



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

**Σχολή Αγρονόμων Τοπογράφων Μηχανικών & Μηχανικών
Γεωπληροφορικής
ΔΠΜΣ Γεωπληροφορικής
Εργαστήριο Φωτογραμμετρίας**

**Μεταπτυχιακή Διπλωματική Εργασία
«Ανάπτυξη Ανασκαφικής Διαδικτυακής Εφαρμογής»**

**Κατερίνα Μητροπούλου
Αγρ. Τοπογράφος Μηχανικός ΕΜΠ**

**Επιβλέπων: Ανδρέας Γεωργόπουλος
Ομότιμος Καθηγητής ΕΜΠ**

**Αθήνα
Ιούλιος 2024**

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες στον επιβλέποντα καθηγητή κο Γεωργόπουλο Ανδρέα για τη βοήθεια και τη στήριξη της επιλογής μου, ώστε να εκπονήσω αυτή την εργασία.

Ιδιαίτερες ευχαριστίες θέλω να απευθύνω στην Αρχαιολόγο και φίλη μου κα Γεωργία Ταμπακοπούλου για την πολύτιμη βοήθεια της στον σχεδιασμό του εννοιολογικού μοντέλου της ανασκαφικής βάσης δεδομένων.

Περίληψη

Στο πλαίσιο εκπόνησης της μεταπτυχιακής διπλωματικής εργασίας αναπτύχθηκε μια ανασκαφική διαδικτυακή εφαρμογή. Η εφαρμογή αναπτύχθηκε στο ολοκληρωμένο περιβάλλον ανάπτυξης κώδικα Microsoft Visual Studio και στο σύστημα διαχείρισης βάσεων δεδομένων Microsoft SQL Server.

Η αρχιτεκτονική που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής ήταν η MVC (Model-View-Controller), η οποία περιλαμβάνει τη δημιουργία μοντέλων (models) για τα δεδομένα, προβολών (views) για τη διεπαφή χρήστη και ελεγκτών (controllers) για τον χειρισμό των λειτουργιών των δεδομένων από τον χρήστη. Ο κώδικας αναπτύχθηκε, χρησιμοποιώντας τις γλώσσες προγραμματισμού `c#`, `html`, `css` και `javascript`.

Το πληροφοριακό σύστημα επιμερίστηκε στην ανάπτυξη έξι υποσυστημάτων και συγκεκριμένα, στο υποσύστημα Λειτουργιών CRUD, στο υποσύστημα Χαρτογραφικού υποβάθρου, στο υποσύστημα Χωρικών Ερωτημάτων, στο υποσύστημα Αποθήκευσης Πολυμέσων, στο υποσύστημα Αυθεντικοποίησης Χρήστη και στο υποσύστημα Διαχείρισης Πρόσβασης.

Τέλος, η εφαρμογή αξιολογήθηκε ως προς την λειτουργικότητα της και προτάθηκαν κάποιες μελλοντικές βελτιώσεις που αφορούν κυρίως διάφορους αυτοματισμούς.

Abstract

An excavation web application was developed as part of the master's thesis. The application was developed in the Microsoft Visual Studio integrated code development environment and the Microsoft SQL Server database management system.

The architecture used to develop the application was MVC (Model-View-Controller), which includes the creation of models for the data, views for the user interface and controllers for handling the operations of the data by the user. The code was developed using c#, html, css and javascript programming languages.

The information system was divided into the development of six subsystems, namely, CRUD Operations subsystem, Mapping Background subsystem, Spatial Query subsystem, Multimedia Storage subsystem, User Authentication subsystem, and the Access Management subsystem.

Finally, the application was evaluated in terms of its functionality and some future improvements were proposed, mainly concerning various automations.

