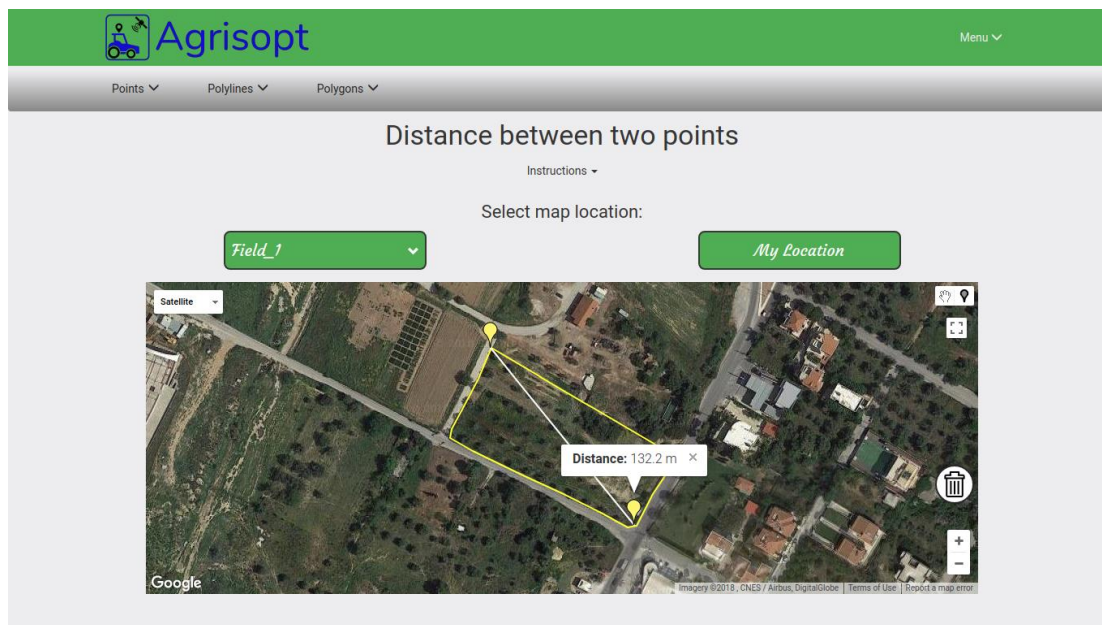
1. Coordinates of a point.
2. Distance between two points.
3. Middle point.
4. Buffer zone.

Στην εικόνα 4.16 παρατηρούμε την επιλογή “Points” με την εμφάνιση χάρτη μέσω του “*My Location*”.



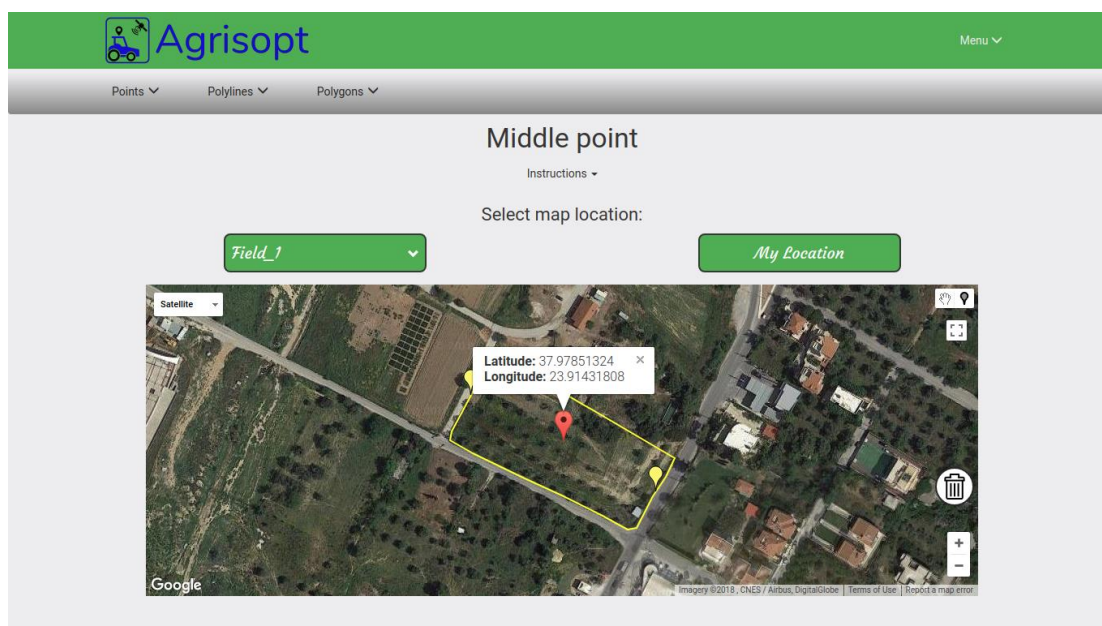
Εικόνα 4.16: Το “Coordinates of a point” αφού έχει επιλεγεί το “My Location” και έχουν επιλεγθεί δύο σημεία πάνω στο χάρτη εμφανίζει τις συντεταγμένες τους

Στην εικόνα 4.17 βλέπουμε την επιλογή “Distance between two points” με εμφάνιση του χάρτη μέσω του “My Fields”. Ο χρήστης κάνοντας αριστερό click σε δύο σημεία μπορεί να δει τη μεταξύ τους απόσταση.



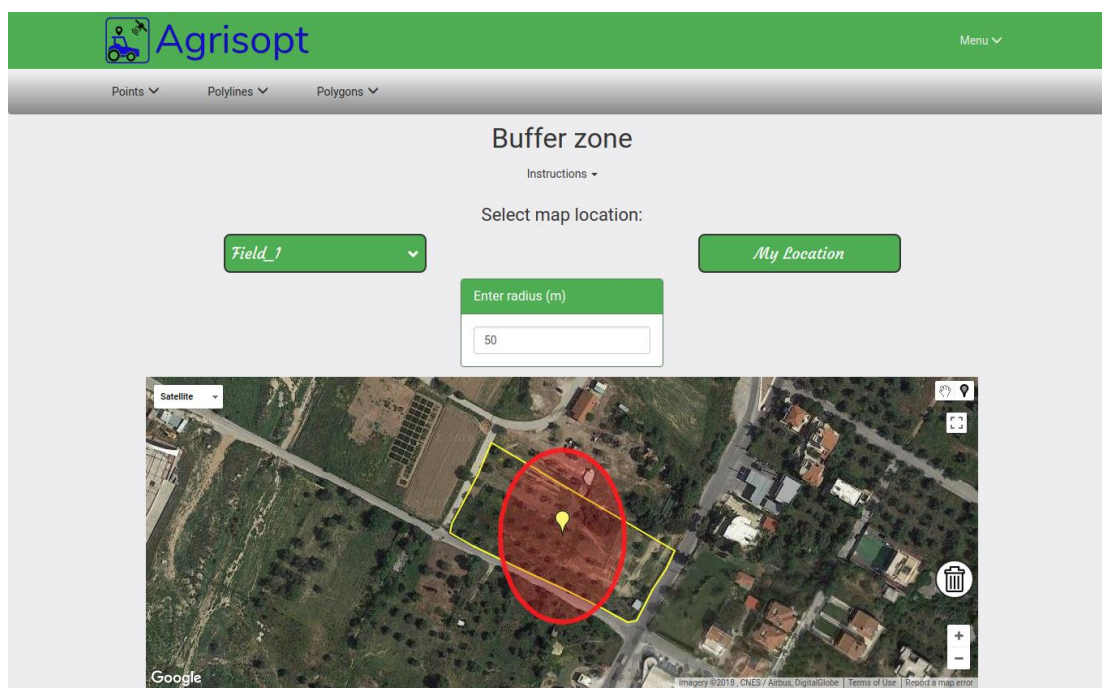
Εικόνα 4.17: Η επιλογή “Distance between two points”

Στην εικόνα 4.18 εμφανίζεται η επιλογή “Middle point”, ο χρήστης κάνοντας click σε δύο σημεία μπορεί να δει τις συντεταγμένες του μέσου τους.



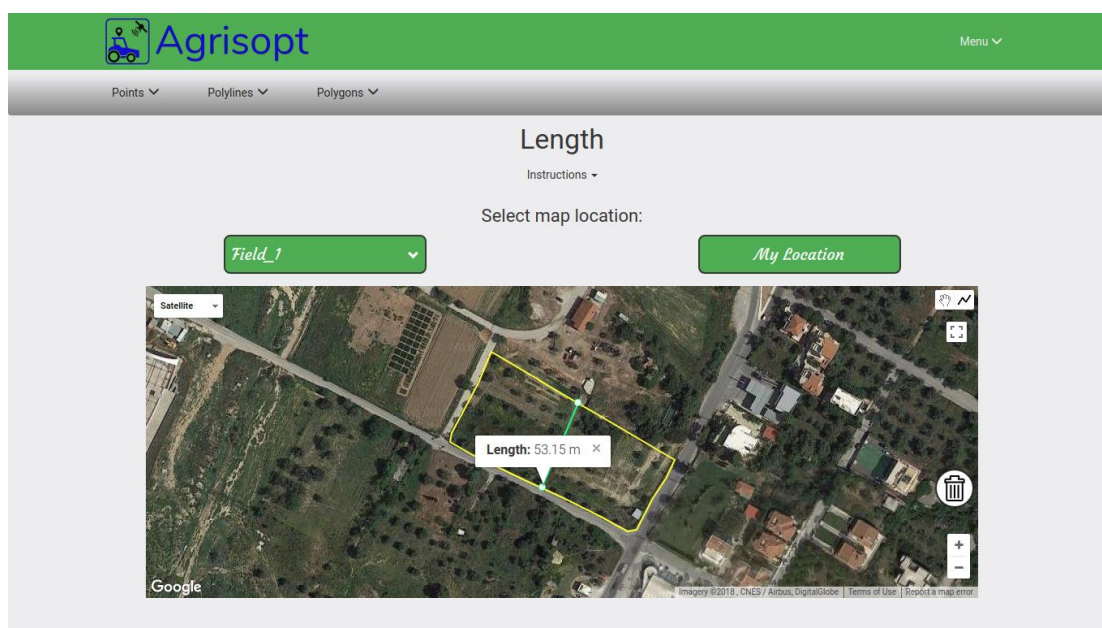
Εικόνα 4.18: Η επιλογή “Middle point”

Στην εικόνα 4.19 εμφανίζεται η επιλογή “Buffer zone”, εκεί ο χρήσης εισάγοντας την ακτίνα της προτίμησής του και κάνοντας αριστερό click σε ένα σημείο στο χάρτη μπορεί να δει τη buffer zone γύρω από αυτό.



Εικόνα 4.19: Η επιλογή “Buffer zone”

Στην εικόνα 4.20 εμφανίζεται η μοναδική επιλογή από την κατηγορία “Polylines”, το “Length”. Στην επιλογή αυτή ο χρήστης κάνοντας αριστερό click σε δύο ή και περισσότερα σημεία πάνω στο χάρτη μπορεί αν δει το συνολικό μήκος της γραμμής που σχηματίζεται.

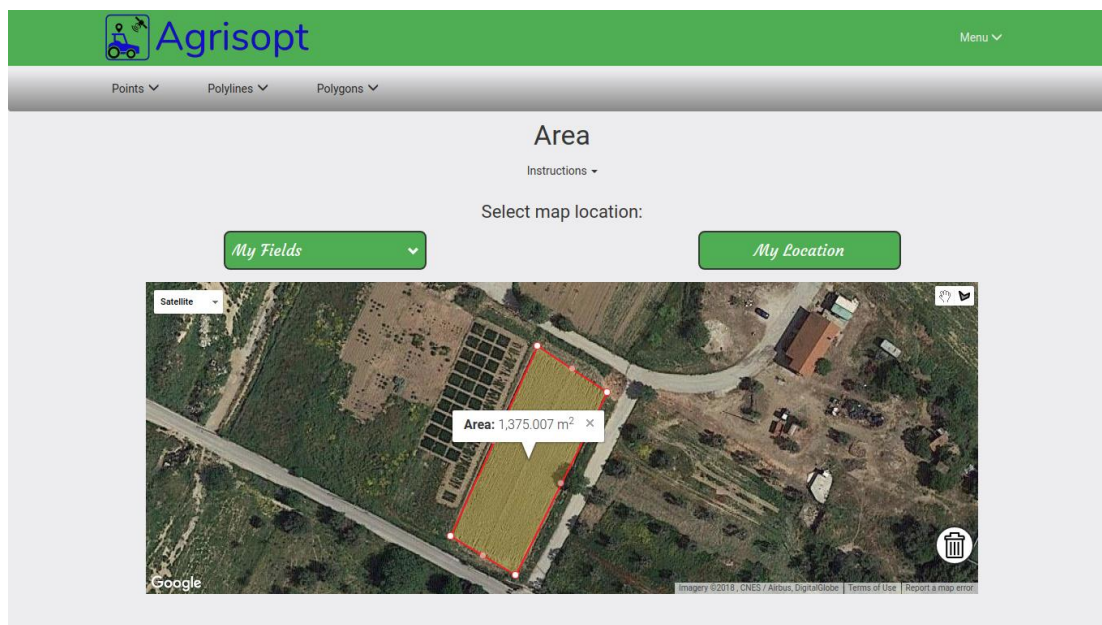


Εικόνα 4.20: Η επιλογή “Length”

Η κατηγορία “Polygons” έχει τις επιλογές:

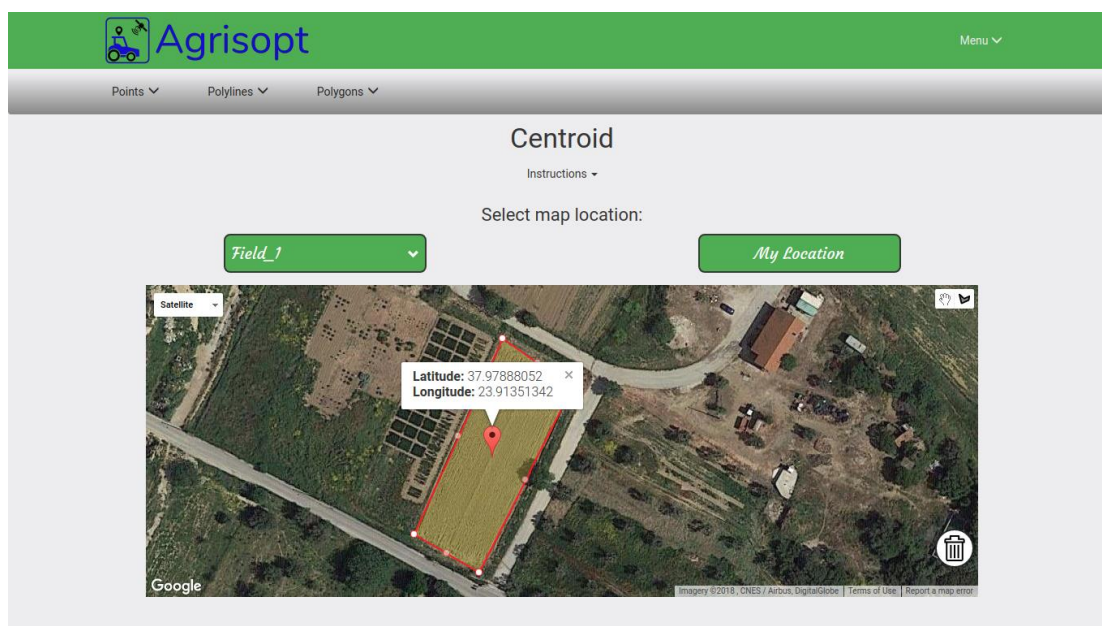
1. Area.
2. Centroid.

Στην εικόνα 4.21 παρουσιάζεται η επιλογή “Area” όπου ο χρήστης μπορεί να χαράξει τα όρια μιας έκτασης και να εμφανιστεί το συνολικό της εμβαδόν.



Εικόνα 4.21: Η επιλογή “Area”

Στην εικόνα 4.22 εμφανίζεται η επιλογή “Centroid” όπου ο χρήστης σχεδιάζοντας ένα πολύγωνο μπορεί να βρει τις συντεταγμένες του κέντρου του.

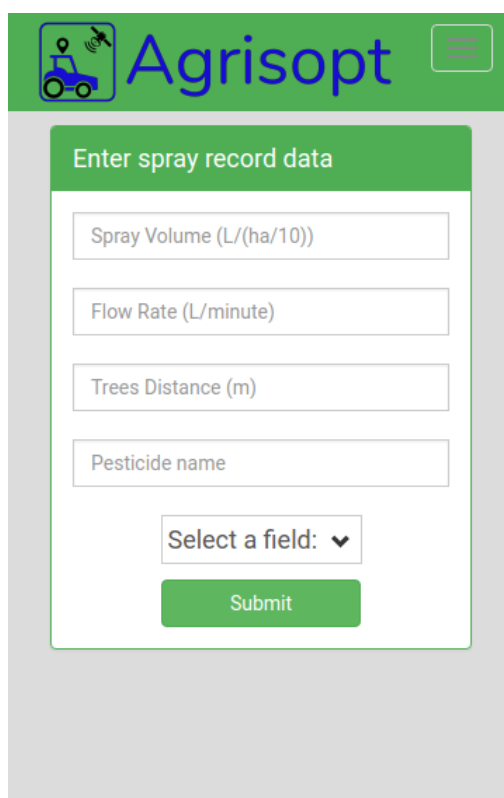


Εικόνα 4.22: Η επιλογή “Centroid”

Επιλογή από το Menu: Spray record

Μέσω της λειτουργίας “Spray record” ένας χρήστης μπορεί με μια συσκευή που έχει GPS να καταγράψει χωρικά δεδομένα κατά τη διάρκεια ενός ψεκασμού σε δενδρώνδεις καλλιέργειες. Για να γίνει αυτό θα πρέπει να έχει ενεργοποιήσει το GPS πιο πριν. Επίσης, είναι εφικτή η λειτουργία και σε τοποθεσίες που δεν είναι δυνατή η σύνδεση στο διαδίκτυο, φθάνει να έχει φορτώσει τη σελίδα αυτή νωρίτερα.

Στην εικόνα 4.24 ζητείται από το χρήστη η καταχώρηση των στοιχείων μια επέμβασης ψεκασμού μέσω μια φόρμας για τον υπολογισμό της βέλτιστης ταχύτητας κίνησης. Επίσης, ο χρήστης θα πρέπει να επιλέξει το αγροτεμάχιο στο οποίο θα γίνει αυτός ο ψεκασμός.



The image shows a mobile application interface for Agrisopt. At the top, there is a green header with the Agrisopt logo and a menu icon. Below the header is a white form titled "Enter spray record data". The form contains four input fields: "Spray Volume (L/(ha/10))", "Flow Rate (L/minute)", "Trees Distance (m)", and "Pesticide name". Below these fields is a dropdown menu labeled "Select a field:" and a green "Submit" button.

Εικόνα 4.23: Η φόρμα καταχώρησης των στοιχείων του ψεκασμού

Στην εικόνα 4.24 εμφανίζεται η φόρμα καταχώρησης των στοιχείων του ψεκασμού μετά τη συμπλήρωσής της, όπου εμφανίζεται ένα μήνυμα στον χρήστη που τον ειδοποιεί να βεβαιωθεί (πριν από την οριστική καταχώρηση) ότι το GPS είναι ενεργό.

Στην εικόνα 4.25 παρουσιάζεται η λειτουργία της καταγραφής των χωρικών δεδομένων. Αρχικά, χρήστης θα πρέπει να πατήσει το κουμπί “Check Location” για να ελεγχθεί από την εφαρμογή η ικανότητα εύρεσης τοποθεσίας από τη συσκευή εκείνη τη στιγμή.

Agrisopt

Enter spray record data

100

50

10

Insignia

Field_1

Before submitting, make sure that the GPS is turned on!

Submit

Εικόνα 4.24: Η φόρμα καταχώρησης των στοιχείων του ψεκασμού μετά τη συμπλήρωσή της

Agrisopt

Check Location

Press the "Check Location" to verify that the location service is working.