Περιεχόμενα

1. Εισαγωγή	8
2. Βιβλιογραφική Επισκόπηση	10
3. Θεωρητική Προσέγγιση	12
3.1 Ορισμός Διαδικτυακής Εφαρμογής.....	12
3.1.1 Η Εξέλιξη των Διαδικτυακών Εφαρμογών	13
3.1.2 Λειτουργία των εφαρμογών Ιστού	14
3.1.3 Τεχνολογίες Εφαρμογών Ιστού Front-End	15
3.1.4 Τεχνολογίες εφαρμογών Ιστού Back-end	16
3.1.5 Πλεονεκτήματα και μειονεκτήματα διαδικτυακών εφαρμογών	18
3.2 Αποκριτικές Εφαρμογές Ιστού	22
3.2.1 Πλεονεκτήματα των Αποκριτικών Εφαρμογών Ιστού	23
3.3 Μοντέλο Ανάπτυξης Εφαρμογών Ιστού MVC (Model – View – Controller)	24
3.3.1 Σύνοψη ιστορία του Πλαισίου MVC	24
3.3.2 Αρχιτεκτονική MVC.....	25
3.3.3 Πλεονεκτήματα του μοτίβου σχεδίασης MVC.....	27
3.4 Αντικειμενοστραφής Προγραμματισμός	30
3.5 Βάσεις Δεδομένων	31
3.5.1 Τύποι βάσεων δεδομένων	31
3.5.2 Σχεδιασμός Βάσεων δεδομένων	35
3.5.3 Εννοιολογικός, Λογικός και Φυσικός σχεδιασμός στις βάσεις δεδομένων.....	36
3.5.4 Το μοντέλο Οντοτήτων-Συσχετίσεων	37
3.5.5 Χωρικές βάσεις δεδομένων	38
3.5.6 Χωρικά ερωτήματα.....	39
3.5.7 Γλώσσα προγραμματισμού SQL	40
4. Ανάπτυξη Εφαρμογής	42
4.1 Εργαλεία Ανάπτυξης της Εφαρμογής.....	42
4.2 Προσέγγιση του εννοιολογικού μοντέλου της χωρικής βάσης δεδομένων.....	44
4.2.1 Σχεδιασμός Εννοιολογικού μοντέλου της χωρικής βάσης της εφαρμογής.....	45
4.2.2 Απεικόνιση του σχήματος της βάσης μέσω του διαγράμματος οντοτήτων – συσχετίσεων	48

4.3 Σχεδιασμός Λογικού και Φυσικού μοντέλου της χωρικής βάσης της εφαρμογής	49
4.4 Δημιουργία περιβάλλοντος εργασίας	52
4.5 Περιγραφή λειτουργιών εφαρμογής.....	53
4.5.1 Υποσύστημα Λειτουργιών CRUD (E01)	54
4.5.2 Υποσύστημα Χαρτογραφικού υποβάθρου (E02)	61
4.5.3 Υποσύστημα Χωρικών Ερωτημάτων (E03).....	63
4.5.4 Υποσύστημα Αποθήκευσης Πολυμέσων (E04)	64
4.5.5 Υποσύστημα Αυθεντικοποίησης Χρήστη (E05)	66
4.5.6 Υποσύστημα Διαχείρισης Πρόσβασης (E06).....	68
4.6 Συνδυασμός Υποσυστημάτων – Τελική Διεπαφή Χρήστη.....	71
5. Συμπερασματικά Σχόλια	81
5.1 Μελλοντικές Βελτιώσεις.....	84
6. Βιβλιογραφία	86

1. Εισαγωγή

Τα τελευταία χρόνια η τεκμηρίωση του αρχαιολογικού έργου γίνεται όλο και περισσότερο με τη συλλογή και αποθήκευση γεω-δεδομένων. Η οργάνωση και η διαχείριση της περιγραφικής και χωρικής πληροφορίας των γεω-δεδομένων καθιστά επιτακτική την ανάγκη της αποθήκευσης τους σε χωρικές βάσεις δεδομένων.

Με απώτερο σκοπό, την αποθήκευση και διαχείριση των ανασκαφικών δεδομένων σε βάσεις δεδομένων από μη εξοικειωμένους χρήστες, βασική επιδίωξη της εργασίας ήταν η ανάπτυξη μιας ανασκαφικής διαδικτυακής εφαρμογής.

Η εφαρμογή επιλέχθηκε να είναι διαδικτυακή λόγω της άμεσης πρόσβασης από οποιαδήποτε συσκευή είναι συνδεδεμένη στο διαδίκτυο, χωρίς να απαιτείται η εγκατάσταση κάποιου επιπρόσθετου λογισμικού. Η μόνη απαίτηση είναι η ύπαρξη περιηγητή διαδικτύου (browser), ο οποίος πλέον είναι εγκατεστημένος σε όλα τα λειτουργικά συστήματα ακόμα και στις φορητές συσκευές. Επιπροσθέτως, οι διαδικτυακές εφαρμογές δεν καταναλώνουν υπολογιστικούς πόρους, διότι δεν εκτελούνται στη συσκευή του χρήστη και δεν καταλαμβάνουν χώρο, αφού το σύνολο της εφαρμογής είναι αποθηκευμένο σε έναν κεντρικό εξυπηρετητή και όχι στον σκληρό δίσκο της συσκευής του χρήστη. Ακόμη, ένα σημαντικό πλεονέκτημα είναι ότι μία ενδεχόμενη αναβάθμιση του συστήματος γίνεται στον κεντρικό εξυπηρετητή και ταυτόχρονα είναι διαθέσιμη σε όλους τους χρήστες.

Η διαδικτυακή εφαρμογή αναπτύχθηκε ως βασικό εργαλείο για την καταγραφή της ανασκαφικής διαδικασίας σε πραγματικό χρόνο και δεν έχει σκοπό να υποκαταστήσει την παραδοσιακή μέθοδο χειρόγραφης καταγραφής στο ανασκαφικό ημερολόγιο. Βασικός σκοπός της δημιουργίας της είναι να λειτουργήσει συμπληρωματικά, αξιοποιώντας τις δυνατότητες και τα οφέλη του να έχεις δεδομένα αποθηκευμένα σε βάσεις δεδομένων.

Η ιδέα σχεδιασμού και υλοποίησης της εφαρμογής προέκυψε από την αναγκαιότητα τυποποίησης και διαχείρισης της ανασκαφικής πληροφορίας, ειδικά στις περιπτώσεις εκτεταμένων χωρικά ή/και χρονικά ανασκαφών. Λαμβάνοντας υπόψη ότι σε διαφορετικά είδη ανασκαφής (σωστική/συστηματική, χερσαία/ενάλια, κοκ) προκύπτουν διαφορετικές στρατηγικές καταγραφής, οι οποίες ασφαλώς δεν μπορούν να ταξινομηθούν σε ένα ενιαίο σύστημα, το οποίο να τις συμπεριλαμβάνει όλες και ταυτόχρονα να είναι εύχρηστο και λειτουργικό, επιλέχθηκε να διαμορφωθεί ο εννοιολογικός σχεδιασμός σύμφωνα με τη βασική πληροφορία που καταγράφεται συνήθως στα αρχαιολογικά ημερολόγια χερσαίων ανασκαφών.

Η εφαρμογή περιλαμβάνει εκτενέστατη ανάπτυξη λογισμικού που ήταν απαραίτητη για την υλοποίηση της. Η αρχιτεκτονική που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής ήταν η MVC (Model-View-Controller), η οποία περιλαμβάνει τη δημιουργία μοντέλων (models) για τα δεδομένα, προβολών (views) για τη διεπαφή χρήστη και ελεγκτών (controllers) για τον χειρισμό των λειτουργιών των δεδομένων από τον χρήστη. Ο κώδικας αναπτύχθηκε, χρησιμοποιώντας τις γλώσσες προγραμματισμού c#, html, css και javascript.

Το πληροφοριακό σύστημα επιμερίστηκε στην ανάπτυξη έξι υποσυστημάτων και συγκεκριμένα, στο υποσύστημα Λειτουργιών CRUD, στο υποσύστημα Χαρτογραφικού υποβάθρου, στο υποσύστημα Χωρικών Ερωτημάτων, στο υποσύστημα Αποθήκευσης Πολυμέσων, στο υποσύστημα Αυθεντικοποίησης Χρήστη και στο υποσύστημα Διαχείρισης Πρόσβασης.