Start recording

Optimal Speed: 3.00 km/h

Latitude

Longitude

Speed (km/h)

Accuracy (m)

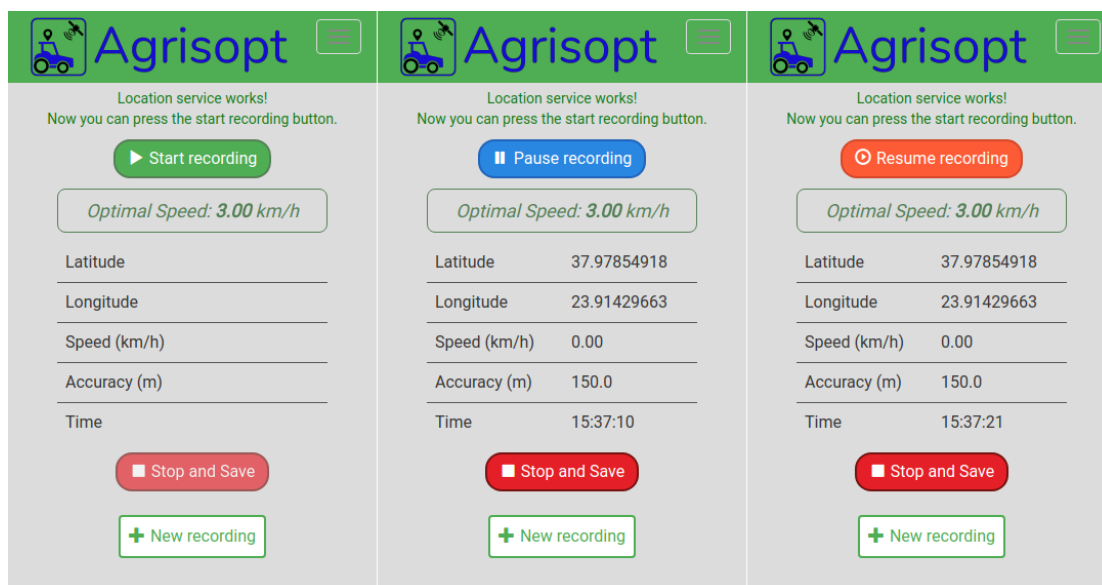
Time

Stop and Save

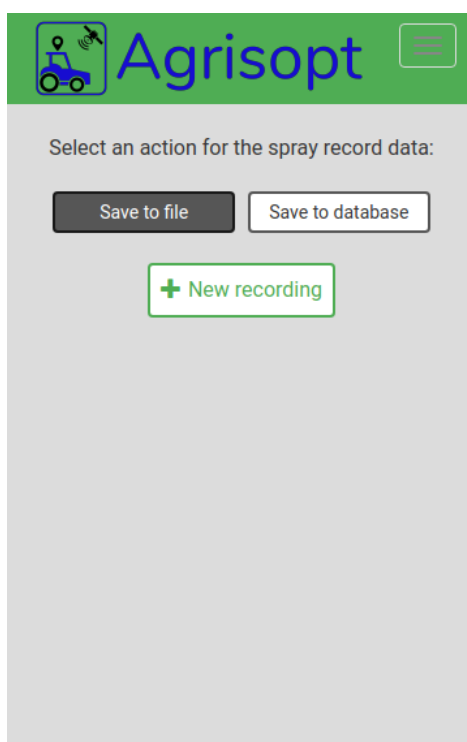
+ New recording

Εικόνα 4.25: Η λειτουργία καταγραφής των χωρικών δεδομένων

Αν ο έλεγχος είναι επιτυχής τότε ο χρήστης μπορεί να ξεκινήσει την καταγραφή όπως εμφανίζεται στην αριστερή όψη της εικόνας 4.26 (κουμπί “Start recording”). Κατά τη διάρκεια που η καταγραφή είναι ενεργή, ο χρήστης μπορεί να τη διακόψει προσωρινά μέσω του κουμπιού “Pause recording” (μεσαία όψη της εικόνας 4.26) και να την ξανακάνει ενεργή μέσω του κουμπιού “Resume recording” (δεξιά όψη της εικόνας 4.26).



Εικόνα 4.26: Η καταγραφή ενός ψεκασμού σε λειτουργία

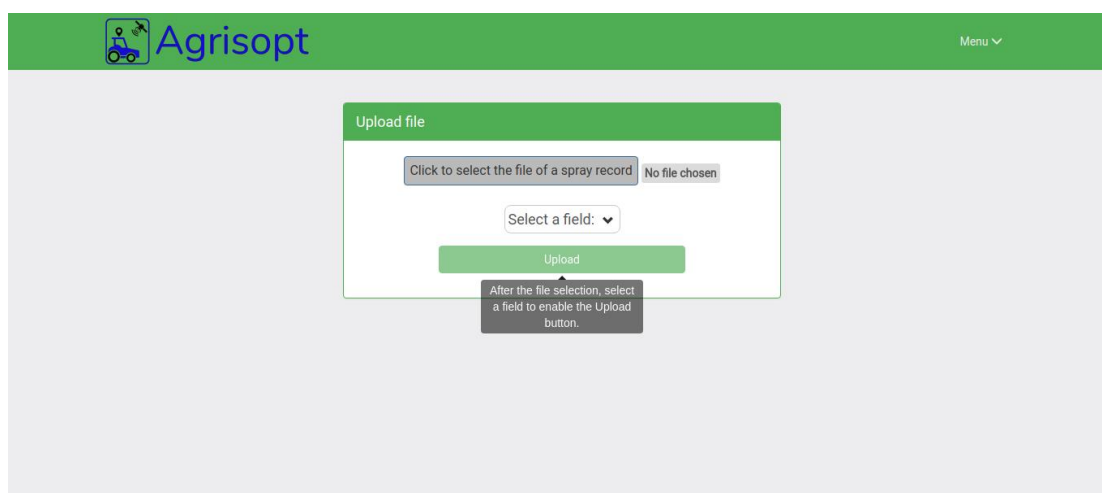


Εικόνα 4.27: Οι επιλογές αποθήκευσης των δεδομένων της καταγραφής

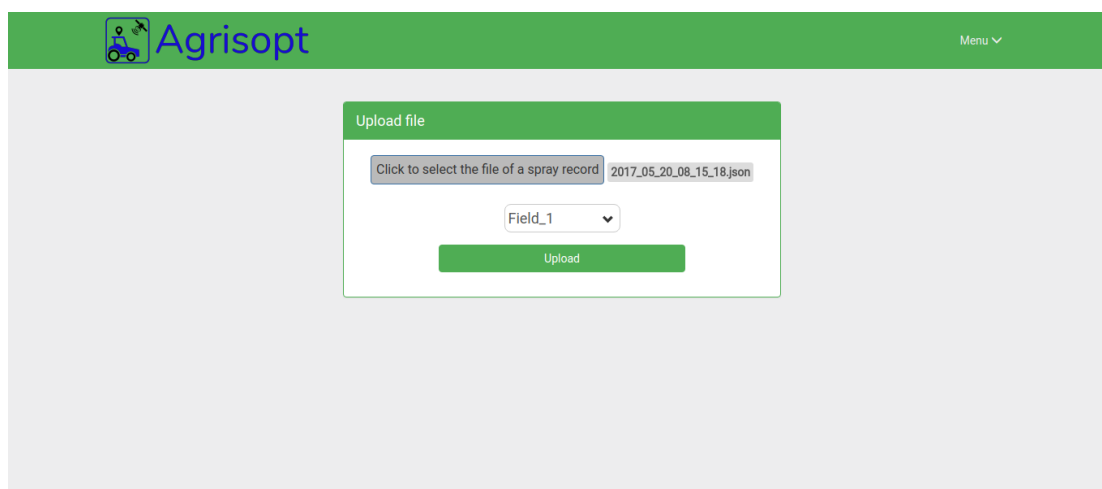
Στην εικόνα 4.27 εμφανίζονται οι δύο επιλογές αποθήκευσης των δεδομένων μιας καταγραφής ψεκασμού. Κατά τη διάρκεια που εμφανίζεται αυτή η σελίδα γίνεται ένας έλεγχος για την ύπαρξη ή μη ενεργούς σύνδεσης στο διαδίκτυο. Αν ο έλεγχος δείξει ότι δεν υπάρχει σύνδεση ή υπάρχει αλλά είναι πάρα πολύ αργή τότε το κουμπί “Save to database” γίνεται ανενεργό και ο χρήστης μπορεί να επιλέξει μόνο την επιλογή “Save to file”.

Επιλογή από το Menu: Upload data

Στην εικόνα 4.28 εμφανίζεται η φόρμα μεταφοράς ενός αρχείου με τα δεδομένα που καταγράφηκαν κατά τη διάρκεια ενός ψεκασμού. Ένας χρήστης μπορεί να αποθηκεύσει τα δεδομένα ενός ψεκασμού στο διακομιστή μέσω αυτής της φόρμας, αυτό μπορεί να χρειαστεί σε περίπτωση που δεν υπήρχε διαθέσιμη σύνδεση στο διαδίκτυο ικανή να μεταφέρει τα δεδομένα κατά το τέλος του ψεκασμού.



Εικόνα 4.28: Η φόρμα μεταφοράς αρχείου στο διακομιστή (αρχικά)



Εικόνα 4.29: Η φόρμα μεταφοράς αρχείου στο διακομιστή (μετά από τις επιλογές του χρήστη)

4.2 Έλεγχος της εφαρμογής

Για τον έλεγχο της λειτουργίας της εφαρμογής φορτώθηκαν δεδομένα στο διακομιστή από μια έκταση του Πανεπιστημίου στην περιοχή Γιαλού Σπάτων (Γεωγραφικό πλάτος: 37.97854918, Γεωγραφικό μήκος: 23.91429663), το είδος των δένδρων είναι ελιές και το φυτοφάρμακο του ψεκασμού είναι το Insignia 20 WG.

Στη συνέχεια ακολούθησε μια επαλήθευση των λειτουργιών και των χαρακτηριστικών της εφαρμογής.

Ο έλεγχος της εφαρμογής διακρίνεται σε πέντε διαφορετικές κατηγορίες:

1. Λειτουργικότητα.
2. Ευχρηστία.
3. Συμβατότητα.
4. Απόδοση.
5. Ασφάλεια.

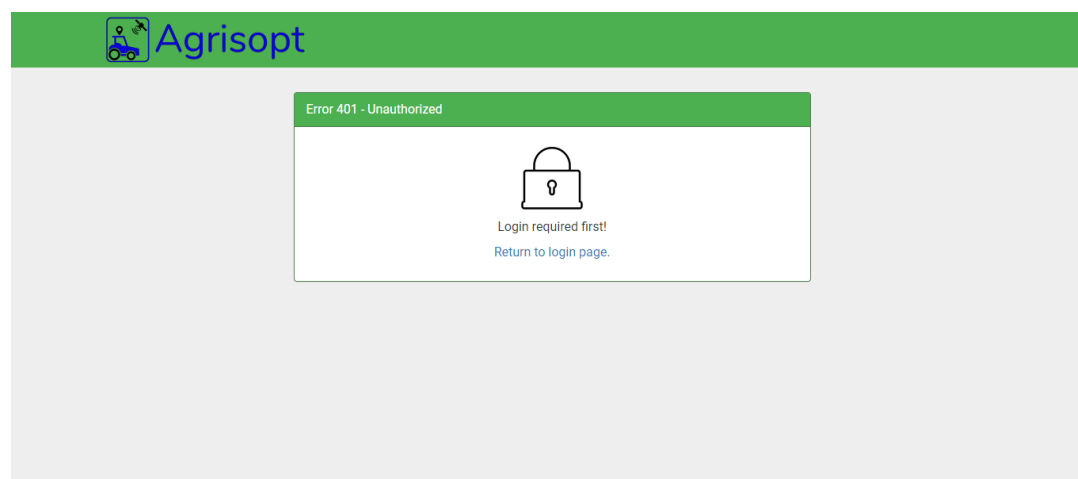
Οι κατηγορίες αυτές αναπτύσσονται ξεχωριστά η κάθε μία στη συνέχεια.

4.2.1 Λειτουργικότητα

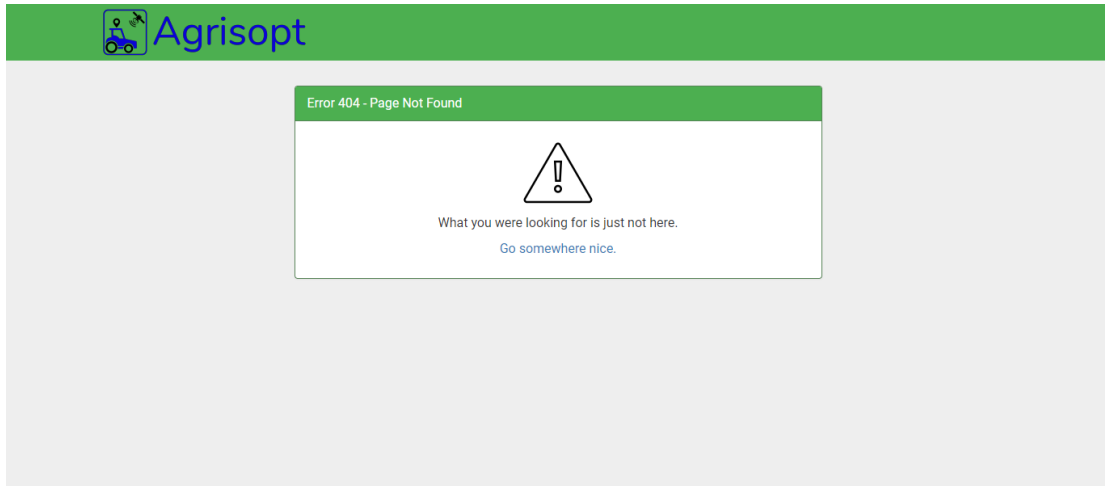
Στον έλεγχο Λειτουργικότητας πραγματοποιήθηκαν έλεγχοι για:

- Λάθη σε συνδέσμους (links).
- Λανθασμένες ανακατευθύνσεις (redirects).
- Το πώς ανταποκρίνεται η εφαρμογή σε δίκτυα χαμηλής ταχύτητας. Μέσω των Google Chrome Dev Tools δοκιμάστηκε η λειτουργία της εφαρμογής σε ταχύτητες δικτύων κινητής 2^{ης} (2G) και 3^{ης} γενιάς (3G), όπου όπως είναι λογικό αυξήθηκε ο χρόνος φόρτωσης των στοιχείων της κάθε σελίδας.
- Την εμφάνιση των σελίδων που αντιστοιχούν στους HTTP κωδικούς κατάστασης (HTTP status codes) 401, 404, 405 και 500.
- Τα μηνύματα που ενημερώνουν το χρήστη για: τα επιτυχή συμβάντα και τα σφάλματα που συμβαίνουν μετά από λανθασμένη εισαγωγή στοιχείων σε φόρμες ή μετά από κάποια πιθανή δυσλειτουργία.

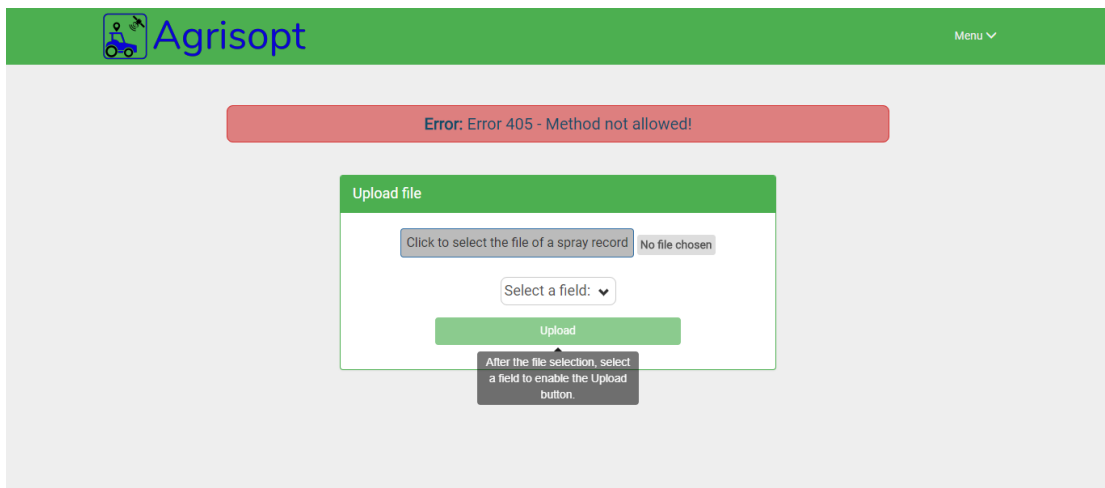
Στη συνέχεια ακολουθούν εικόνες από την εφαρμογή (screenshots), όπου εμφανίζονται οι σελίδες για τους HTTP κωδικούς κατάστασης 401, 404, 405 και 500.



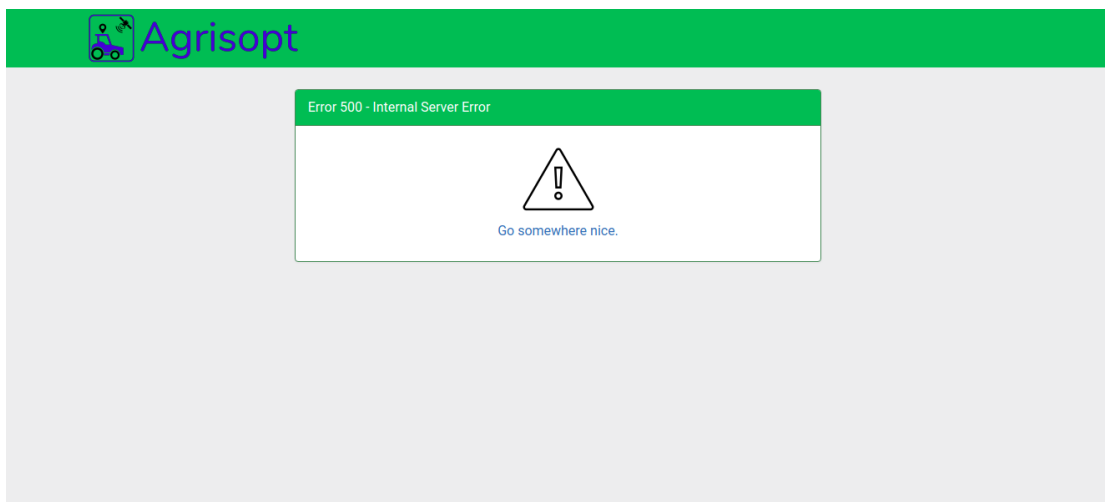
Εικόνα 4.30: Κωδικός 401 - Unauthorized



Εικόνα 4.31: Κωδικός 404 - Page not Found



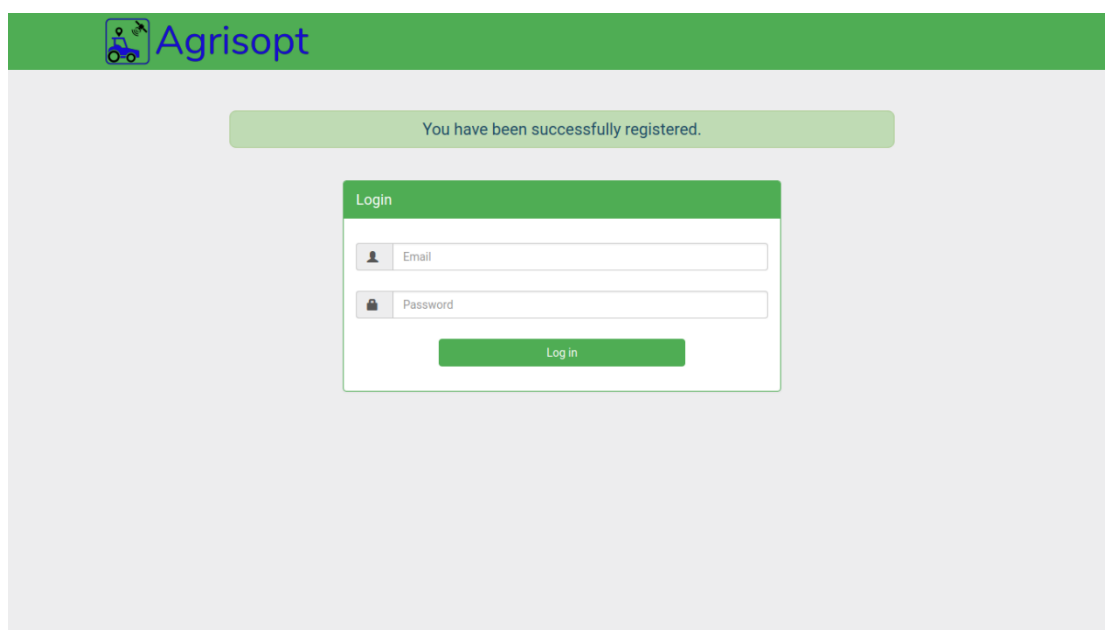
Εικόνα 4.32: Κωδικός 405 - Method not allowed



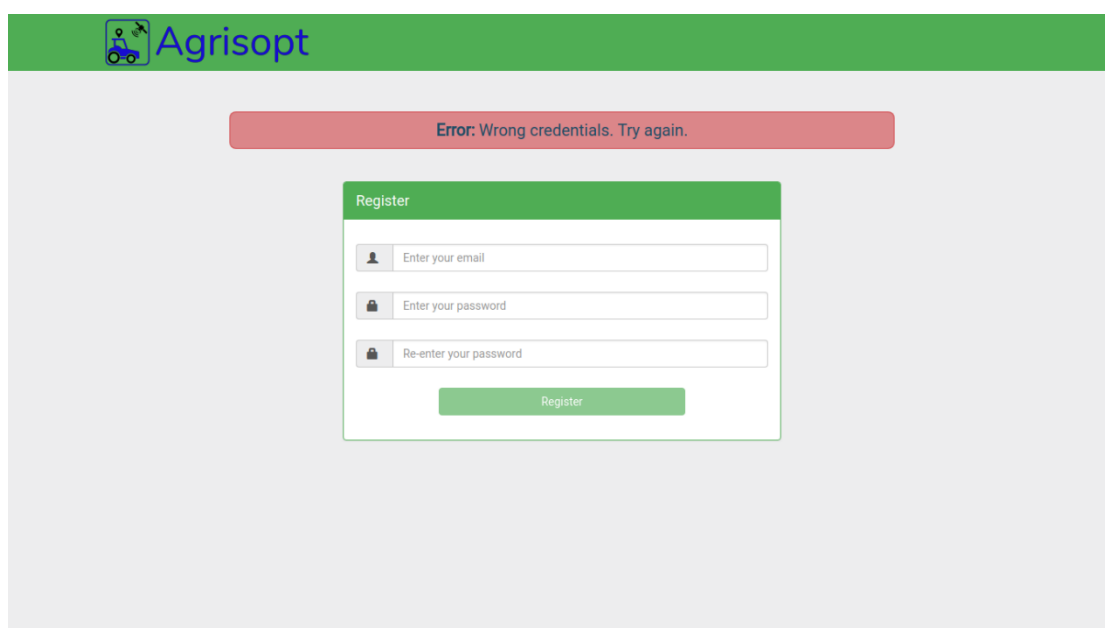
Εικόνα 4.33: Κωδικός 500 - Internal Server Error

Στη συνέχεια ακολουθούν εικόνες από την εφαρμογή (screenshots), όπου παρουσιάζονται τα μηνύματα που ενημερώνουν το χρήστη για επιτυχή ή μη συμβάντα.