Όσον αφορά, την διάρθρωση της εργασίας, στο δεύτερο κεφάλαιο παρουσιάζεται η σχετική βιβλιογραφία. Στο τρίτο κεφάλαιο περιγράφονται οι θεωρητικές έννοιες προσέγγισης του θέματος. Στο τέταρτο κεφάλαιο παρουσιάζεται η ανάπτυξη της εφαρμογής και στο πέμπτο κεφάλαιο παρουσιάζονται κάποια συμπεράσματα και κάποιες μελλοντικές βελτιώσεις. Τα συμπεράσματα επικεντρώθηκαν κυρίως σε θέματα λειτουργικότητας της εφαρμογής και στις μελλοντικές βελτιώσεις προτάθηκαν κυρίως κάποιοι αυτοματισμοί, οι οποίες θα ενισχύσουν την ορθότητα της πληροφορίας των δεδομένων την αποτελεσματικότερη θέασή τους και τη στατιστική ανάλυση της πληροφορίας της βάσης με τον υπολογισμό κάποιων μεγεθών που θεωρείται κρίσιμο να μελετηθούν.

2. Βιβλιογραφική Επισκόπηση

Μία τεχνική συστήματος διαχείρισης του πολιτιστικού αποθέματος της Αρχαίας Αγοράς στην Αθήνα αναπτύχθηκε από τους S. Hatzithoma – Panagiotou et al το 2014. Το σύστημα βασίζεται στο λογισμικό ArcGIS, χρησιμοποιεί ως χαρτογραφικό υπόβαθρο την ορθοφωτογραφία της περιοχής και για το σχεδιασμό του ελήφθησαν υπόψη διάφορες παράμετροι. Για την ανάπτυξη του αντικειμενοστραφούς συστήματος προηγήθηκε ο εννοιολογικός σχεδιασμός της Βάσης Δεδομένων με το Μοντέλο Οντοτήτων-Συσχετίσεων (E-R Model). Τον σχεδιασμό χωρικών δεδομένων του συστήματος στο ArcGIS, ακολούθησε η κατασκευή της Βάσης Δεδομένων που περιλαμβάνει τις περιγραφικές πληροφορίες για τα Μνημεία. Το εργαλείο που επιλέχθηκε για το σκοπό αυτό ήταν μια γεωγραφική βάση δεδομένων που δημιουργήθηκε στο ArcGIS και επεκτάθηκε και αναπτύχθηκε πλήρως σε περιβάλλον MS Access. Η εφαρμογή ArcGIS και η βάση δεδομένων της Microsoft Access αλληλεπιδρούν και συνδέονται σωστά, αλλά μπορούν να χρησιμοποιηθούν χωριστά, ανάλογα με τις ανάγκες του χρήστη.

Δημιουργήθηκε επίσης μια εφαρμογή Java για την ευκολότερη παρουσίαση των δεδομένων που είναι αποθηκευμένα στη βάση δεδομένων. Η εφαρμογή εκτελείται χρησιμοποιώντας Mozilla Firefox. Η εφαρμογή βασίζεται σε ερωτήματα SQL που διατυπώθηκαν προσεκτικά για την επιλογή και εξαγωγή των περιγραφικών πληροφοριών της βάσης δεδομένων μαζί με τις σχέσεις τους, σε αρχεία XML. Αυτά τα αρχεία χρησιμοποιήθηκαν σε έναν κώδικα Java script, ο οποίος εκτελείται με χρήση του Mozilla Firefox και συμβάλλει σε μια απλή, οργανωμένη παρουσίαση των δεδομένων. Η Βάση Δεδομένων Πρόσβασης περιλαμβάνει μια ποικιλία πληροφοριών για τα Μνημεία της Αρχαίας Αθηναϊκής Αγοράς που αποθηκεύονται σε μορφή πίνακα. Για την παρουσίασή τους δημιουργήθηκε η εφαρμογή Java. Το πρώτο βήμα ήταν ο σχηματισμός ερωτημάτων στο MS Access και η εξαγωγή των αποτελεσμάτων της εκτέλεσής τους σε αρχεία XML. Τα ερωτήματα δημιουργήθηκαν χρησιμοποιώντας γλώσσα SQL. Σχηματίστηκαν με σκοπό την επιλογή των κατάλληλων αρχείων από τη Βάση Δεδομένων, μαζί με κάποιες από τις περιγραφικές τους πληροφορίες και ταυτόχρονα για τη διατήρηση των σχέσεων μεταξύ τους και με τα πολύγωνα του ArcGIS. Το σενάριο Java δομήθηκε με βάση αυτά τα αρχεία XML.

Ακόμη, μία τεχνική ανάπτυξης γεωχωρικής εφαρμογής αναπτύχθηκε από τους S. Tarinaki et al το 2016 για τον αρχαιολογικό χώρο της Ελεύθερας στην Κρήτη. Μια ειδική εφαρμογή, με το όνομα ARCHAEOsystem, αναπτύχθηκε σε γλώσσα προγραμματισμού Visual Basic .NET χρησιμοποιώντας το ολοκληρωμένο περιβάλλον ανάπτυξης κώδικα το Visual Studio 2010. Αυτή η εφαρμογή χρησιμοποιεί

το .NET API του AutoCAD Map 3D 2012 και επιτρέπει τη διαχείριση και την οπτικοποίηση περιγραφικών και χωρικών δεδομένων σε ένα ολοκληρωμένο περιβάλλον. Τα χωρικά αντικείμενα που είναι αποθηκευμένα στον SQL Server οπτικοποιούνται στον χάρτη AutoCAD χρησιμοποιώντας την τεχνολογία πρόσβασης δεδομένων ανοικτού κώδικα Feature Data Object (FDO). Μέσω της τεχνολογίας FDO, ο χάρτης AutoCAD δεν χρησιμοποιείται με τον παραδοσιακό τρόπο επεξεργασίας αντικειμένων CAD, αλλά λειτουργεί περισσότερο σαν ένα εργαλείο επεξεργασίας και οπτικοποίησης δεδομένων που είναι αποθηκευμένα σε άλλο αποθηκευτικό χώρο, σε αυτήν την περίπτωση στον SQL Server. Η πρόσβαση σε θεματικά δεδομένα, αποθηκευμένα και στον SQL Server, επιτυγχάνεται με τη χρήση του Entity Framework (EF), μέσω της ειδικής εφαρμογής που αναπτύχθηκε. Το EF είναι ένα πλαίσιο ανοιχτού κώδικα που μετατρέπει τους πίνακες της βάσης δεδομένων σε αντικείμενα στο περιβάλλον προγραμματισμού εφαρμογών του Microsoft Visual Studio.

Επίσης, μια τεχνική αρχαιολογικού πληροφοριακού συστήματος αναπτύχθηκε από τους Α. Γεωργόπουλος et al για τον αρχαιολογικό χώρο της Πολιόχνης της Λήμνου. Το σύστημα αποτελείται από τρία υποσυστήματα, τη την Βάση Δεδομένων (PostgreSQL με επέκταση PostGIS), το Γεωχωρικό Σύστημα πληροφοριών (Quantum GIS) και την Ειδική Εφαρμογή για την διαχείριση της Βάσης Δεδομένων σε περιβάλλον Desktop.

Η Βάση Δεδομένων βασίζεται σε μια σχεσιακή βάση δεδομένων (RDBMS) ανοιχτού κώδικα, PostgreSQL με την επέκταση PostGIS για την υποστήριξη χωρικών δεδομένων. Το PostgreSQL είναι ένα δωρεάν σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), ανοιχτού κώδικα και πολλαπλών πλατφορμών, που δίνει έμφαση στην επεκτασιμότητα και στην συμμόρφωση με την SQL. Υποστηρίζει συναλλαγές, προβολές, ενεργοποιήσεις, αποθηκευμένες διαδικασίες και προνόμια πολλών χρηστών. Το σύστημα αυτό περιλαμβάνει τους πίνακες και τις συσχετίσεις για όλες τις χωρικές και περιγραφικές οντότητες.