Στην εικόνα 4.34 παρουσιάζεται το μήνυμα της επιτυχούς δημιουργίας λογαριασμού, το μήνυμα αυτό εμφανίζεται πάνω από τη φόρμα εισόδου στο λογαριασμό, εκεί όπου γίνεται αυτόματη ανακατεύθυνση του χρήστη μετά από μια επιτυχή δημιουργία λογαριασμού.



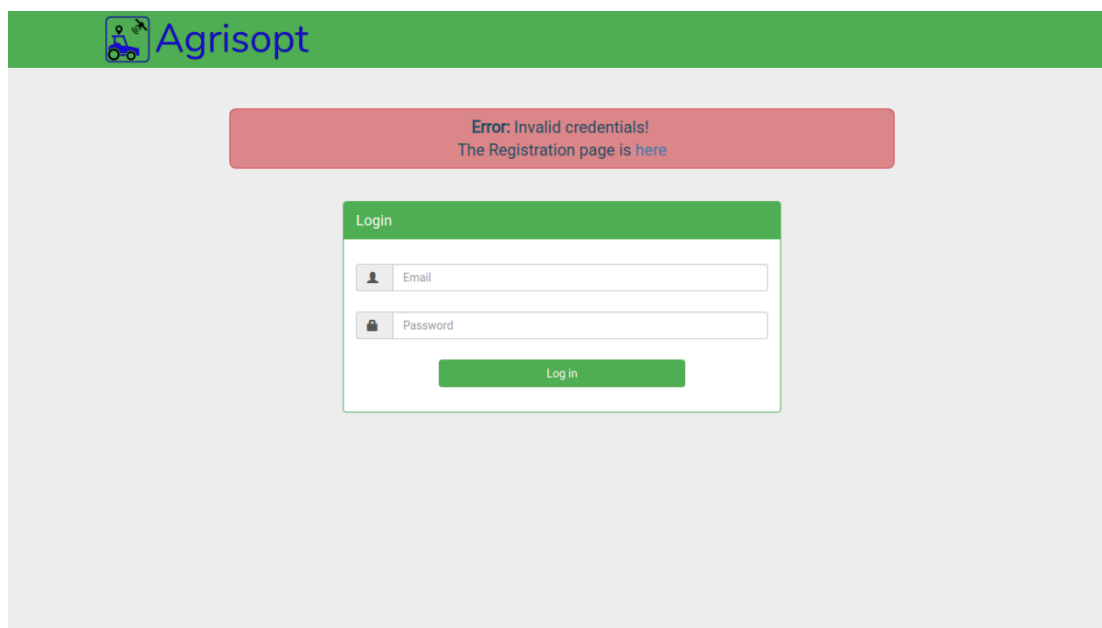
Εικόνα 4.34: Το μήνυμα της επιτυχούς δημιουργίας λογαριασμού και η ανακατεύθυνση στη φόρμα εισόδου



Εικόνα 4.35: Το μήνυμα που εμφανίζεται κατά τη λάθος εισαγωγή στοιχείων

Στην εικόνα 4.35 βλέπουμε το μήνυμα που εμφανίζεται στο χρήστη στην περίπτωση που: 1) εισάγει όνομα χρήστη που δεν έχει τη μορφή ενός email ή 2) εισάγει όνομα χρήστη που αντιστοιχεί σε στοιχεία που υπάρχουν ήδη στη βάση δεδομένων.

Στην εικόνα 4.36 βλέπουμε το μήνυμα που εμφανίζεται όταν κάποιος στη φόρμα εισόδου στο λογαριασμό (Login form) πληκτρολογεί ένα όνομα χρήστη που δεν υπάρχει στη βάση δεδομένων ή ένα υπαρκτό όνομα χρήστη με λάθος κωδικό πρόσβασης.

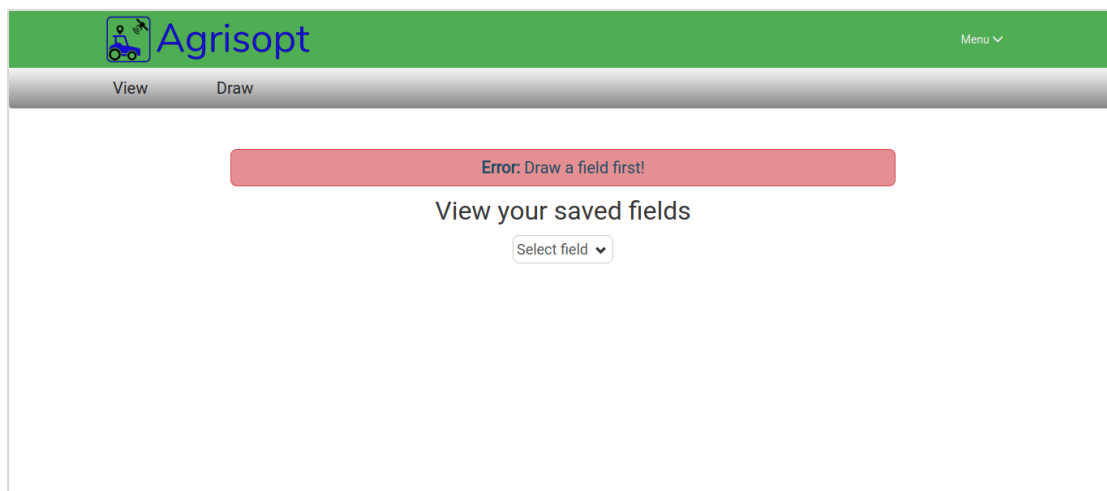


Εικόνα 4.36: Το μήνυμα που εμφανίζεται όταν υπάρχει λανθασμένη εισαγωγή στοιχείων

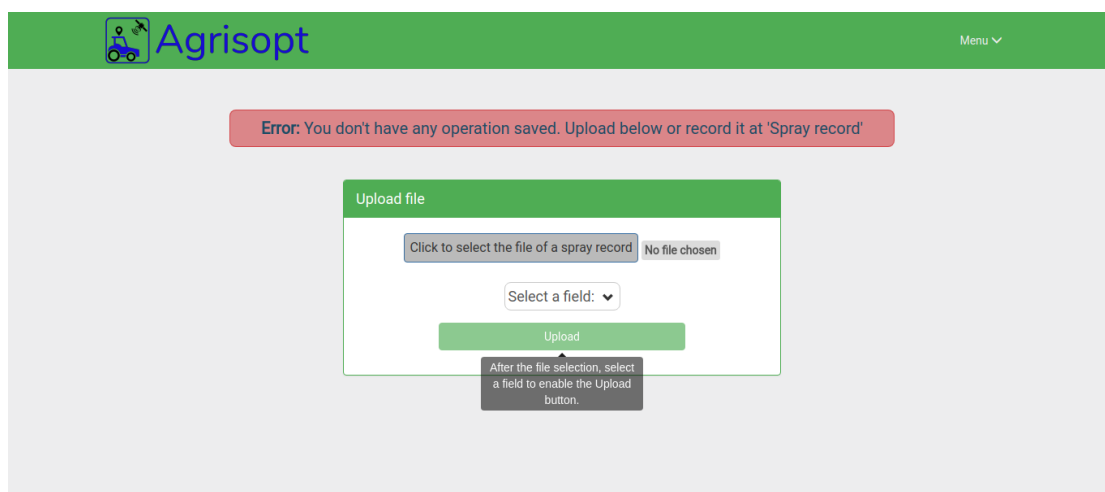
Επιλογή από το Menu: Operations

Στην εικόνα 4.37 βλέπουμε την περίπτωση όπου ένα χρήστης που δεν έχει καταχωρήσει κάποια έκταση στο σύστημα προσπαθήσει να κατευθυνθεί στην επιλογή Operations, τότε το σύστημα θα τον ανακατευθύνει στην επιλογή Fields με το ανάλογο μήνυμα.

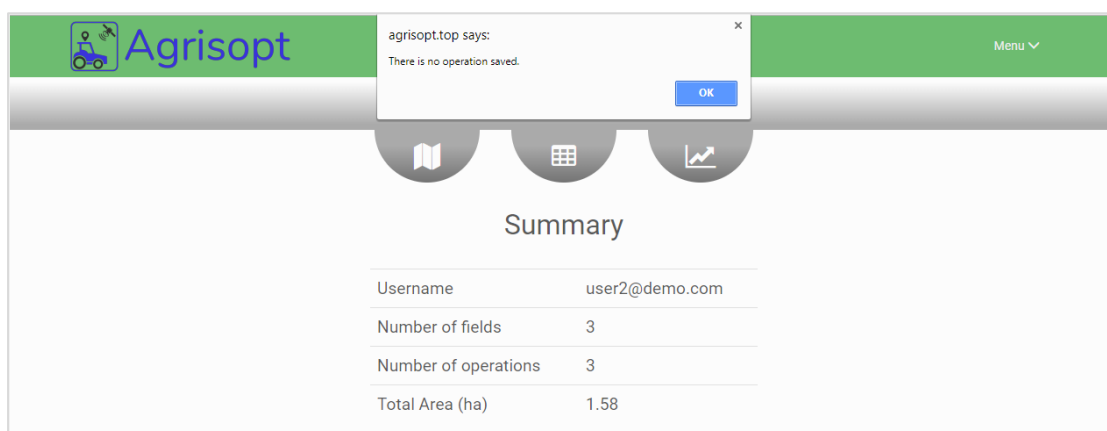
Αντίστοιχα στην εικόνα 4.38 παρουσιάζεται η περίπτωση όπου ένας χρήστης που έχει καταχωρήσει τουλάχιστον μια έκταση, αλλά δεν έχει καταχωρήσει τουλάχιστον μια επέμβαση ψεκασμού, προσπαθήσει να κατευθυνθεί στην επιλογή Operations, τότε το σύστημα θα τον ανακατευθύνει στην επιλογή Upload data με το ανάλογο μήνυμα.



Εικόνα 4.37: Νέος χρήστης χωρίς καταχωρημένη έκταση



Εικόνα 4.38: Νέος χρήστης με καταχωρημένη έκταση, χωρίς όμως να έχει αποθηκεύσει κάποια επέμβαση ψεκασμού

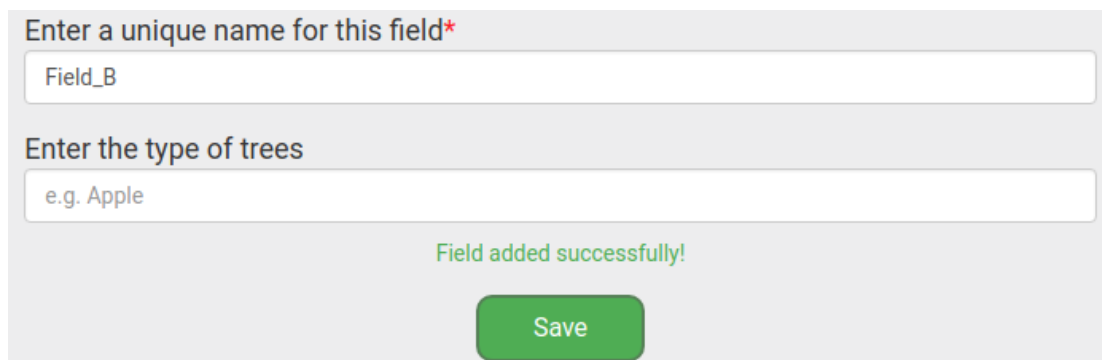


Εικόνα 4.39: Το μήνυμα που εμφανίζεται για ένα αγροτεμάχιο που δεν έχει καμία επέμβαση ψεκασμού αποθηκευμένη

Στην εικόνα 4.39 βλέπουμε το μήνυμα που εμφανίζεται όταν ο χρήστης επιλέξει ένα αγροτεμάχιο στο οποίο δεν αντιστοιχεί καμία καταχωρημένη επέμβαση ψεκασμού.

Επιλογή από το Menu: Fields

Στην εικόνα 4.40 εμφανίζεται το μήνυμα μια επιτυχούς καταχώρησης των στοιχείων ενός αγροτεμαχίου.



Enter a unique name for this field*

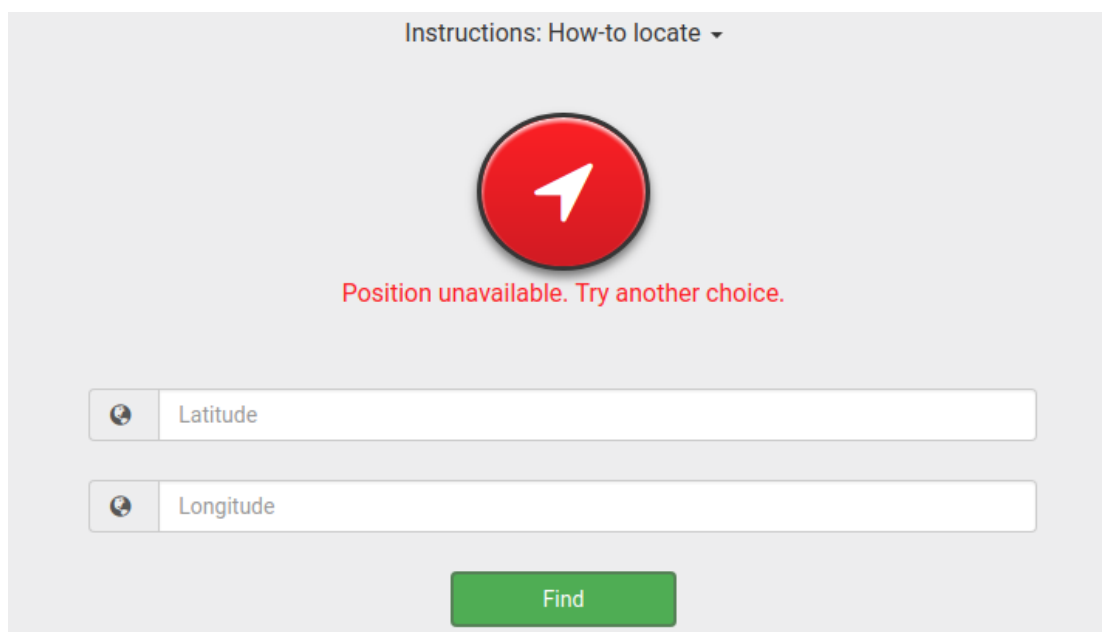
Enter the type of trees

Field added successfully!

Save

Εικόνα 4.40: Το μήνυμα μιας επιτυχούς αποθήκευσης αγροτεμαχίου

Στην εικόνα 4.41 βλέπουμε το μήνυμα που εμφανίζεται όταν η εφαρμογή δεν μπορεί να εντοπίσει τη θέση του χρήστη.



Instructions: How-to locate ▾

Position unavailable. Try another choice.

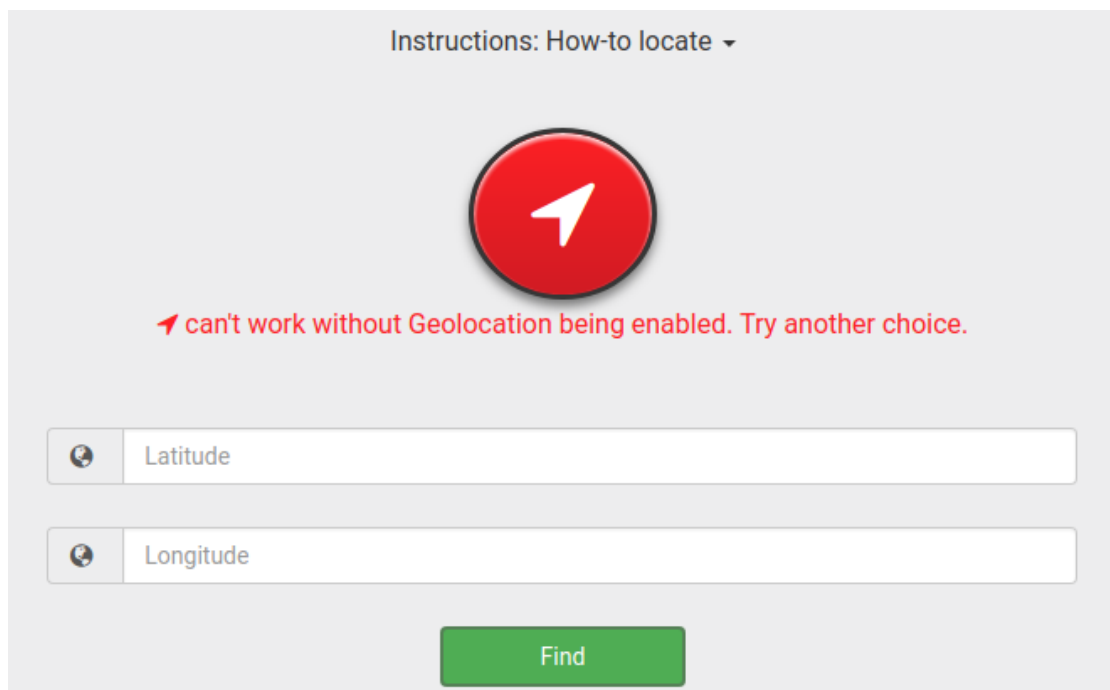
Latitude

Longitude

Find

Εικόνα 4.41: Το μήνυμα για την περίπτωση αδυναμίας εύρεσης θέσης

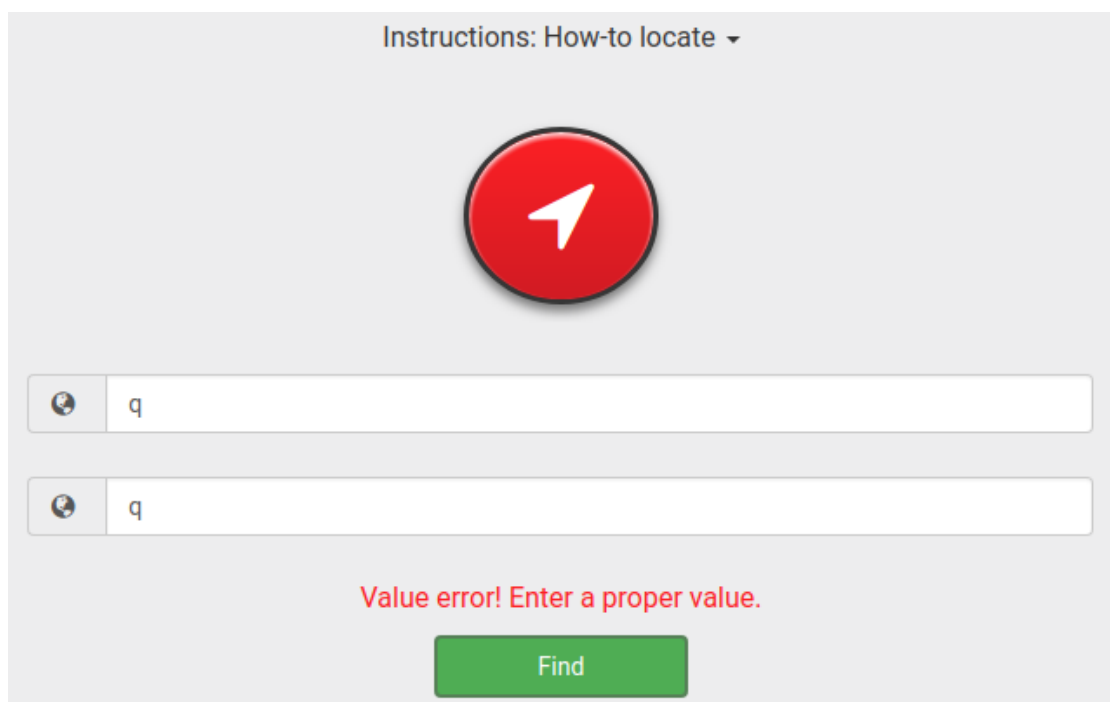
Στην εικόνα 4.42 παρουσιάζεται το μήνυμα που αντιστοιχεί στην περίπτωση που ο χρήστης δεν δώσει τη συγκατάθεσή του για τη χρήση του Geolocation από την εφαρμογή.



The screenshot shows a user interface for finding a location. At the top, it says "Instructions: How-to locate" with a dropdown arrow. In the center is a red circular icon with a white location pin. Below the icon is a red error message: "can't work without Geolocation being enabled. Try another choice." There are two input fields: the first is labeled "Latitude" and the second is labeled "Longitude". At the bottom is a green "Find" button.

Εικόνα 4.42: Το μήνυμα για την περίπτωση που δεν υπάρχει άδεια χρήσης του Geolocation

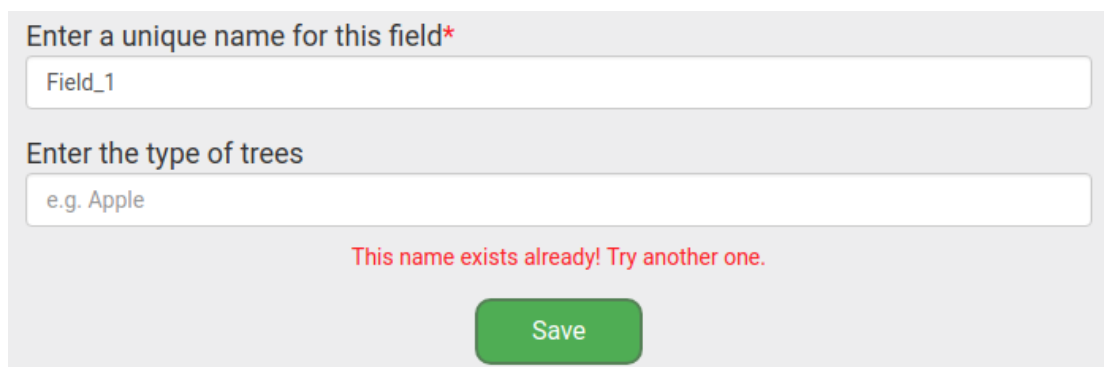
Όταν ένας χρήστης στη φόρμα εισαγωγής συντεταγμένων κατά λάθος εισάγει αντί για αριθμούς γράμματα ή σύμβολα τότε εμφανίζεται το μήνυμα της εικόνας 4.43.