Το σύστημα προβολής, δημιουργίας και τροποποίησης χωρικών δεδομένων είναι μια λύση ανοιχτού κώδικα του Quantum GIS®. Το Quantum GIS είναι μια δωρεάν και ανοιχτού κώδικα εφαρμογή συστήματος γεωγραφικών πληροφοριών για υπολογιστές που υποστηρίζει την προβολή, την επεξεργασία και την ανάλυση γεωχωρικών δεδομένων.

Η εφαρμογή για την εισαγωγή περιγραφικών δεδομένων έχει αναπτυχθεί σε περιβάλλον Desktop και αποτελείται από φόρμες που αντιπροσωπεύουν τις σχέσεις των πινάκων στη βάση δεδομένων και έτσι είναι ένας προηγμένος

τύπος εφαρμογής CRUD (Δημιουργία, Ανάγνωση, Ενημέρωση, Διαγραφή). Ο χρήστης μπορεί να διαβάσει οντότητες, να τις τροποποιήσει, και να δημιουργήσει νέες, να κλειδώσει σχέσεις και να δει πολλές οντότητες σε μία φόρμα.

Μια ακόμη τεχνική διαδικτυακής βάσης δεδομένων αναπτύχθηκε από τον Λιανό το 2023. Η ανάπτυξη της βάσης έγινε με τη βοήθεια του λογισμικού KoboToolBox. Επίσης, πραγματοποιήθηκε μία πρακτική εφαρμογή στο πεδίο και παρουσιάστηκαν τα αποτελέσματα. Επιπλέον, αναλύθηκε η εξαγωγή των χωρικών δεδομένων της βάσης στο QGIS.

3. Θεωρητική Προσέγγιση

Στο κεφάλαιο αυτό, θα περιγραφούν οι θεωρητικές έννοιες προσέγγισης του θέματος της εργασίας. Πιο συγκεκριμένα, θα αναλυθεί ο όρος της διαδικτυακής εφαρμογής, θα περιγραφεί το μοντέλο αρχιτεκτονικής ανάπτυξης εφαρμογών ιστού το MVC, θα γίνει κάποια περιγραφή των στοιχείων του αντικειμενοστραφή προγραμματισμού, τέλος θα αναλυθεί ένα μέρος της θεωρίας των βάσεων δεδομένων.

3.1 Ορισμός Διαδικτυακής Εφαρμογής

Διαδικτυακή εφαρμογή (web application ή web app) ονομάζεται κάθε εφαρμογή η οποία είναι διαθέσιμη στους χρήστες της μέσω του Διαδικτύου (Internet) ή του ενδοδικτύου (Intranet) ενός οργανισμού και ο χρήστης χρειάζεται μόνο τον περιηγητή (browser) του για να την χρησιμοποιήσει. Οι εφαρμογές αυτές συνήθως εκτελούνται σε ισχυρές υπολογιστικές μηχανές οι οποίες έχουν τον ρόλο του σταθμού εξυπηρέτησης και παρέχουν τις υπηρεσίες τους σε περισσότερους του ενός χρήστη.

Οι εφαρμογές ιστού λειτουργούν όπως τα παραδοσιακά προγράμματα λογισμικού, αλλά αντί να εκτελείται κώδικας τοπικά, εκτελείται σε απομακρυσμένους διακομιστές και χρησιμοποιείται το Διαδίκτυο για να παρέχει τη διεπαφή χρήστη μέσω ενός προγράμματος περιήγησης ιστού. Αυτή η αρχιτεκτονική καθιστά τις εφαρμογές ιστού ανεξάρτητες από την πλατφόρμα, που σημαίνει ότι μπορούν να εκτελούνται σε οποιαδήποτε συσκευή με πρόγραμμα περιήγησης ιστού και σύνδεση στο διαδίκτυο.

Η λειτουργικότητα των εφαρμογών ιστού μπορεί να ανταγωνιστεί τις εφαρμογές επιτραπέζιου υπολογιστή, ενώ προσφέρει πανταχού παρούσα πρόσβαση και ευκολία κοινής χρήσης.

(<https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app>, last visit 29/5/2024)

3.1.1 Η Εξέλιξη των Διαδικτυακών Εφαρμογών

Στις πρώτες μέρες του Διαδικτύου στη δεκαετία του 1990, οι ιστότοποι ήταν στατικοί, κυρίως ήταν ιστοσελίδες συνδεδεμένες μέσω υπερσυνδέσμων. Οι πρώτες διαδικτυακές εφαρμογές που εμφανίστηκαν έπρεπε να είναι πιο λειτουργικές. Μερικά πρώτα παραδείγματα εφαρμογών ιστού ήταν τα φόρουμ, όπως επίσης και οι απλές εφαρμογές ηλεκτρονικών αγορών. Αυτές οι εφαρμογές ιστού επέτρεπαν ουσιαστική αλληλεπίδραση με τον χρήστη, αλλά η εμπειρία του χρήστη θα μπορούσε να ήταν πιο εμφανής. Κάθε ενέργεια απαιτούσε επαναφόρτωση της σελίδας, κάτι που έκανε τις εφαρμογές αργές.

Στις αρχές της δεκαετίας του 2000, οι διαδικτυακές εφαρμογές άρχισαν να γίνονται πιο εξελιγμένες με την εισαγωγή της τεχνολογίας AJAX (Asynchronous JavaScript και XML). Η συγκεκριμένη τεχνολογία είναι ένα σύνολο τεχνικών ανάπτυξης ασύγχρονων εφαρμογών στο Διαδίκτυο. Με την AJAX οι εφαρμογές ιστού μπορούν να στέλνουν και να λαμβάνουν δεδομένα στο παρασκήνιο χωρίς να τις απασχολεί τι προβάλλεται στην σελίδα. Ο διαχωρισμός της ανταλλαγής δεδομένων με την προβολή τους επιτρέπει στην ιστοσελίδα να αλλάζει το περιεχόμενο της δυναμικά χωρίς να χρειάζεται ολοκληρωτική ανανέωσή της.

Περίπου την ίδια περίοδο, οι εφαρμογές ιστού άρχισαν να αξιοποιούν εκτενέστερα τα API. Η τεχνολογία API (Application Programming Interface) είναι γνωστή και ως 'Διασύνδεση Προγραμματισμού Εφαρμογών' και είναι η διεπαφή των προγραμματιστικών διαδικασιών που παρέχει ένα λειτουργικό σύστημα, ή κάποιες εφαρμογές προκειμένου να γίνονται αιτήσεις από άλλα προγράμματα προς αυτά. Οι υπηρεσίες ιστού όπως οι Χάρτες Google και οι πλατφόρμες κοινωνικών μέσων εισήγαγαν API που επέτρεψαν σε άλλους ιστότοπους να ενσωματώσουν τη λειτουργικότητά τους.

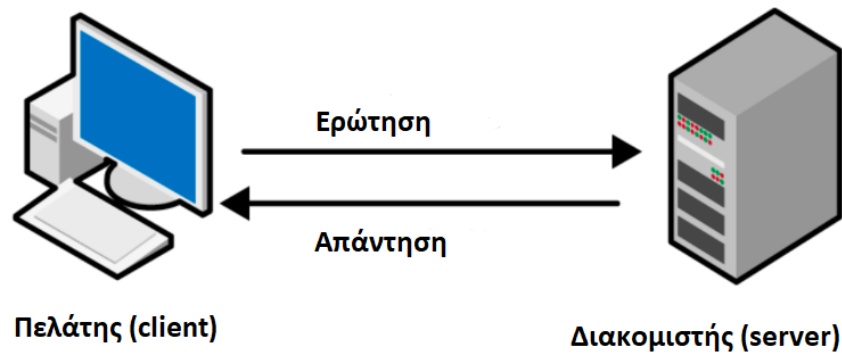
Στα τέλη της δεκαετίας του 2000, τα ολοκληρωμένα πλαίσια (frameworks) εφαρμογών ιστού όπως το Ruby on Rails κέρδισαν δημοτικότητα. Σε συνδυασμό με τη συνεχή υιοθέτηση της τεχνολογίας AJAX, αυτά τα πλαίσια (frameworks)