The screenshot shows the same user interface as in Figure 4.42. The "Latitude" and "Longitude" input fields now contain the letter "q". Below the input fields is a red error message: "Value error! Enter a proper value." The "Find" button remains at the bottom.

Εικόνα 4.43: Το μήνυμα για τις λάθος τιμές εισαγωγή συντεταγμένων

Στην εικόνα 4.44 βλέπουμε το μήνυμα που εμφανίζεται όταν ο χρήστης εισάγει ένα όνομα αγροτεμαχίου το οποίο το έχει χρησιμοποιήσει ήδη νωρίτερα για κάποιο άλλο.



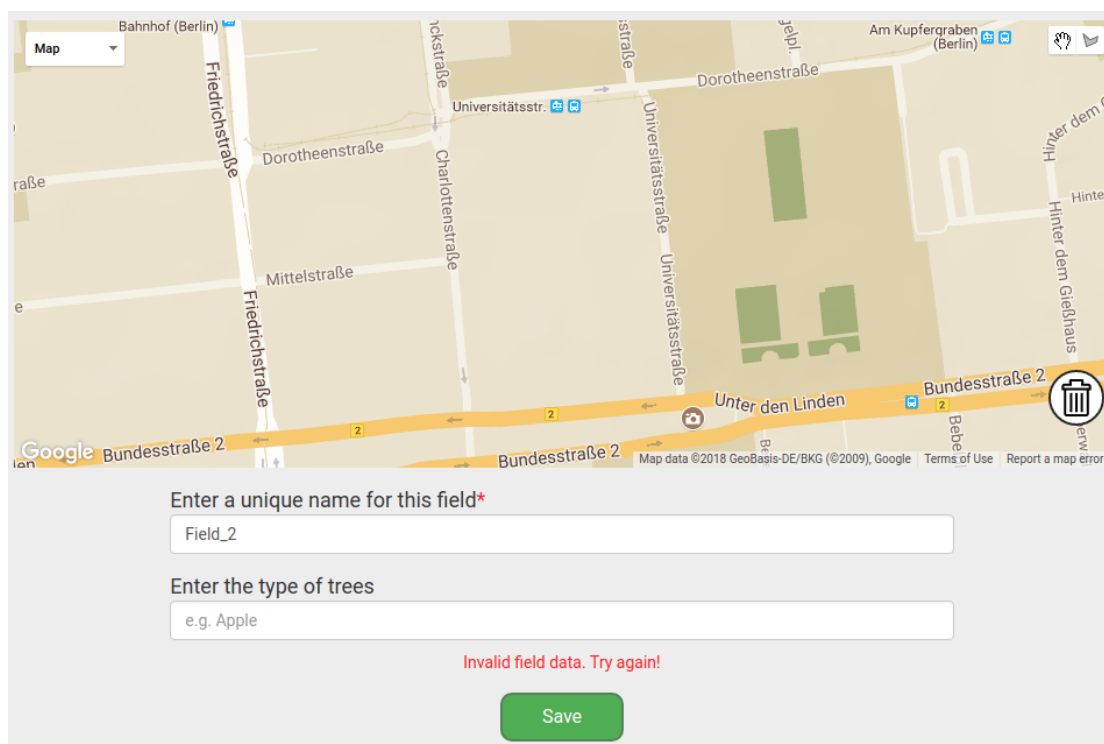
Enter a unique name for this field*

Enter the type of trees

This name exists already! Try another one.

Εικόνα 4.44: Το μήνυμα που ειδοποιεί για την ύπαρξη του ονόματος του αγροτεμαχίου στη βάση δεδομένων

Στην περίπτωση όπου ένας χρήστης δεν ψηφιοποιήσει τα όρια ενός πολυγώνου και προσπαθήσει να το καταχωρήσει ελλιπές τότε εμφανίζεται το μήνυμα της εικόνας 4.45.



Map

Enter a unique name for this field*

Enter the type of trees

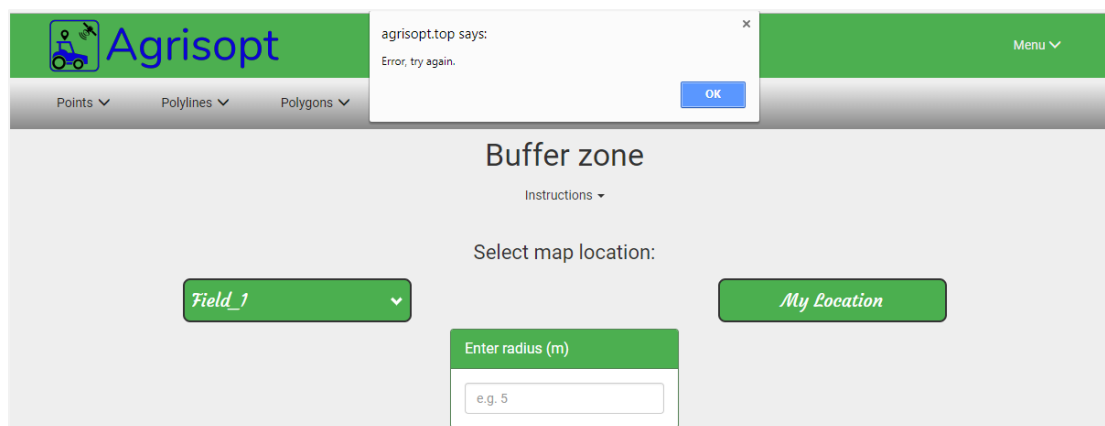
Invalid field data. Try again!

Εικόνα 4.45: Το μήνυμα για την έλλειψη συντεταγμένων των ορίων του αγροτεμαχίου

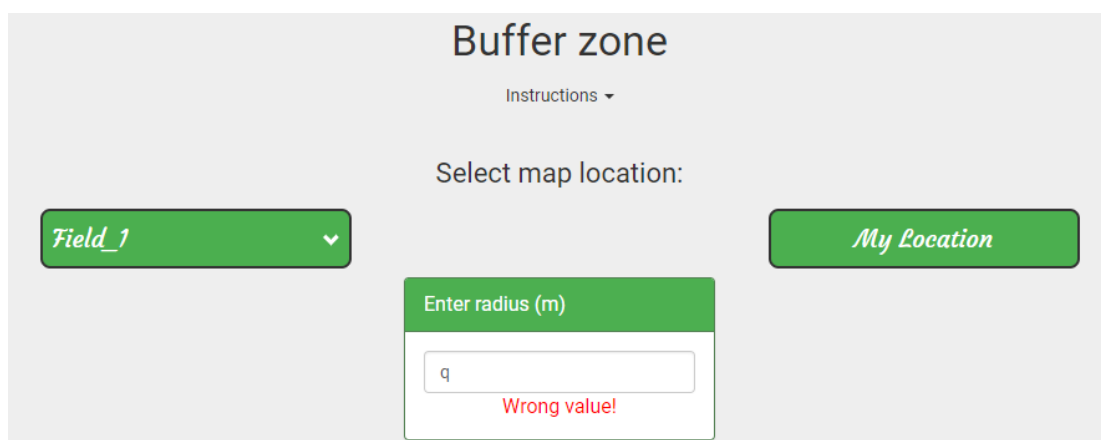
Επιλογή από το Menu: Spatial tools

Στην εικόνα 4.46 παρουσιάζεται το μήνυμα που θα εμφανιστεί στην περίπτωση όπου ο χρήστης έχοντας επιλέξει να χρησιμοποιήσει το εργαλείο “Buffer zone” δεν εισάγει κάποια τιμή στη φόρμα.

Αντίστοιχα στην εικόνα 4.47 εμφανίζεται το μήνυμα που αντιστοιχεί στην περίπτωση όπου ο χρήστης αντί για κάποιο αριθμό πληκτρολογεί κάποιο γράμμα ή σύμβολο.



Εικόνα 4.46: Ο χρήστης δεν εισάγει κάποια τιμή στη φόρμα



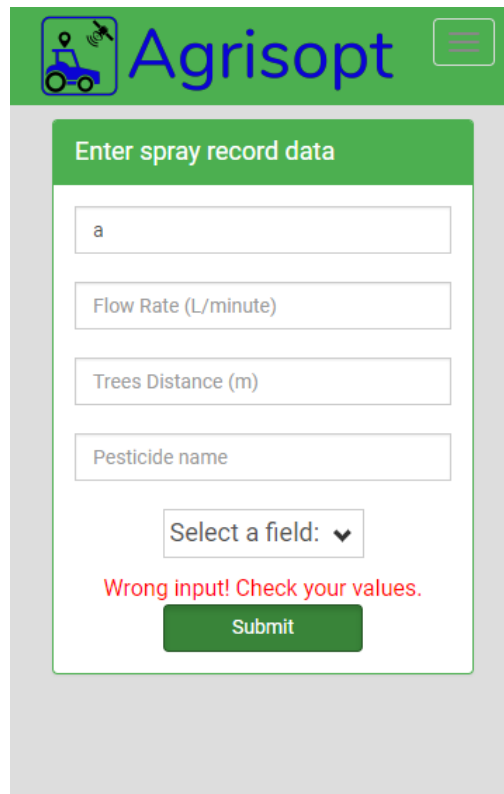
Εικόνα 4.47: Ο χρήστης λανθασμένα εισάγει αντί για αριθμό ένα ή περισσότερα γράμματα

Επιλογή από το Menu: Spray record

Στην εικόνα 4.48 βλέπουμε την περίπτωση όπου ο χρήστης σε τουλάχιστον ένα πεδίο είτε δεν εισάγει τις επιτρεπτές τιμές, είτε δεν εισάγει καθόλου τιμές.

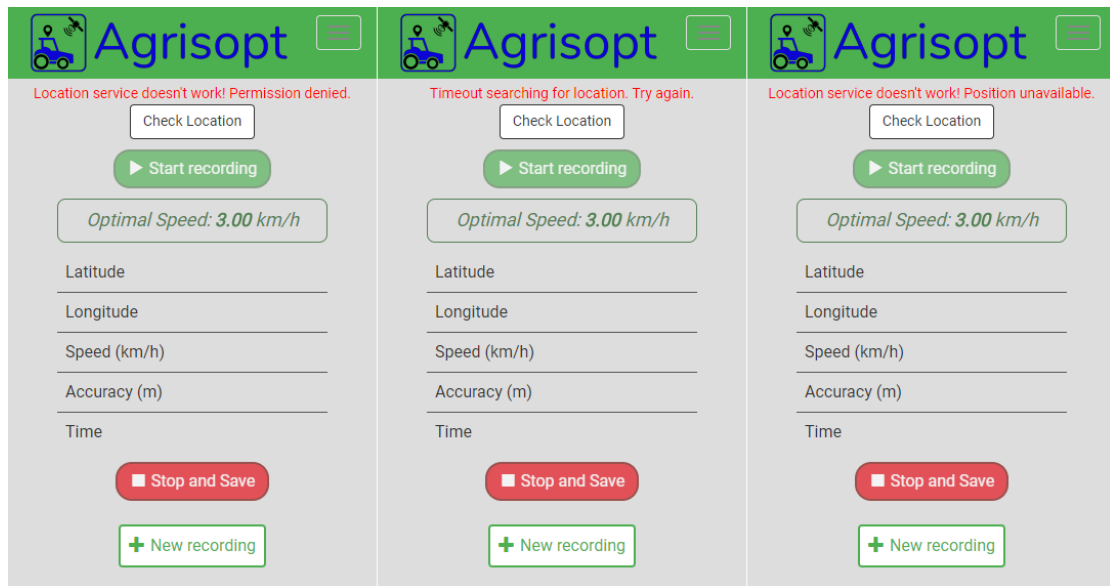
Στην εικόνα 4.49 παρουσιάζονται τρεις περιπτώσεις όπου συμβαίνει σφάλμα κατά τον έλεγχο για τη εύρεση τοποθεσίας. Στην 1^η περίπτωση (από αριστερά) εμφανίζεται το μήνυμα “Location service doesn’t work! Permission denied.”, σε αυτή την

περίπτωση ο χρήστης δεν έδωσε τη συγκατάθεσή του για τη χρήση του Geolocation. Στη 2^η περίπτωση εμφανίζεται το μήνυμα “Timeout searching for location”, αυτό το μήνυμα θα εμφανιστεί όταν η συσκευή του χρήστη δεν δώσει μέσα σε ένα χρονικό διάστημα την τοποθεσία. Σε αυτή την περίπτωση ο χρήστης μπορεί να ξαναδοκιμάσει για να βρεθεί η τοποθεσία του. Στην 3^η περίπτωση εμφανίζεται το μήνυμα “Location service doesn’t work! Position unavailable”, σε αυτή την περίπτωση η εφαρμογή δεν μπορεί να εντοπίσει με επιτυχία τη θέση του χρήστη.



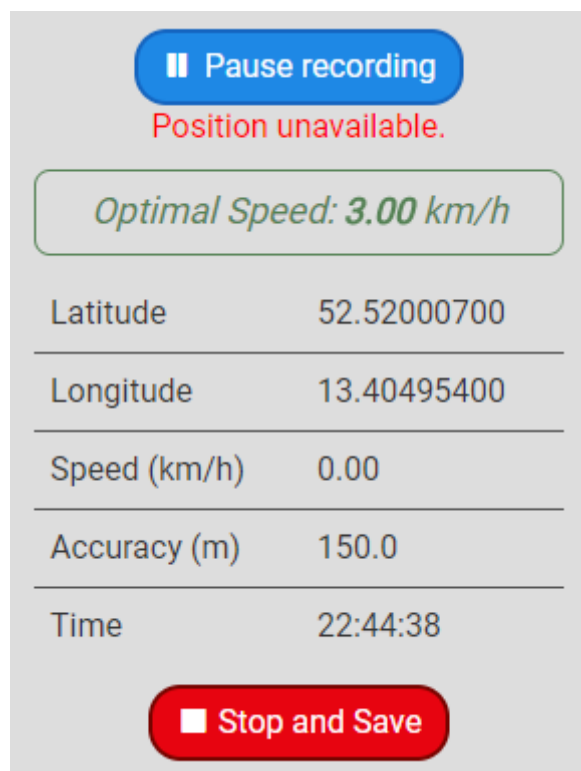
The image shows a mobile application interface for 'Agrisopt'. At the top, there is a green header with the app's logo and name. Below the header is a form titled 'Enter spray record data'. The form contains several input fields: a text field with the value 'a', a field for 'Flow Rate (L/minute)', a field for 'Trees Distance (m)', and a field for 'Pesticide name'. Below these fields is a dropdown menu labeled 'Select a field:'. A red error message 'Wrong input! Check your values.' is displayed above a green 'Submit' button.

Εικόνα 4.48: Το μήνυμα για την περίπτωση της ελλιπούς ή λανθασμένης συμπλήρωσης των πεδίων της φόρμας



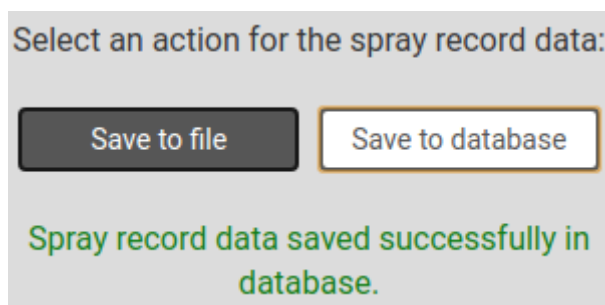
Εικόνα 4.49: Τρία διαφορετικά μηνύματα που επεξηγούν το λόγο της αδυναμίας εύρεσης τοποθεσίας

Στην εικόνα 4.50 εμφανίζεται το μήνυμα που θα δει ο χρήστης αν κατά τη διάρκεια της καταγραφής των χωρικών δεδομένων ενός ψεκασμού υπάρξει αδυναμία εύρεσης της θέσης.



Εικόνα 4.50: Αδυναμία εύρεσης της θέσης κατά τη διάρκεια του ψεκασμού

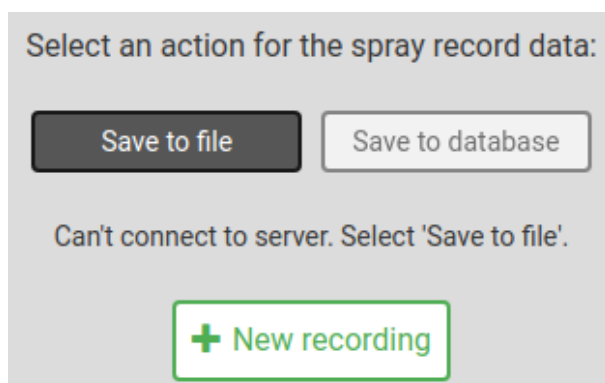
Στην εικόνα 4.51 παρουσιάζεται το μήνυμα της επιτυχούς καταχώρησης των δεδομένων της καταγραφής ενός ψεκασμού.



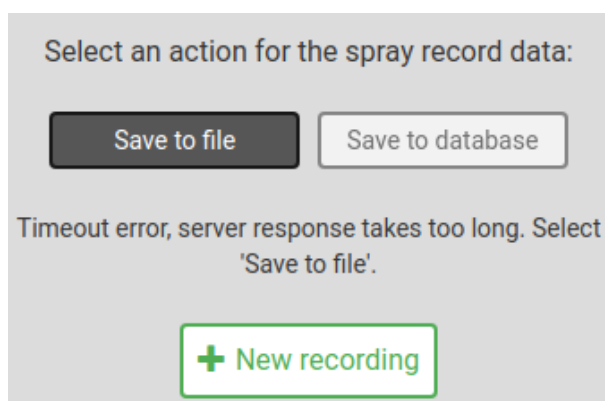
Εικόνα 4.51: Το μήνυμα της επιτυχούς αποθήκευσης ενός ψεκασμού

Στην εικόνα 4.52 εμφανίζεται το μήνυμα που ειδοποιεί το χρήστη για την αδυναμία σύνδεσης με το διακομιστή. Σε αυτή την περίπτωση ο χρήστης θα πρέπει να αποθηκεύσει τα δεδομένα σε ένα αρχείο.

Αντίστοιχα, στην εικόνα 4.53 εμφανίζεται το μήνυμα που αντιστοιχεί στην περίπτωση μιας κακής ποιότητας σύνδεση με το διαδίκτυο, τότε ο χρήστης θα πρέπει να επιλέξει την αποθήκευση σε αρχείο.

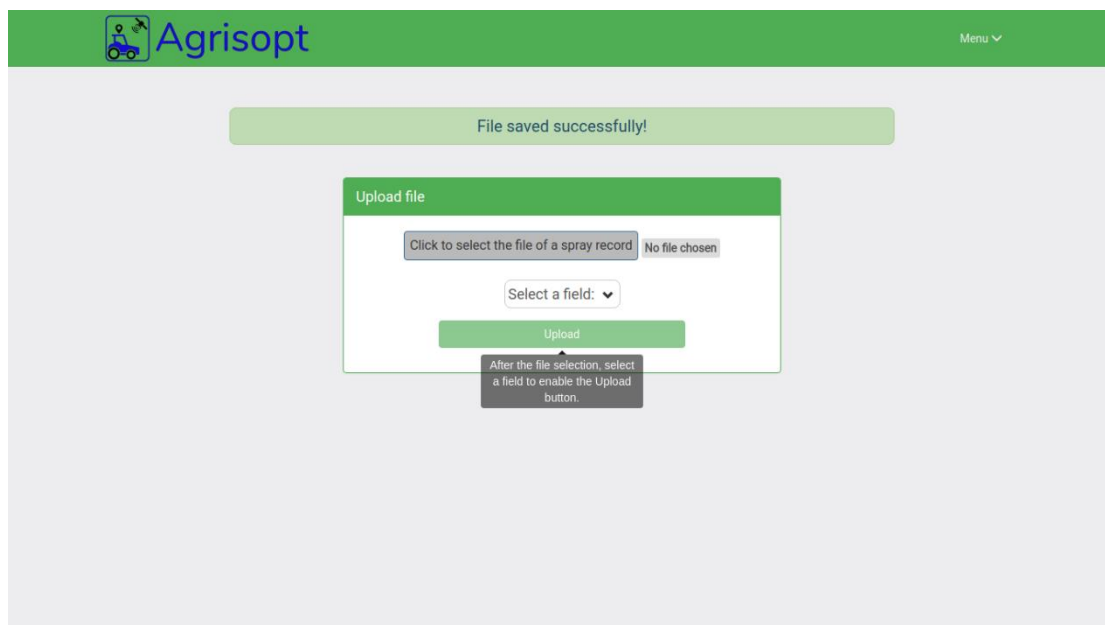


Εικόνα 4.52: Αδυναμία σύνδεσης με το διακομιστή



Εικόνα 4.53: Κακής ποιότητας σύνδεση στο διαδίκτυο

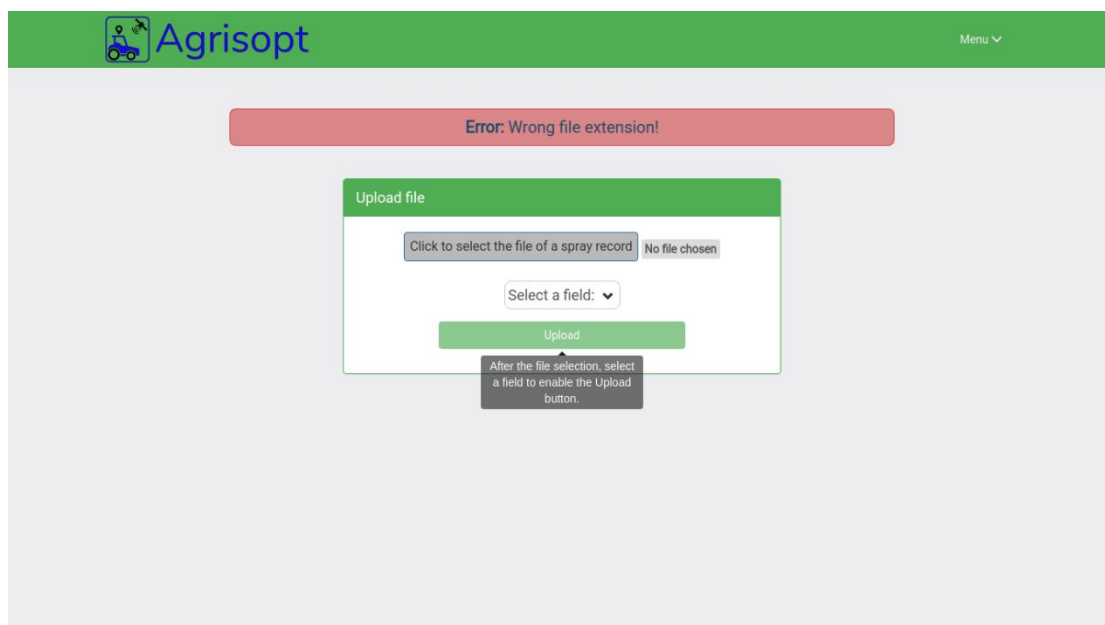
Επιλογή από το Menu: Upload data



Εικόνα 4.54: Το μήνυμα της επιτυχούς μεταφοράς του αρχείου

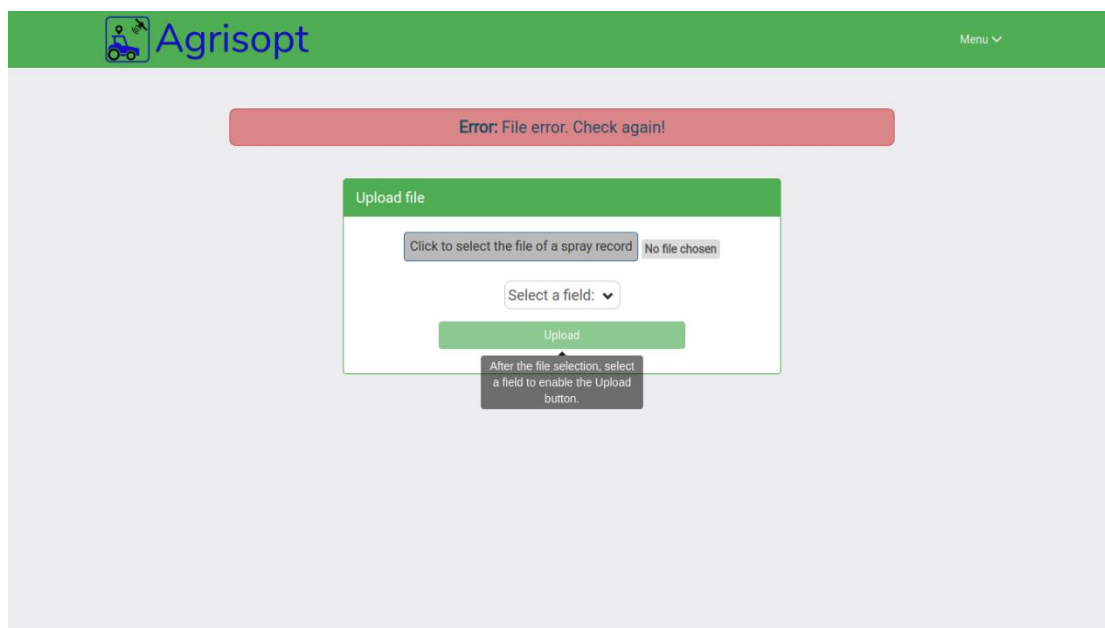
Στην εικόνα 4.54 βλέπουμε τη φόρμα μέσω της οποίας ο χρήστης μπορεί να επιλέξει το αρχείο που θα ανεβάσει στο διακομιστή.

Στην εικόνα 4.55 εμφανίζεται το μήνυμα που βλέπει ο χρήστης μετά από μια επιτυχή μεταφορά αρχείου.



Εικόνα 4.55: Το μήνυμα λάθους τύπου επέκτασης αρχείου

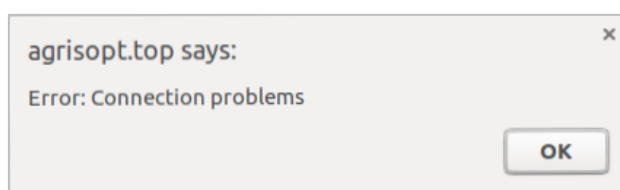
Στην εικόνα 4.56 εμφανίζεται το μήνυμα που παρουσιάζεται στο χρήστη όταν επιλέγει ένα αρχείο με λάθος τύπο επέκτασης.



Εικόνα 4.56: Μήνυμα λάθους μορφής αρχείου

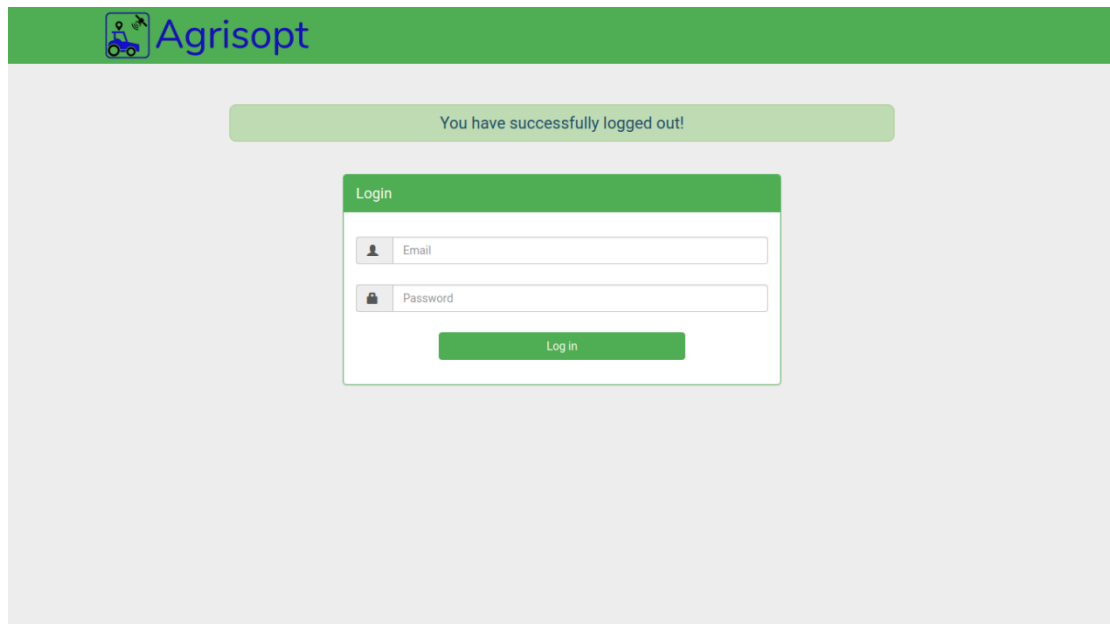
Στην εικόνα 4.56 εμφανίζεται το μήνυμα που βλέπει ο χρήστης αν επιλέξει να ανεβάσει ένα αρχείο που έχει λάθος μορφή.

Στην εικόνα 4.57 εμφανίζεται το γενικό μήνυμα που θα δει ο χρήστης όταν κατά την πραγματοποίηση ενός οποιοδήποτε AJAX request της εφαρμογής υπάρξουν προβλήματα με τη σύνδεση στο διαδίκτυο.



Εικόνα 4.57: Προβλήματα με της σύνδεση στο διαδίκτυο

Στην εικόνα 4.58 βλέπουμε το μήνυμα που εμφανίζεται σε ένα χρήστη αφού αποσυνδεθεί επιτυχώς από τον λογαριασμό του.



Εικόνα 4.58: Το μήνυμα που εμφανίζεται στο χρήστη μετά την έξοδό του από το λογαριασμό

4.2.2 Ευχρηστία

Η ευχρηστία αναλύεται σε τέσσερις υποκατηγορίες οι οποίες είναι οι εξής:

1. Χρώμα.
2. Γραμματοσειρές.
3. Μηνύματα ενημέρωσης ή βοήθειας του χρήστη.
4. Responsive design.

Χρώμα

Ως κύρια χρώματα για την εφαρμογή επιλέχθηκαν τα ψυχρά χρώματα (πράσινο κυρίως και δευτερευόντως μπλε), επειδή αυτού του τύπου τα χρώματα αυξάνουν την αίσθηση ηρεμίας. Για τα μηνύματα που ειδοποιούν το χρήστη για σφάλματα επιλέχθηκε το κόκκινο χρώμα το οποίο ανήκει στα θερμά χρώματα και προκαλεί ερεθισμό. Από τα ουδέτερα χρώματα, το μαύρο και το σκούρο γκρι επιλέχθηκαν για το κείμενο, ενώ το λευκό και το ανοιχτό γκρι επιλέχθηκαν για φόντο του κειμένου.

Γραμματοσειρές

Η κύρια γραμματοσειρά της εφαρμογής είναι η **Roboto**, είναι μια γραμματοσειρά που αναπτύχθηκε από την Google για το λειτουργικό της σύστημα Android. Διατίθεται με άδεια Apache και σχεδιάστηκε για την υποστήριξη οθονών υψηλής ανάλυσης.

View your saved fields

Εικόνα 4.59: Ένα δείγμα της γραμματοσειρά Roboto από την εφαρμογή

Το logo της εφαρμογής χρησιμοποιεί τη γραμματοσειρά **Nunito**.



Εικόνα 4.60: Η γραμματοσειρά Nunito

Το μήνυμα στο κέντρο της αρχικής σελίδας χρησιμοποιεί τη γραμματοσειρά **Michroma**.



Εικόνα 4.61: Η γραμματοσειρά Michroma

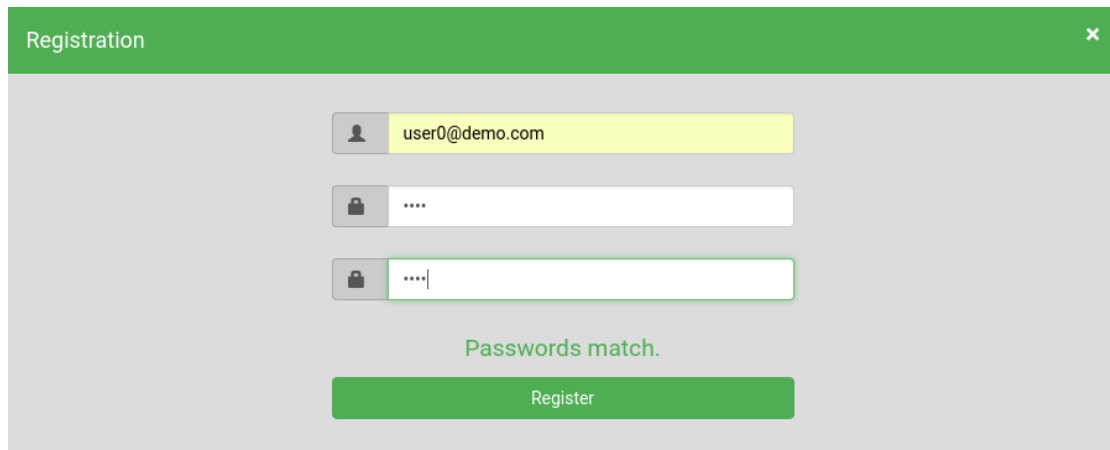
Στη επιλογή “Spatial tools” τα πεδία My Field και My Location χρησιμοποιούν τη γραμματοσειρά **Courgette**.



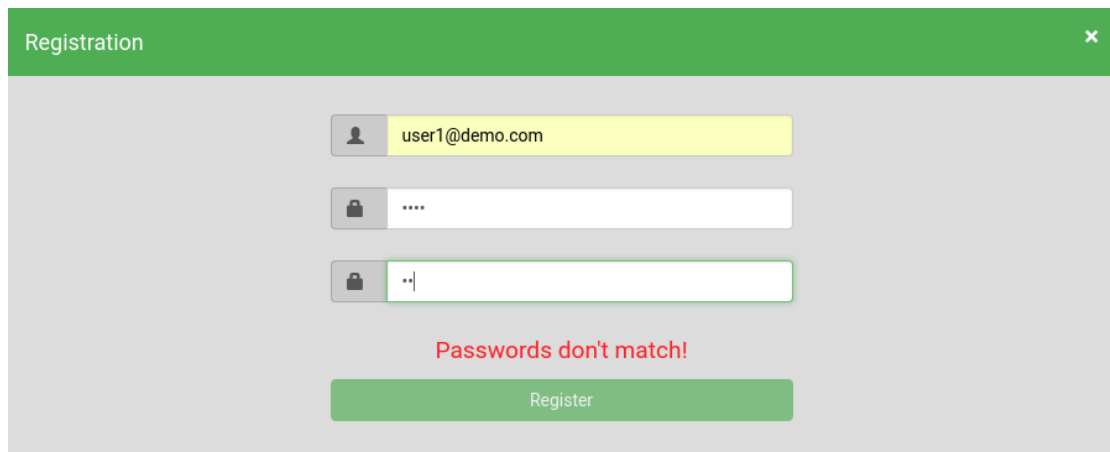
Εικόνα 4.62: Η γραμματοσειρά Courgette

Μηνύματα ενημέρωσης ή βοήθειας του χρήστη

Στις εικόνες 4.63 και 4.64 παρουσιάζονται τα μηνύματα που ενημερώνουν το χρήστη για το αν οι κωδικοί που πληκτρολόγησε στη φόρμα εγγραφής ταιριάζουν μεταξύ τους. Στην περίπτωση που δεν ταιριάζουν τότε το κουμπί για την καταχώρηση της εγγραφής παραμένει ανενεργό όπως φαίνεται στην εικόνα 4.64.

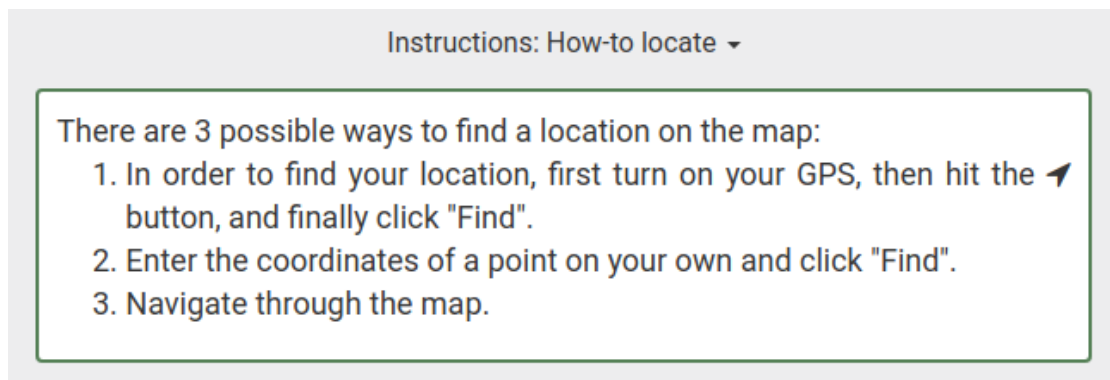


Εικόνα 4.63: Οι κωδικοί ταιριάζουν μεταξύ τους



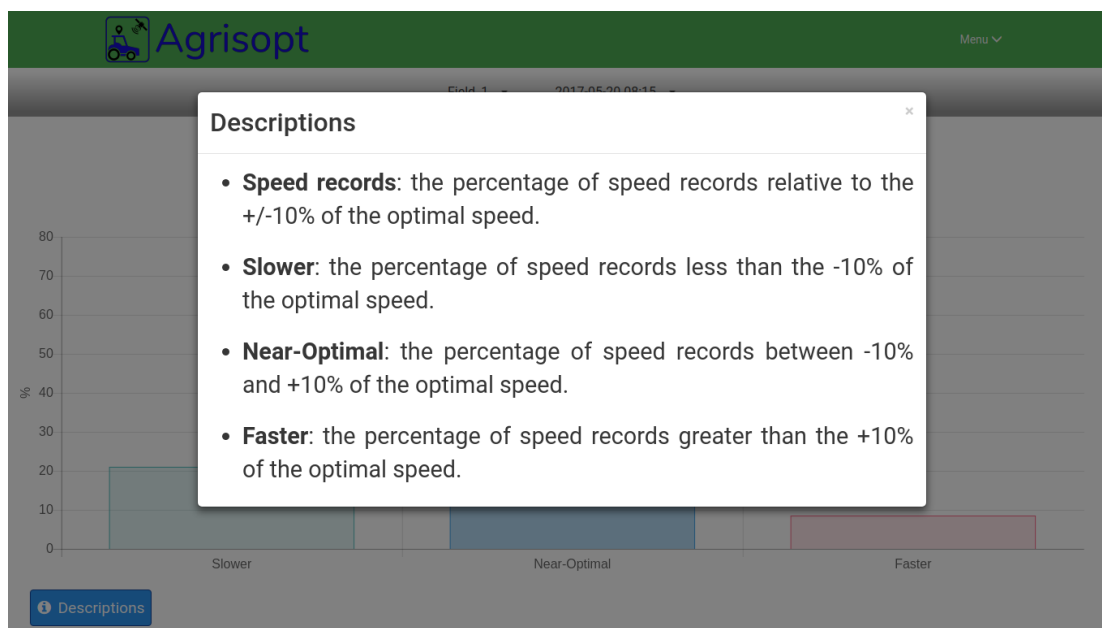
Εικόνα 4.64: Οι κωδικοί δεν ταιριάζουν μεταξύ τους

Στην εικόνα 4.65 εμφανίζονται οι οδηγίες που μπορεί να δει ο χρήστης στην επιλογή Fields: Draw, σχετικά με τους τρόπους με τους οποίους μπορεί να γίνει η εύρεση της τοποθεσίας του αγροτεμαχίου προς ψηφιοποίηση.



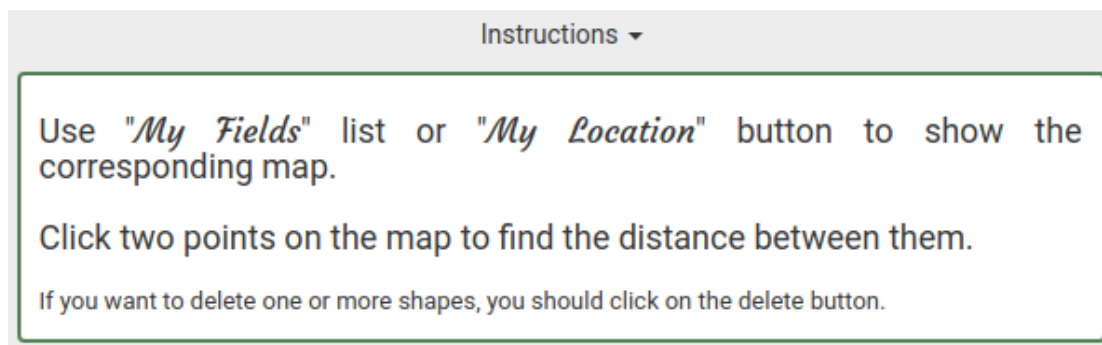
Εικόνα 4.65: Οδηγίες για την εύρεση της τοποθεσίας του αγροτεμαχίου

Στην εικόνα 4.66 εμφανίζονται οι περιγραφές των στοιχείων που αποτελούν το γράφημα της ταχύτητας ενός ψεκασμού στην επιλογή Operations.



Εικόνα 4.66: Περιγραφές των στοιχείων του γραφήματος της ταχύτητας

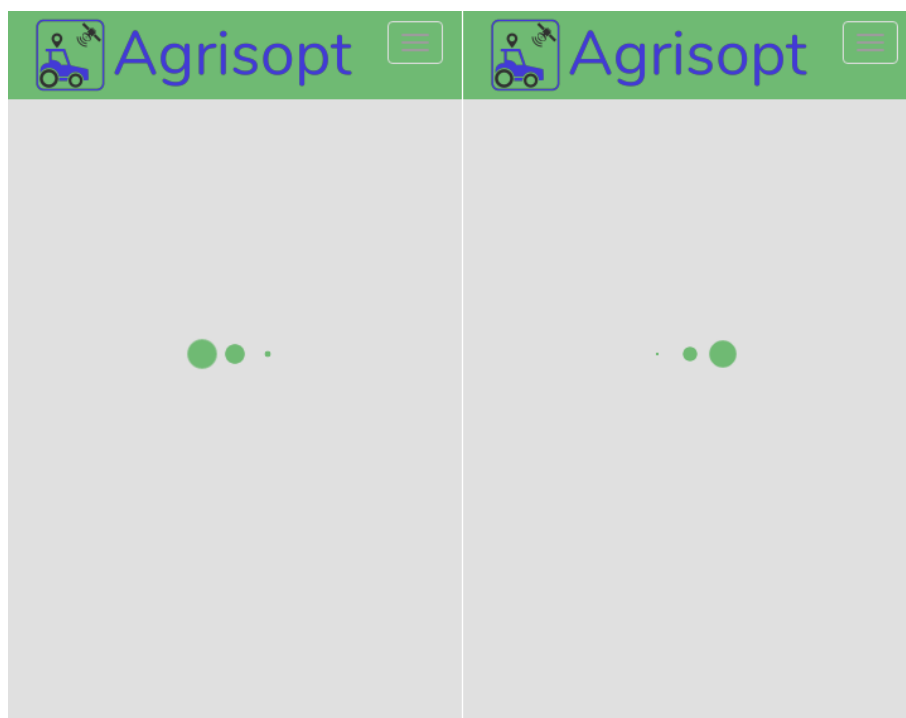
Στην εικόνα 4.67 παρουσιάζονται οι οδηγίες που εμφανίζονται στην επιλογή Spatial tools. Για κάθε μια από τις επτά συνολικά επιλογές εργαλείων χωρικών υπολογισμών υπάρχουν και ξεχωριστές οδηγίες. Στην εικόνα 4.67 φαίνονται οι οδηγίες που αντιστοιχούν στο εργαλείο "Distance between two points".



Εικόνα 4.67: Η οδηγίες για την επιλογή "Distance between two points"

Στην εικόνα 4.68 παρουσιάζεται η εικόνα που εμφανίζεται πάνω στην εφαρμογή όταν πραγματοποιείται ένα AJAX request. Ουσιαστικά πρόκειται για ένα κινούμενο σχέδιο με τις τρεις κύκλους των οποίων αυξομειώνεται η διάμετρος. Σκοπός του είναι να δείξει στο χρήστη ότι εκείνη τη στιγμή η εφαρμογή είναι απασχολημένη και δεν μπορεί να αλληλοεπιδράσει άμεσα με αυτή. Το σχέδιο ξεκινάει να εμφανίζεται όταν αποστέλλεται ένα AJAX request και διαρκεί έως ότου να έρθει μια απάντηση από το διακομιστή. Η συνήθης διάρκεια εμφάνισης του για μια γρήγορη σύνδεση σε δίκτυο

κινητής 3^{ης} γενιάς (3G) είναι ενδεικτικά στα 450-600 ms (ανάλογα και με τη επιλογή από το μενού).



Εικόνα 4.68: Οι τρεις κύκλοι διαρκώς αυξομειούμενης διαμέτρου

Στην εικόνα 4.69 παρουσιάζεται το μήνυμα που βλέπει ο χρήστης όταν έχει συμπληρώσει όλα τα πεδία της φόρμας. Είναι μια παλλόμενη ειδοποίηση που τον προειδοποιεί να ελέγξει ότι το GPS της συσκευής είναι ενεργό. Όπως φαίνεται έχουν επιλεγεί θερμά χρώματα για να διεγείρουν την προσοχή του χρήστη.

Enter spray record data

100

50

10

Insignia 20WG

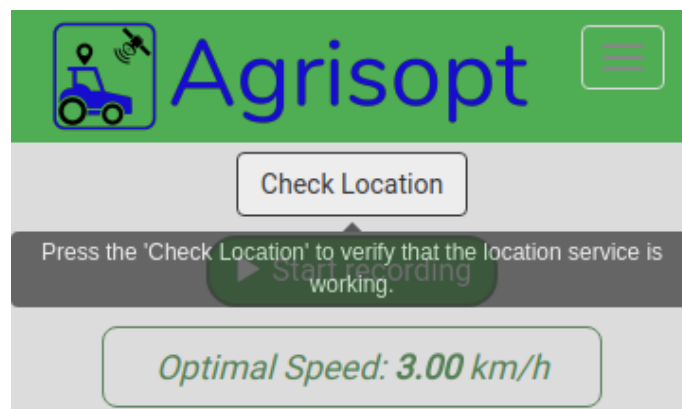
Field_1

Before submitting, make sure that the GPS is turned on!

Submit

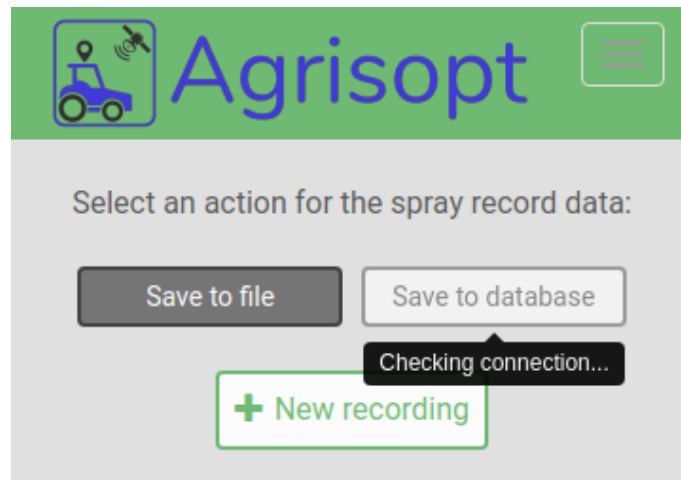
Εικόνα 4.69: Το παλλόμενο μήνυμα-προειδοποίηση για τον έλεγχο της κατάστασης του GPS της συσκευής

Στην εικόνα 4.70 φαίνεται το μήνυμα για το σκοπό της ύπαρξης του κουμπιού “Check Location”.



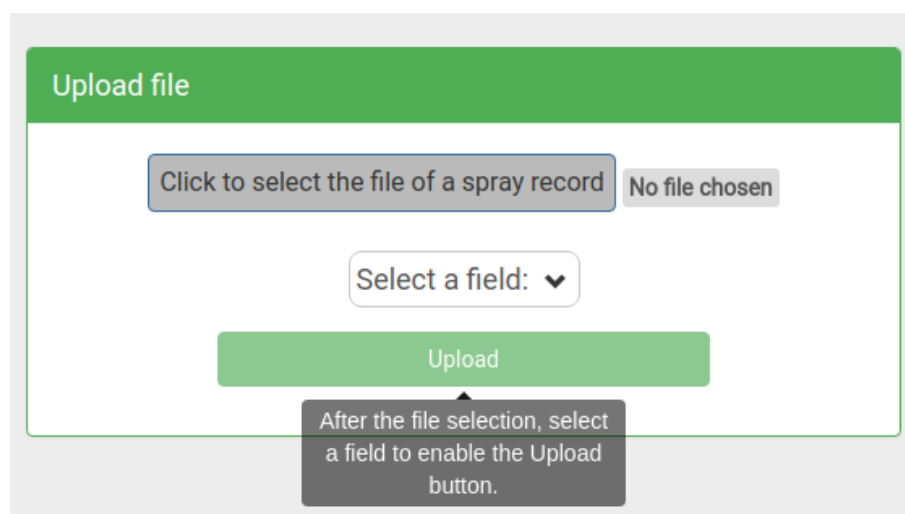
Εικόνα 4.70: Το μήνυμα για το κουμπί “Check Location”

Στην εικόνα 4.71 παρουσιάζεται το μήνυμα που εμφανίζεται όταν γίνεται έλεγχος για την ύπαρξη ενεργούς σύνδεσης με το διακομιστή για την αποθήκευση των δεδομένων της καταγραφής ενός ψεκασμού.



Εικόνα 4.71: Έλεγχος της σύνδεσης με το διακομιστή

Στην εικόνα 4.72 φαίνεται το μήνυμα που βλέπει ο χρήστης όταν κατευθυνθεί στην επιλογή Upload data. Το μήνυμα τον ενημερώνει ότι για να γίνει ενεργό το κουμπί “Upload” θα πρέπει μετά την επιλογή του αρχείου να επιλέξει και το αγρόκτημα στο οποίο αντιστοιχούν τα δεδομένα του ψεκασμού.



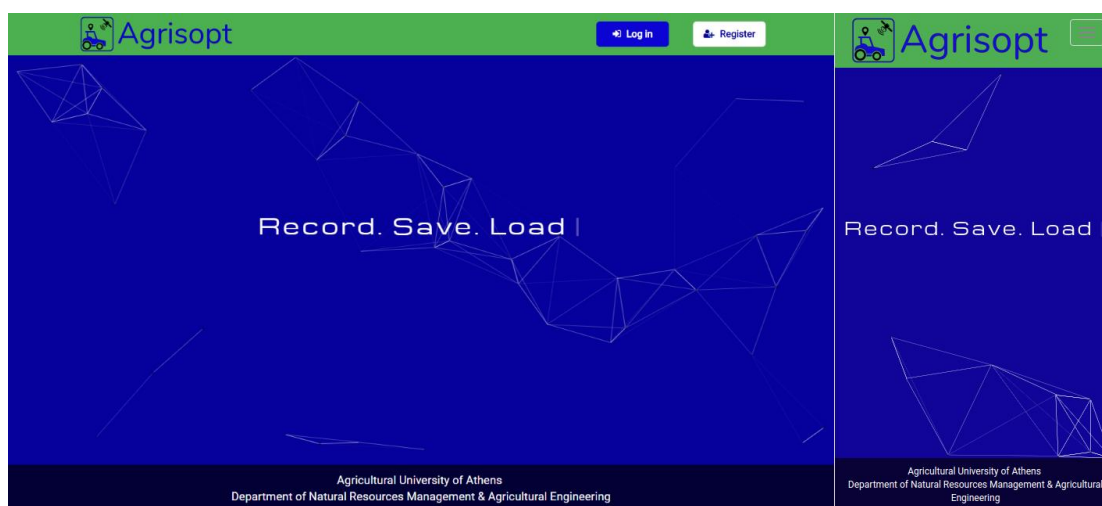
Εικόνα 4.72: Το μήνυμα για τον τρόπο ενεργοποίησης του κουμπιού “Upload”

Responsive design

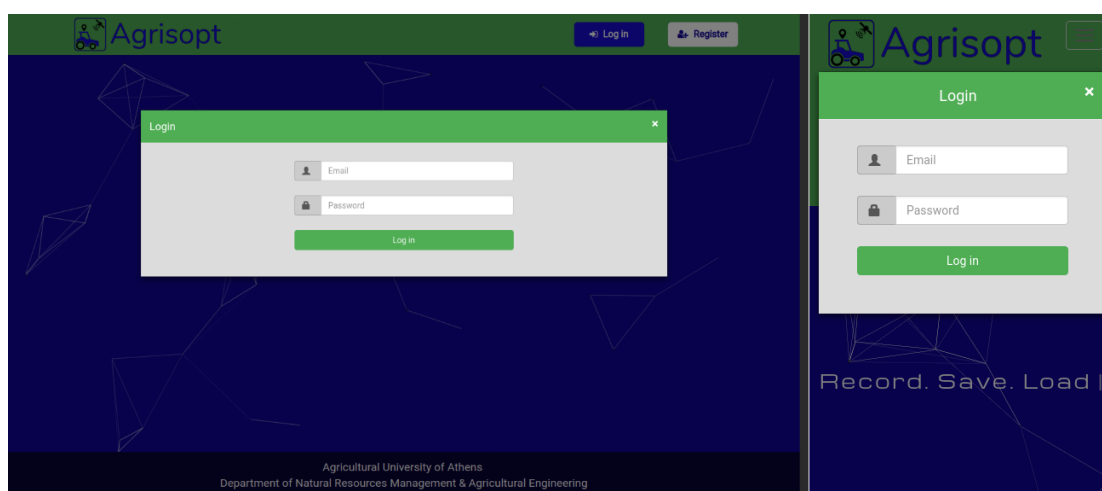
Όταν το 2007 παρουσιάστηκε για 1^η φορά το iPhone ήταν ένα σημείο καμπής για το web design. Μέχρι τότε, οι ιστοσελίδες έπρεπε να δουλεύουν σε οθόνες desktop και laptop υπολογιστών, οι οποίες είχαν μια σχετικά μικρή διακύμανση σε διαστάσεις. Με τη είσοδο, όμως των έξυπνων κινητών θα έπρεπε να υπάρξουν αλλαγές στο σχεδιασμό των ιστοσελίδων. Αρχικά οι σχεδιαστές επέλεξαν να έχουν ένα σχεδιασμό για τους desktop και laptop υπολογιστές και ένα για τα iPhone. Επειδή όμως στη

συνέχεια υπήρξε είσοδος κινητών τηλεφώνων και tablets με πολλές διαφορετικές διαστάσεις φάνηκε ότι η μέθοδος που ίσχυε μέχρι τότε δεν ήταν πλέον λειτουργική και θα έπρεπε να αντικατασταθεί. Η λύση για το σχεδιασμό ιστοσελίδων που θα δουλεύουν σε οθόνες όλων των μεγεθών ήταν το Responsive Web Design (RWD). Ουσιαστικά το RWD είναι μια μέθοδος ευέλικτου σχεδιασμού ιστοσελίδων, οι οποίες είναι ικανές να εντοπίσουν τις διαστάσεις της εκάστοτε οθόνης στην οποία προβάλλονται και να ρυθμίσουν το σχεδιασμό τους ανάλογα.

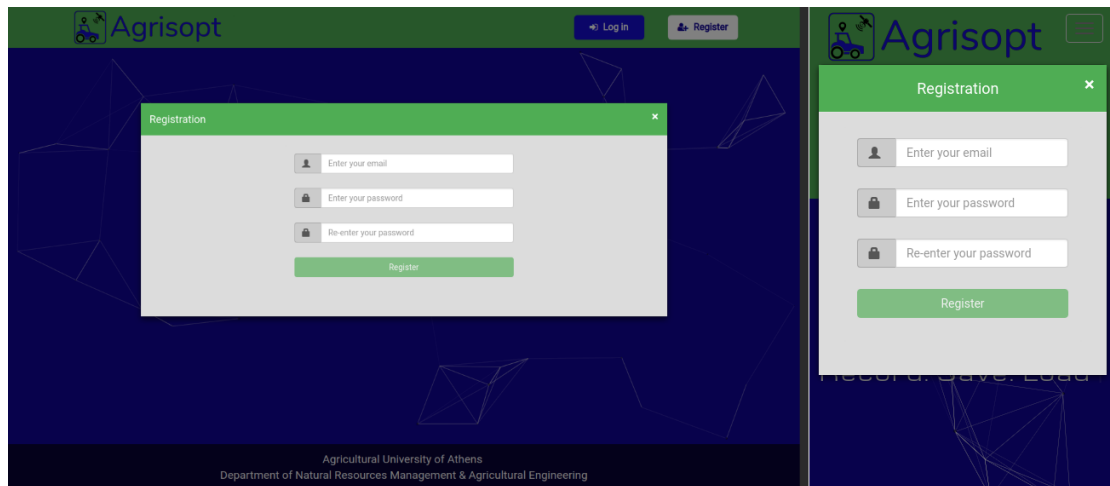
Κατά το σχεδιασμό της εφαρμογής έγινε χρήση τεχνικών του RWD με σκοπό η εφαρμογή να είναι όσο το δυνατόν το ίδιο λειτουργική σε οθόνες: desktop και laptop υπολογιστών, tablets και κινητών τηλεφώνων. Στη συνέχεια ακολουθούν εικόνες από την εφαρμογή (screenshots), όπου στα αριστερά φαίνεται η εμφάνιση της εφαρμογής σε μια οθόνη σταθερού υπολογιστή και στα δεξιά σε μια οθόνη κινητού τηλεφώνου.



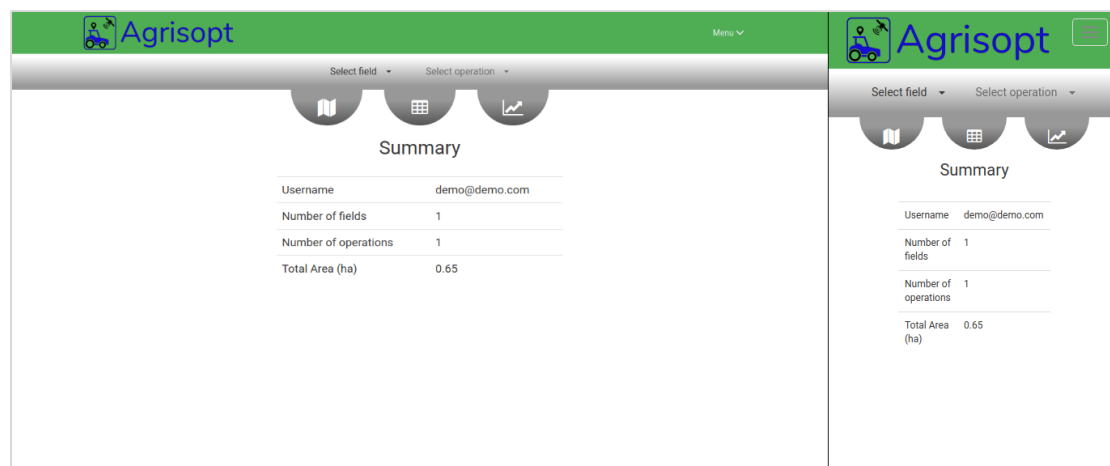
Εικόνα 4.73: Η αρχική σελίδα



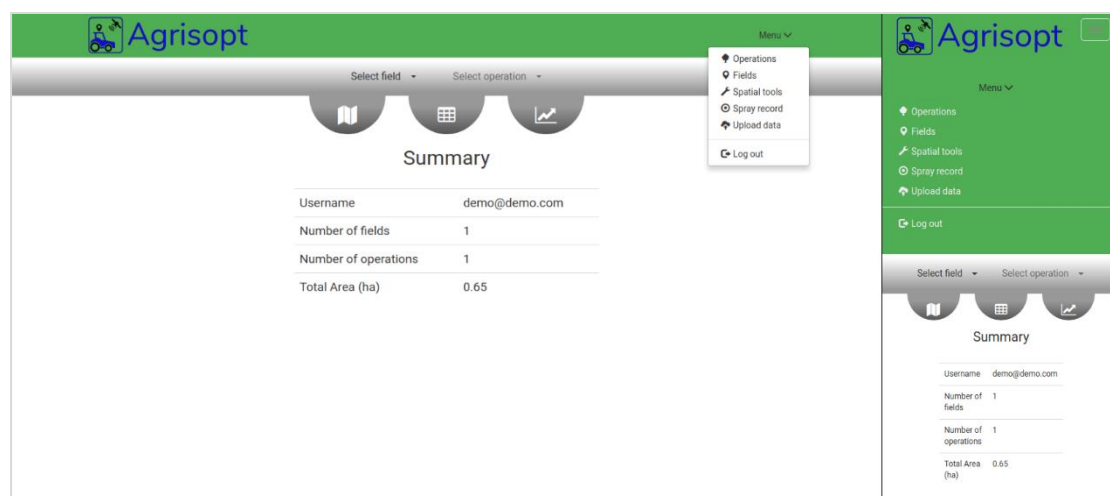
Εικόνα 4.74: Η φόρμα εισόδου στο λογαριασμό του χρήστη



Εικόνα 4.75: Η φόρμα εγγραφής χρήστη



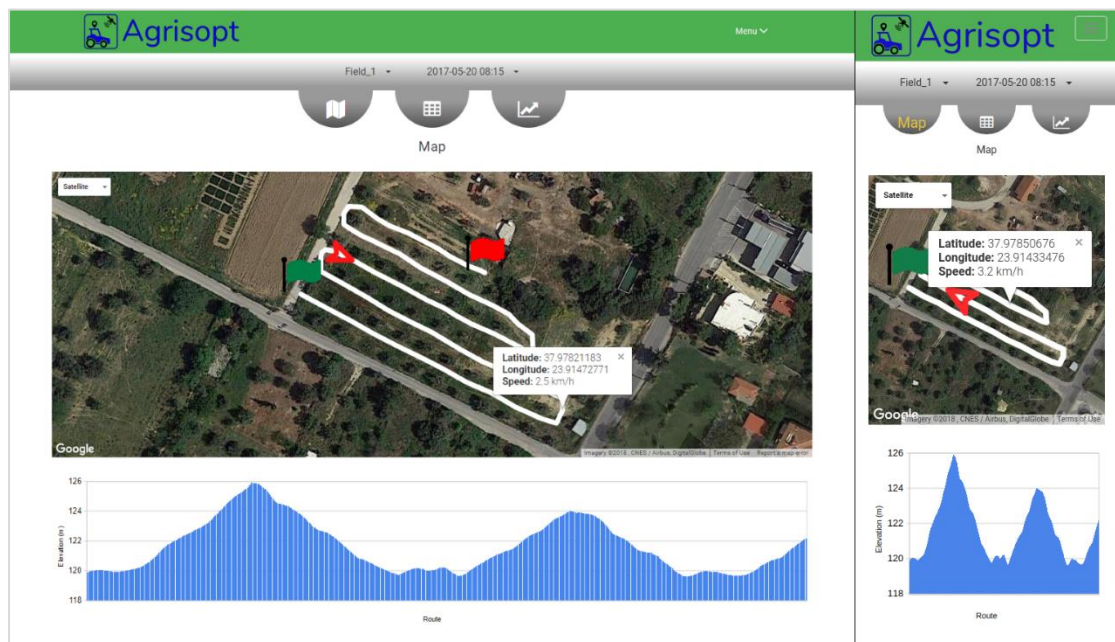
Εικόνα 4.76: Η σελίδα παρουσίασης των επεμβάσεων ψεκασμού (Operations)



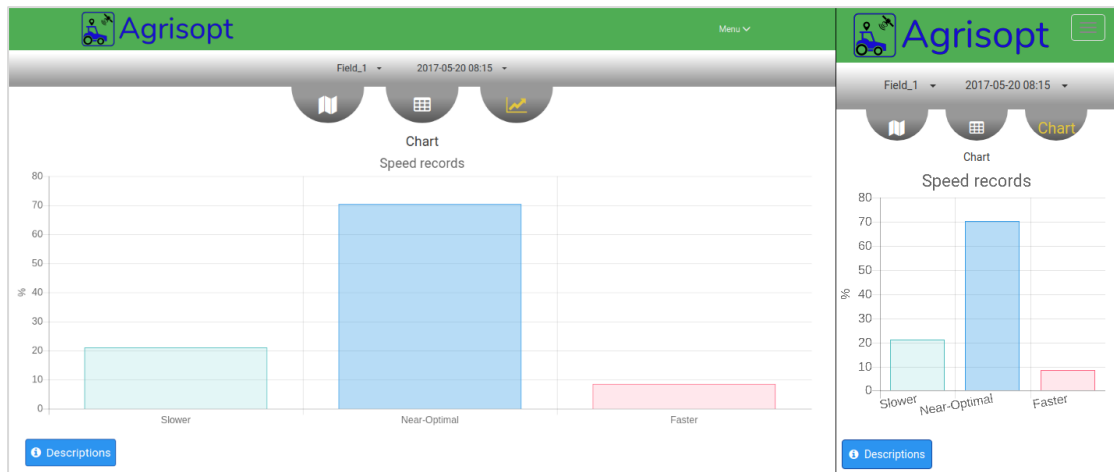
Εικόνα 4.77: Το μενού της εφαρμογής

| Data | |
|--------------------------|---------------------------|
| Distance travelled (km) | 0.48 |
| Optimal speed (km/h) | 3.00 |
| Average speed (km/h) | 2.93 |
| Spray volume (L/(ha/10)) | 100 |
| Flow rate (L/min) | 50.0 |
| Duration | 09 minutes and 29 seconds |
| Pesticide name | Insignia 20 WG |

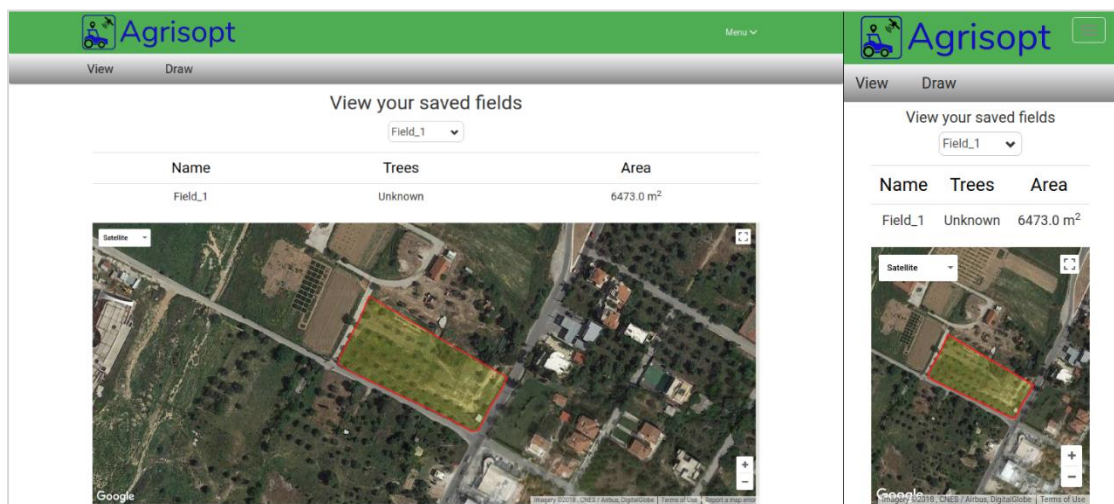
Εικόνα 4.78: Ο πίνακας με τα δεδομένα του ψεκασμού



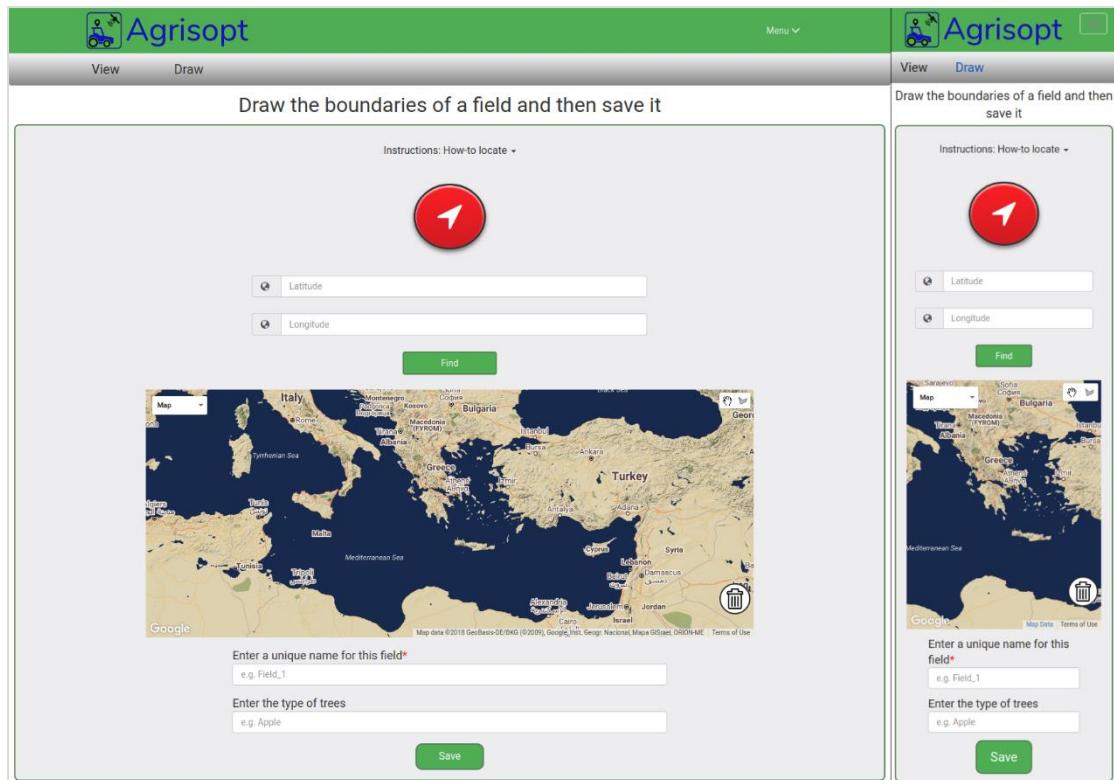
Εικόνα 4.79: Η προβολή του χάρτη μιας επέμβασης ψεκασμού



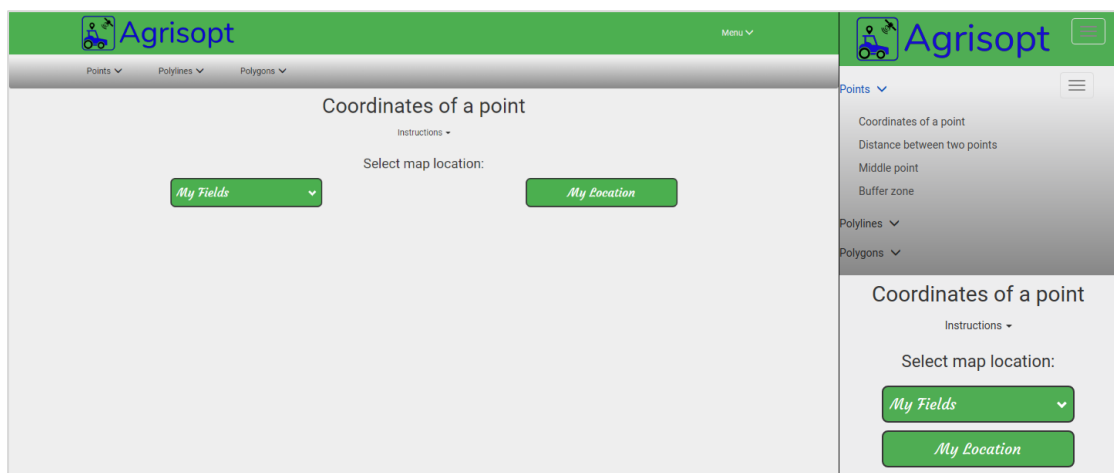
Εικόνα 4.80: Η προβολή του γραφήματος της ταχύτητας ενός ψεκασμού



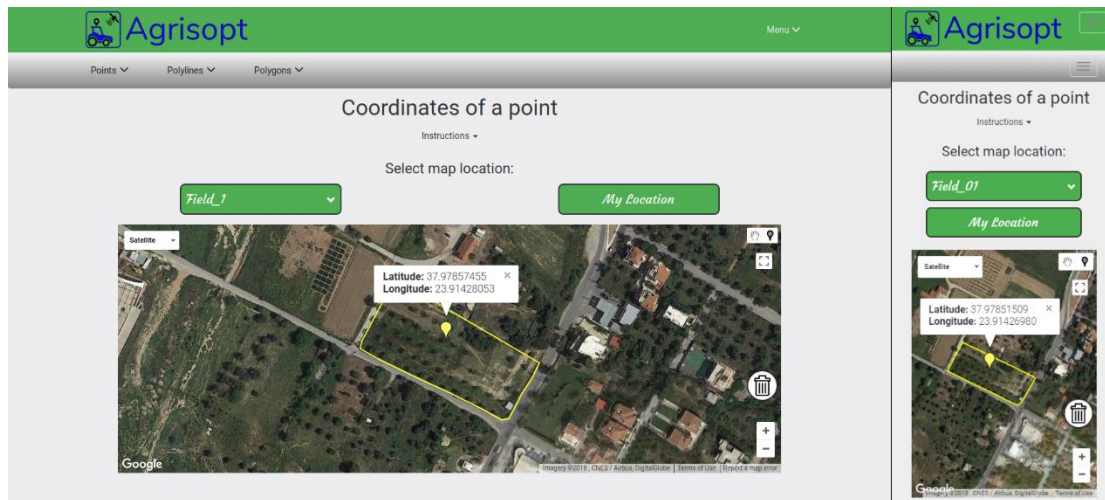
Εικόνα 4.81: Τα στοιχεία ενός αγροκτήματος



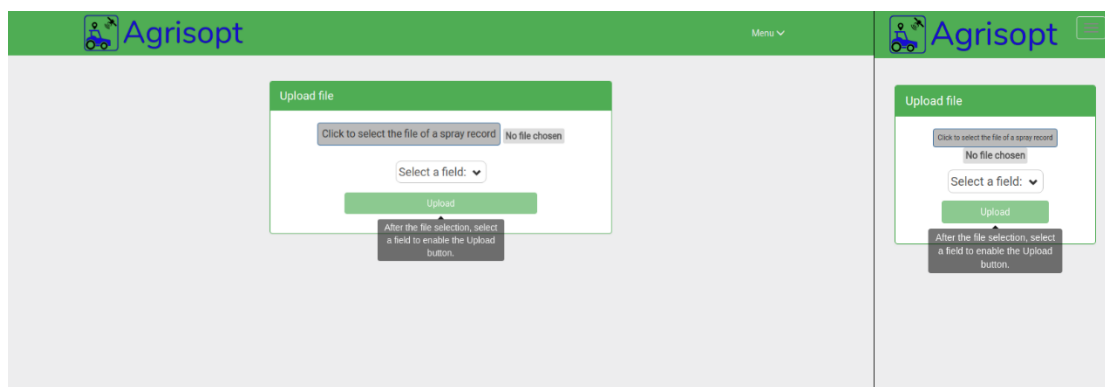
Εικόνα 4.82: Η φόρμα ψηφιοποίησης ενός αγροκτήματος



Εικόνα 4.83: Η επιλογή Spatial tools



Εικόνα 4.84: Το εργαλείο “Coordinates of a point”



Εικόνα 4.85: Η φόρμα της επιλογής Upload data

4.2.3 Συμβατότητα

Ο έλεγχος της συμβατότητας διακρίνεται σε τρεις κατηγορίες:

1. Συσκευές.
2. Λειτουργικά Συστήματα.
3. Φυλλομετρητές (Browsers).

Συσκευές

Οι συσκευές στις οποίες έγινε έλεγχος είναι οι εξής:

1. Desktop και Laptop Υπολογιστής.
2. Κινητό τηλέφωνο.
3. Tablet.

Σύμφωνα με στοιχεία από το netmarketshare.com το μερίδιο της αγοράς (market share) για το μέσο που χρησιμοποιούν οι χρήστες για πρόσβαση στο διαδίκτυο για το έτος 2017 είναι:

| Τύπος συσκευής | Μερίδιο |
|----------------------------|----------------|
| Desktop/Laptop υπολογιστές | 53.70% |
| Κινητά τηλέφωνα | 41.03% |
| Tablets | 5.22% |
| Άλλες συσκευές | 0.05% |

Επομένως, οι τύποι των συσκευών στις οποίες έγινε έλεγχος καλύπτουν το 99.95% των χρηστών του διαδικτύου.

Λειτουργικά Συστήματα

Τα λειτουργικά συστήματα ανά τύπο συσκευής στα οποία έγινε έλεγχος είναι:

Για desktop/laptop υπολογιστές:

Windows (Έκδοση 8.1).

Linux (Ubuntu 16.04).

Για κινητά τηλέφωνα:

Android (Έκδοση 6.0.1).

Για tablets:

Android (Έκδοση 4.4.2).

Με βάση τα στοιχεία από το netmarketshare.com το μερίδιο αγοράς για το έτος 2017 ανάλογα με το λειτουργικό σύστημα για κάθε τύπο συσκευής διαμορφώνεται ως εξής:

| Τύπος συσκευής | Λειτουργικό Σύστημα | Μερίδιο αγοράς |
|----------------------------|----------------------------|-----------------------|
| Desktop/Laptop υπολογιστές | Windows | 88.87% |
| | Linux | 2.33% |
| Κινητά τηλέφωνα | Android | 68.69% |
| Tablets | Android | 40.88% |

Επομένως, το μερίδιο αγοράς των χρηστών του διαδικτύου που χρησιμοποιούν τα λειτουργικά συστήματα στα οποία έγινε έλεγχος της εφαρμογής διαμορφώνεται για τους desktop/laptop υπολογιστές στο 91.2%, για τα κινητά τηλέφωνα στο 68.69% και για τα tablets στο 40.88%.

Φυλλομετρητές (Browsers)

Οι φυλλομετρητές στους οποίους έγινε έλεγχος ανά τύπο συσκευής είναι:

Για desktop/laptop υπολογιστές:

Google Chrome (Έκδοση 62.0.3202.75 (64-bit)).

Mozilla Firefox (Έκδοση 57.0.4 (64-bit)).

Για κινητά τηλέφωνα:

Google Chrome (Έκδοση 56.0.2924.87).

Mozilla Firefox (Έκδοση 57.0).

Για tablets:

Google Chrome (Έκδοση 63.0.3239.111).

Mozilla Firefox (Έκδοση 57.0.4).

Σύμφωνα με στοιχεία από το netmarketshare.com το μερίδιο της αγοράς για το έτος 2017 ανάλογα με το φυλλομετρητή για κάθε κατηγορία συσκευής είναι:

| Τύπος συσκευής | Φυλλομετρητής | Μερίδιο αγοράς |
|----------------------------|----------------------|-----------------------|
| Desktop/Laptop υπολογιστές | Google Chrome | 58.90% |
| | Mozilla Firefox | 13.29% |
| Κινητά τηλέφωνα | Google Chrome | 58.60% |
| | Mozilla Firefox | 0.51% |
| Tablets | Google Chrome | 34.49% |
| | Mozilla Firefox | 0.31% |

Επομένως, με βάση τους φυλλομετρητές το μερίδιο αγοράς των χρηστών του διαδικτύου που χρησιμοποιούν τους φυλλομετρητές στους οποίους έγινε έλεγχος της

εφαρμογής είναι για τους desktop/laptop υπολογιστές 72.19%, για τα κινητά τηλέφωνα 59.11% και για τα tablets 34.80%.

4.2.4 Απόδοση

Για τον έλεγχο της απόδοσης χρησιμοποιήθηκε το ApacheBench, ένα πρόγραμμα γραμμής εντολών το οποίο χρησιμοποιείτε για τη μέτρηση της απόδοσης των HTTP διακομιστών ιστού. Στα αποτελέσματα του ελέγχου εμφανίζεται ο χρόνος που χρειάστηκε ο διακομιστής για να εξυπηρετήσει τα αιτήματα του ApacheBench (συνολικός και ανά αίτημα χρόνος), τα αιτήματα που μπορούσε να εξυπηρετήσει ο διακομιστής ανά δευτερόλεπτο, ο ρυθμός μεταφοράς των δεδομένων, αν υπήρχαν αιτήματα που απέτυχε η εξυπηρέτησή τους και κάποια στατιστικά.

Τα τμήματα της εφαρμογής στα οποία εφαρμόστηκε ο έλεγχος είναι τα εξής:

| Τμήμα (Επιλογή Μενού) | URL στο οποίο αντιστοιχεί |
|--|---------------------------|
| Αρχική σελίδα | / |
| Σελίδα επεμβάσεων ψεκασμού (Operations) | /operations |
| Σελίδα αγροκτημάτων (Fields) | /field |
| Σελίδα χωρικών εργαλείων (Spatial tools) | /spatial-tools |
| Σελίδα καταγραφής ψεκασμού (Spray record) | /spray-record |
| Σελίδα φόρτωσης αρχείου ψεκασμού (Upload data) | /upload-form |

Το πρόγραμμα ApacheBench κάνει τα αιτήματα με βάση το URL που του δίνεται ως όρισμα για αυτό και στη συνέχεια για κάθε τμήμα της εφαρμογής θα αναφέρεται το URL στο οποίο αντιστοιχεί.

Κάποια γενικά στοιχεία του ελέγχου:

| | |
|----------------------|--|
| Λογισμικό διακομιστή | nginx/1.6.2 |
| Hostname | agrisopt.top |
| Θύρα διακομιστή | 443 |
| Πρωτόκολλο SSL/TLS | TLSv1.2,ECDHE-RSA-AES128-GCM-SHA256,2048,128 |

Οι εντολές που δόθηκαν:

```
$ ab -n 1000 -c 20 https://agrisopt.top/
```

```
$ ab -C session=.eJwIz..... -n 1000 -c 20 https://agrisopt.top/operations
```

\$ ab -C session=.eJwIz..... -n 1000 -c 20 <https://agrisopt.top/field>

\$ ab -C session=.eJwIz..... -n 1000 -c 20 <https://agrisopt.top/spatial-tools>

\$ ab -C session=.eJwIz..... -n 1000 -c 20 <https://agrisopt.top/spray-record>

\$ ab -C session=.eJwIz..... -n 1000 -c 20 <https://agrisopt.top/upload-form>

Για κάθε εντολή πραγματοποιούνται 1000 αιτήματα (-n 1000) προς το διακομιστή με τον ταυτοχρονισμό να είναι ίσος με 20 (-c 20). Για να υπάρχει πρόσβαση σε όλα τα URLs (εκτός αυτού της αρχικής σελίδας) χρειάστηκε ένα session cookie (-C session=.eJwIz.....).

Αποτελέσματα

Πίνακας 4.1: Ο συνολικός χρόνος για να ολοκληρωθεί ένας έλεγχος και η εξυπηρέτηση αιτημάτων ανά δευτερόλεπτο

| URL | Χρόνος για να ολοκληρωθεί ο έλεγχος (δευτερόλεπτα) | Εξυπηρέτηση αιτημάτων ανά δευτερόλεπτο (μέση τιμή) |
|----------------|--|--|
| / | 34.109 | 29.32 |
| /operations | 33.245 | 30.08 |
| /field | 32.505 | 30.76 |
| /spatial-tools | 33.187 | 30.13 |
| /spray-record | 32.696 | 30.58 |
| /upload-form | 32.573 | 30.70 |

Χρόνος ανά αίτημα

Ο χρόνος ανά αίτημα διακρίνεται σε δύο κατηγορίες, στη μια έχουμε το χρόνο ανά αίτημα για μια ομάδα ταυτόχρονων αιτημάτων, ενώ στην άλλη έχουμε το χρόνο για ένα μοναδικό αίτημα. Ουσιαστικά ο χρόνος ανά αίτημα της 2^{ης} κατηγορίας προκύπτει από τη διαίρεση του χρόνου της 1^{ης} με την τιμή του επιπέδου του ταυτοχρονισμού. Στο συγκεκριμένο έλεγχο το επίπεδο του ταυτοχρονισμού ήταν ίσο με 20, άρα αν έχουμε για την 1^η κατηγορία χρόνο ανά αίτημα ίσο με 1000 ms, τότε ο χρόνος ανά αίτημα για τη 2^η κατηγορία θα είναι ίσος με 50 ms.

Πίνακας 4.2: Ο χρόνος ανά αίτημα

| URL | για ομάδα ταυτόχρονων αιτημάτων (ms, μέση τιμή) | για ένα μοναδικό αίτημα (ms, μέση τιμή) |
|----------------|--|--|
| / | 682.188 | 34.109 |
| /operations | 664.908 | 33.245 |
| /field | 650.107 | 32.505 |
| /spatial-tools | 663.748 | 33.187 |
| /spray-record | 653.916 | 32.696 |
| /upload-form | 651.454 | 32.573 |

Πίνακας 4.3: Το σύνολο των Kilobytes που μεταφέρθηκαν και ο ρυθμός μεταφοράς των δεδομένων

| URL | Σύνολο Kilobytes που μεταφέρθηκαν | Ρυθμός μεταφοράς (Kilobytes/δευτερόλεπτο) |
|----------------|--|--|
| / | 13.055 | 382.76 |
| /operations | 6.529 | 196.40 |
| /field | 5.133 | 157.94 |
| /spatial-tools | 8.671 | 261.30 |
| /spray-record | 7.297 | 223.20 |
| /upload-form | 5.962 | 183.06 |

Πίνακας 4.4: Αιτήματα που ολοκληρώθηκαν και αιτήματα που απέτυχαν να ολοκληρωθούν

| URL | Ολοκληρωμένα αιτήματα | Μη Ολοκληρωμένα αιτήματα |
|----------------|------------------------------|---------------------------------|
| / | 1000 | 0 |
| /operations | 1000 | 0 |
| /field | 1000 | 0 |
| /spatial-tools | 1000 | 0 |
| /spray-record | 1000 | 0 |

| | | |
|--------------|------|---|
| /upload-form | 1000 | 0 |
|--------------|------|---|

Πίνακας 4.5: Ποσοστό αιτημάτων τα οποία εξυπηρετήθηκαν εντός ενός ορισμένου χρόνου (ms)

| | / | /operations | /field | /spatial-tools | /spray-record | /upload-form |
|------|------|-------------|--------|----------------|---------------|--------------|
| 50% | 663 | 653 | 637 | 649 | 640 | 638 |
| 66% | 674 | 662 | 644 | 656 | 646 | 644 |
| 75% | 685 | 666 | 648 | 662 | 650 | 648 |
| 80% | 692 | 670 | 650 | 666 | 653 | 650 |
| 90% | 713 | 678 | 658 | 680 | 660 | 658 |
| 95% | 731 | 690 | 668 | 696 | 671 | 671 |
| 98% | 755 | 708 | 687 | 711 | 738 | 697 |
| 99% | 807 | 744 | 727 | 727 | 751 | 723 |
| 100% | 1679 | 1527 | 1354 | 1466 | 1292 | 1417 |

4.2.5 Ασφάλεια

Τα μέτρα ασφαλείας διακρίνονται σε δύο κατηγορίες. Στη 1^η έχουμε τα μέτρα ασφαλείας που πάρθηκαν για την εφαρμογή και στη 2^η τα μέτρα ασφαλείας που πάρθηκαν για το διακομιστή. Στη συνέχεια ακολουθεί παράθεση των μέτρων ασφαλείας ανά κατηγορία.

Μέτρα ασφαλείας στην εφαρμογή

- Χρήση TLS Certificate από τον οργανισμό Let's Encrypt. Η σύνδεση στην εφαρμογή πραγματοποιείται μέσω κρυπτογράφησης και επικυρωμένης αυθεντικότητας, χρησιμοποιώντας το πρωτόκολλο TLS 1.2, το ECDHE_RSA με P-256 (ένα ισχυρό κλειδί που βασίζεται στον κρυπταλγόριθμο RSA) και το AES_128_GCM (ένας ισχυρός κρυπταλγόριθμος).
- Ασφαλής τρόπος αποθήκευσης των κωδικών στη βάση δεδομένων. Οι κωδικοί στη βάση δεδομένων για λόγους ασφαλείας δεν αποθηκεύονται με τη μορφή που τους εισάγει ο χρήστης. Στους κωδικούς προστίθεται μια τυχαία ακολουθία μήκους 8 χαρακτήρων και στη συνέχεια κατακερματίζονται με τη συνάρτηση κατακερματισμού SHA-256.

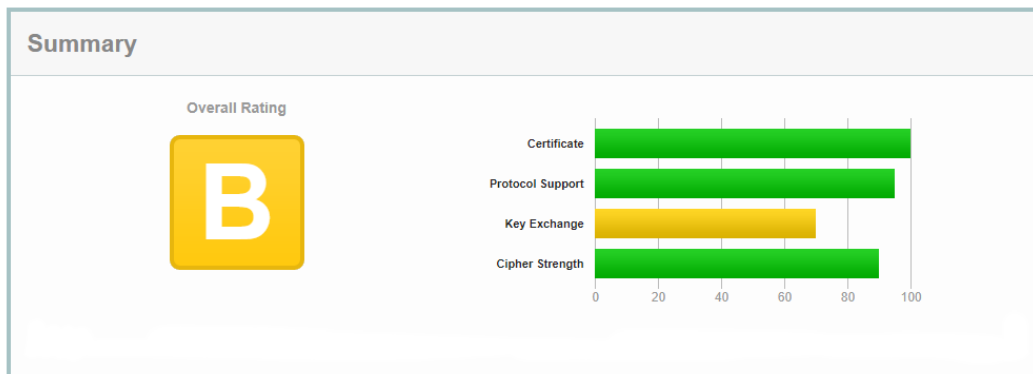
- Ύπαρξη προστασίας για την κακόβουλη τεχνική SQL Injection. Λόγω της χρήσης του ORM SQLAlchemy κατά την εκτέλεση των εντολών προς τη βάση δεδομένων δεν εκτελείται απευθείας κώδικας SQL, αλλά μέθοδοι της SQLAlchemy οι οποίες απορρίπτουν παραμέτρους και ειδικούς χαρακτήρες που θα μπορούσαν να μεταφραστούν ως εντολές SQL και να επιτεθούν στη βάση δεδομένων.
- Ύπαρξη συστήματος επικύρωσης της αυθεντικότητας του χρήστη της εφαρμογής. Για το σκοπό αυτό χρησιμοποιείται μια επέκταση του Flask framework το Flask-Login, το οποίο περιλαμβάνει προστασία ενός session για να προφυλάξει τα sessions των χρηστών από πιθανές κλοπές.
- Κατά την αποθήκευση του αρχείου με τα δεδομένα καταγραφής του ψεκασμού, γίνεται έλεγχος για τον περιορισμό φόρτωσης μη επιτρεπτών καταλήξεων αρχείων. Στη φόρμα "Upload file" μπορούν να ανεβούν αρχεία μόνο με την κατάληξη .json. Στη συνέχεια, αφού ένα αρχείο περάσει τον 1ο έλεγχο γίνεται έλεγχος της καταλληλότητας του περιεχομένου τους.
- Στις φόρμες Εγγραφής και Εισόδου του χρήστη (Registration & Login forms) γίνεται έλεγχος για αποφυγή αυτοματοποιημένης εισαγωγής στοιχείων από spam bots.
- Στο σύστημα καταγραφής (logging system) της εφαρμογής αποθηκεύονται πιθανές κακόβουλες ενέργειες είτε από ανώνυμους, είτε από εγγεγραμμένους χρήστες.

Μέτρα ασφαλείας στον διακομιστή

- Η σύνδεση στο διακομιστή γίνεται μέσω SSH (Secure Shell) χρησιμοποιώντας ένα λογαριασμό χρήστη-διαχειριστή που δημιουργήθηκε για την εφαρμογή.
- Η επικύρωση της αυθεντικότητας του διαχειριστή, κατά τη σύνδεση μέσω SSH, γίνεται μέσω ενός ζεύγους Δημόσιου-Ιδιωτικού κλειδιού (Public-Private Key pair) και ακολούθως μέσω ενός passphrase.
- Απενεργοποιήθηκε η δυνατότητα επικύρωσης της αυθεντικότητας χρήστη μέσω κωδικού.
- Καταργήθηκε η δυνατότητα σύνδεσης μέσω SSH του root χρήστη.
- Χρησιμοποιείται το εργαλείο fail2ban, το οποίο μπλοκάρει τις διευθύνσεις IP από τις οποίες γίνονται επανειλημμένα ενέργειες με σκοπό να αποκτήσουν πρόσβαση στον διακομιστή.
- Γίνεται καταγραφή σε αρχεία πιθανών κακόβουλων ενεργειών και πραγματοποιείται έλεγχος αυτών των αρχείων κατά περιόδους. Ένα από αυτά τα αρχεία χρησιμοποιεί το fail2ban για να μπλοκάρει τις διευθύνσεις IP.
- Η βάση δεδομένων είναι ρυθμισμένη έτσι ώστε να μην έχει πρόσβαση σε αυτή κάποιος εκτός του περιβάλλοντος του διακομιστή.
- Τα αρχεία που φορτώνονται στο διακομιστή μέσω της φόρμας "Upload file" αποθηκεύονται σε ένα κατάλογο που έχει περιορισμούς στα δικαιώματα και αφού διαβαστεί το περιεχόμενό τους, διαγράφονται από το δίσκο.

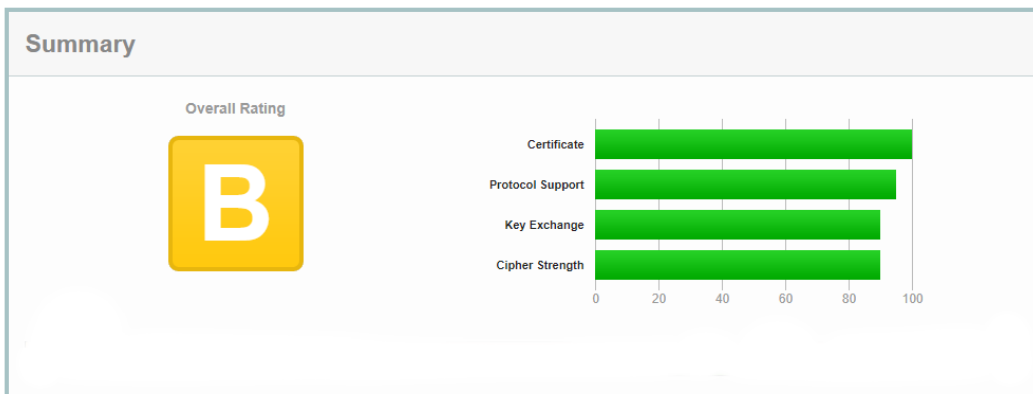
Στην εικόνα 4.86 παριτίθεται τα αποτελέσματα της αξιολόγησης από την ιστοσελίδα ssllabs.com για τον έλεγχο που πραγματοποιήθηκε στις ρυθμίσεις του SSL του διακομιστή. Στην εικόνα 4.87 παρατίθεται τα αντίστοιχα αποτελέσματα μιας δημοφιλούς ιστοσελίδας ως σημείο αναφοράς.

SSL Report: agrisopt.top



Εικόνα 4.86: Τα αποτελέσματα του ελέγχου των ρυθμίσεων SSL του διακομιστή

SSL Report: www.facebook.com



Εικόνα 4.87: Τα αντίστοιχα αποτελέσματα του ελέγχου μιας δημοφιλούς ιστοσελίδας

5 Συμπεράσματα

Τα κυριότερα συμπεράσματα που προέκυψαν κατά την εκπόνηση της παρούσας μεταπτυχιακής εργασίας είναι τα ακόλουθα:

- Οι δυνατότητες που μας δίνει το λογισμικό ανοιχτού κώδικα για ποιοτικές λύσεις δημιουργίας λογισμικού. Η εφαρμογή στην πλευρά του διακομιστή βασίστηκε αποκλειστικά σε λύσεις λογισμικού ανοιχτού κώδικα, όπως η Python, το Flask Web Framework και η PostgreSQL. Στην πλευρά του πελάτη με την εξαίρεση των Google Maps API και Google Charts API, οι υπόλοιπες τεχνολογίες που χρησιμοποιήθηκαν ήταν ανοιχτού κώδικα, όπως το jQuery, το Bootstrap και η Charts.js.
- Η ευκολία χρήσης του πρότυπου ανταλλαγής αρχείων JSON, με το πρότυπο αυτό ήταν αρκετά εύκολη και γρήγορη η ανάπτυξη της λειτουργικότητας για τη μεταφορά των δεδομένων.
- Οι δυνατότητες για απομακρυσμένη πρόσβαση που δίνουν οι εφαρμογές ιστού από διαφορετικά λειτουργικά συστήματα και συσκευές σε μια εφαρμογή που έχει την ίδια λειτουργικότητα, είτε τη χρησιμοποιεί κάποιος μέσα από το κινητό του, είτε από τον υπολογιστή του σπιτιού ή του γραφείου.
- Η ταχύτητα ανάπτυξης που προσφέρουν οι δυναμικές γλώσσες σεναρίων, όπως η Python και η JavaScript, για τουλάχιστον μικρού έως μεσαίου μεγέθους εφαρμογές.
- Τα μέτρα ασφαλείας που θα πρέπει να λάβει κάποιος που αναπτύσσει μια εφαρμογή ιστού. Για να δημιουργηθεί μια εφαρμογή στο διαδίκτυο που θα έχει μια σχετική ασφάλεια χρειάζονται κάποια επιπλέον βήματα κατά την υλοποίησή της και ένας διαρκής έλεγχος για ενημερώσεις και αναβαθμίσεις.

Μελλοντικές προεκτάσεις ή βελτιώσεις

Στη συνέχεια παρατίθενται μερικές προτάσεις για μια μελλοντική αναβάθμιση της παρούσας εφαρμογής:

- Επειδή, στην τωρινή περίοδο ζούμε την άνθιση της Επιστήμης των Δεδομένων και επειδή η εφαρμογή έχει τη δυνατότητα να συλλέξει ένα μεγάλο αριθμό δεδομένων, θα μπορούσαν αυτά τα δεδομένα να αξιοποιηθούν από κάποιον ειδικό για την εξαγωγή συμπερασμάτων.
- Η προσθήκη μιας εφαρμογής για κινητά τηλέφωνα θα βελτίωνε την εμπειρία χρήσης του μέσου χρήστη. Η εφαρμογή αυτή θα μπορούσε να συλλέγει τα δεδομένα που απαιτούνται και στη συνέχεια να τα στέλνει στο διακομιστή.
- Εφόσον υπάρχουν χωρικά δεδομένα στη βάση της εφαρμογής θα μπορούσαν με αυτά να γίνουν προχωρημένες χωρικές αναλύσεις για την εξαγωγή χρήσιμων συμπερασμάτων. Οι αναλύσεις αυτές θα μπορούσαν να

δημιουργηθούν μέσα στην ίδια την εφαρμογή, αφού κατά το σχεδιασμό της επιλέχθηκε η Python ως γλώσσα για την πλευρά του διακομιστή, λόγω της ευρείας χρήσης της στον τομέα των Γεωγραφικών Πληροφοριακών Συστημάτων, κάτι που έχει ως συνέπεια την ύπαρξη πολλών βιβλιοθηκών μέσω των οποίων μπορούν να γίνουν προχωρημένες χωρικές αναλύσεις. Από την άλλη, η επιλογή της PostgreSQL έγινε λόγω της ύπαρξης της επέκτασης PostGIS η οποία περιλαμβάνει λειτουργίες που βρίσκονται στις χωρικές βάσεις δεδομένων.

- Θα μπορούσε επίσης, να γίνει ένας επανασχεδιασμός του πηγαίου κώδικα με στόχο τη βέλτιστη επεκτασιμότητα και συντηρησιμότητά του.
- Για να υπάρξει πλήρης χρήση λογισμικού ανοιχτού κώδικα θα ήταν δυνατό να αντικατασταθεί το Google Maps API από τις JavaScript χωρικές βιβλιοθήκες OpenLayers ή Leaflet. Αντίστοιχα, το γράφημα που χρησιμοποιείται το Google Charts API θα μπορούσε να γίνει στην Chart.js.

Βιβλιογραφία

Αγγλική γλώσσα

- Beaulieu Alan. Learning SQL. 2nd edition. Sebastopol: O'Reilly, 2009.
- Boehlje, M.D., Eidman, V.R. Farm Management. Wiley, New York, 1984.
- Brookshear J. Glenn. Computer Science. 11th edition. Boston: Addison-Wesley, 2012.
- Brown Ethan. Learning JavaScript. 3rd edition. Sebastopol: O'Reilly, 2016.
- Ceder L. Vernon. The Quick Python Book. 2nd edition. Greenwich: Manning 2010.
- Crowther Rob, Lennon Joe, Blue Ash, Wanish Greg. HTML5 in Action. New York, 2014.
- Daniele Sarri , Luisa Martelloni, Marco Vieri, Development of a prototype of telemetry system for monitoring the spraying operation in vineyards. Computers and Electronics in Agriculture. Volume 142, Part A.2017. Pages 248-259. ISSN 0168-1699.
- Fountas S., Carli G., Sørensen C.G., Tsiropoulos Z., Cavalaris C., Vatsanidou A., Liakos B., Canavari M., Wiebensohn J., Tisserye B. Farm management information systems: Current situation and future perspectives. Computers and Electronics in Agriculture. Elsevier, 2015.
- Garrard Chris. Geoprocessing with Python. Shelter Island: Manning, 2016.
- Gennick Jonathan. SQL Pocket Guide. 3rd edition. Sebastopol: O'Reilly, 2011.
- Grinberg Miguel. Flask Web Development. Sebastopol: O'Reilly, 2014.
- Harvey Francis. Primer of GIS Fundamental Geographic and Cartographic Concepts. New York: The Guilford Press, 2008.
- Husemann Christoph, Novković Nebojša. Farm Management Information Systems: A Case Study On A German Multifunctional Farm. Economics of Agriculture, 2014.
- Juba Salahaldin, Vannahme Achim, Volkov Andrey. Learning PostgreSQL Birmingham: Packt Publishing, 2015.
- Konecny Gottfried. Geoinformation Remote Sensing, Photogrammetry, and Geographic Information Systems. 2nd edition. Boca Raton: CRC Press, 2014.
- Lawson Lartey , Pedersen Søren Marcus, Sørensen Claus Grøn, Pesonen Liisa, Fountas Spyros, Werner Armin, Oudshoorn Frank W., Herold Luzia, Chatzinikos

Thanos, Kirketerp Inger Marie, Blackmore Simon. A four nation survey of farm information management and advanced farming systems: A descriptive analysis of survey responses, Computers and Electronics in Agriculture, Elsevier, 2011.

- Lutz Mark. Learning Python. 5th edition. Sebastopol: O'Reilly, 2013.
- Lutz Mark. Python Pocket Reference. 5th edition. Sebastopol: O'Reilly, 2015.
- Martelli Alex, Ravenscroft Anna, Holden Steve. Python in a Nutshell. 3rd edition. Sebastopol: O'Reilly, 2017.
- Meyer A. Eric. CSS Pocket Reference. 4th edition. Sebastopol: O'Reilly, 2011.
- Moore K. Samuel. "Superaccurate GPS Coming to Smartphones in 2018". IEEE Spectrum. November 2017.
- Myers Jason, Copeland Rick. Essential SQLAlchemy. Sebastopol: O'Reilly, 2016.
- Novkovic Nebojsa, Huseman Christoph, Zoranovic Tihomir, Mutavdzic Beba, 2015, Farm Management Information Systems. HAICTA 2015: 705-712.
- Pesonen Liisa, Teye Frederick, Ronkainen Ari, Koistinen Markku, Kaivosoja Jere, Suomi Pasi, Linkolehto Raimo. Cropinfra: An Internet-based service infrastructure to support crop production in future farms. Biosystems Engineering. Volume 120. 2014. Pages 92-101. ISSN 1537-5110
- Peterson P. Michael. Mapping in the cloud. New York: The Guilford Press, 2014.
- Robbins N. Jennifer. HTML5 Pocket Reference. 5th edition. Sebastopol: O'Reilly, 2013.
- Rockoff Larry. The Language of SQL. 2nd edition. Indianapolis: Addison-Wesley, 2017.
- Salami Payman, and Ahmadi Hojat. Review of Farm Management Information Systems (FMIS). New York Science Journal, 2010.
- Sørensen C.G., Pesonen L., Bochtis D.D., Vougioukas S.G., Suomi P., Functional requirements for a future farm management information system. Computers and Electronics in Agriculture, Elsevier, 2010.
- Sørensen, G.C., Fountas, S., Nash, E., Pesonen, L., Bochtis, D., Pedersen, S.M., Basso, B., Blackmore, S.B.. Conceptual model of a future farm management information system. Comput. Electron. Agric. 72, 37–47, 2010.
- Stojanovic V., Falconer Ruth, Isaacs J., Blackwood D., Gilmour D., Kiezebrink D., Wilson J., Streaming and 3D mapping of AGRI-data on mobile devices, Computers and Electronics in Agriculture, Volume 138, 2017, Pages 188-199,ISSN 0168-1699

- Tanenbaum Andrew, Wetherall David. Computer Networks. 5th edition. Massachusetts: Pearson, 2011.
- Toal Ray, Rivera Rachel, Schneider Alexander, Choe Eileen. Programming Language Explorations. Boca Raton: CRC Press, 2017.
- Tsiropoulos Zisis & Fountas Spyros. (2015). Farm management information system for fruit orchards. 429-436. 10.3920/978-90-8686-814-8_53.
- Zakas Nicholas. JavaScript for Web Developers. 3rd edition. Indianapolis: John Wiley & Sons, Inc, 2012.

Ελληνική γλώσσα

- Αλεξανδρής Πέτρος. «Ψηφιακή γεωργία, χωρικά καθορισμένη». Agricola (Καθημερινή). 17 Φεβρουαρίου 2018. 58-60.
- Γέμτος Θεοφάνης, Καβαλάρης, Χρήστος. Μηχανήματα καλλιεργητικών φροντίδων. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών, 2015. Διαθέσιμο στο: <http://hdl.handle.net/11419/1325>
- Φουντάς Σπύρος, Γέμτος Θεοφάνης. Γεωργία ακριβείας. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών, 2015. Διαθέσιμο στο: <http://hdl.handle.net/11419/2670>
- Elmasri Ramez, Navathe B. Shamkant. Θεμελιώδεις Αρχές Συστημάτων Βάσεων Δεδομένων 6^η έκδοση. Αθήνα: Δίαυλος, 2012.