επέτρεψαν στους προγραμματιστές να δημιουργήσουν εξαιρετικά διαδραστικές εφαρμογές ιστού. Το Gmail και τα Έγγραφα Google ήταν μερικά από τα πρώτα παραδείγματα αυτής της νέας γενιάς πολύπλοκων εφαρμογών ιστού.

Στις αρχές της δεκαετίας του 2010, εμφανίστηκαν πλαίσια (frameworks) JavaScript όπως το AngularJS, επιτρέποντας στους προγραμματιστές να δημιουργήσουν ακόμη πιο εξελιγμένες εφαρμογές. Πιο πρόσφατα πλαίσια (frameworks) όπως το React συνέχισαν να υπερβαίνουν τα όρια του δυνατού στον Ιστό. Οι σημερινές εφαρμογές ιστού μπορούν να προσφέρουν εμπειρίες χρηστών έναντι των εγγενών εφαρμογών για τους επιτραπέζιους υπολογιστές και για κινητές συσκευές χάρη στις σύγχρονες δυνατότητες του προγράμματος περιήγησης και των πλαισίων (frameworks) JavaScript.

3.1.2 Λειτουργία των εφαρμογών Ιστού

Οι εφαρμογές ιστού λειτουργούν χρησιμοποιώντας μια αρχιτεκτονική πελάτη-διακομιστή (client-server). Ο πελάτης (client) είναι το πρόγραμμα περιήγησης ιστού με το οποίο αλληλεπιδρά ο χρήστης για να χρησιμοποιήσει την εφαρμογή. Ο διακομιστής (server) είναι ο χώρος όπου αποθηκεύονται και υποβάλλονται σε επεξεργασία η λογική και τα δεδομένα της εφαρμογής. Όταν ένας χρήστης πλοηγείται σε μια διεύθυνση URL εφαρμογής ιστού στο πρόγραμμα περιήγησής του, το πρόγραμμα περιήγησης στέλνει ένα αίτημα στον διακομιστή ιστού για αυτήν την εφαρμογή. Ο διακομιστής ιστού επεξεργάζεται το αίτημα και επιστρέφει μια απάντηση, συνήθως ως σελίδα γλώσσας προγραμματισμού HTML στο πρόγραμμα περιήγησης για εμφάνιση. Καθώς ο χρήστης αλληλεπιδρά με την εφαρμογή ιστού, περισσότερα αιτήματα αποστέλλονται στον διακομιστή. Αυτά τα αιτήματα μπορεί να είναι η ανάκτηση περισσότερων πληροφοριών, η υποβολή δεδομένων ή η ενεργοποίηση ενέργειας από την πλευρά του διακομιστή. Ο διακομιστής μπορεί να ανακτήσει ή να ενημερώσει πληροφορίες σε μια βάση δεδομένων για να εκπληρώσει το αίτημα και να στείλει μια απάντηση μόλις ολοκληρωθεί. Το πρόγραμμα περιήγησης και ο διακομιστής ιστού επικοινωνούν χρησιμοποιώντας το πρωτόκολλο HTTP (Πρωτόκολλο μεταφοράς υπερκειμένου). Τα αιτήματα και οι απαντήσεις συμβαίνουν συνεχώς καθώς ο χρήστης περιηγείται στην εφαρμογή.



Εικόνα 3.1: Απεικόνιση μοντέλου πελάτη (client) – διακομιστή (server)(Πηγή: https://darvishdarab.github.io/cs421_f20/docs/readings/client_server/)

Για τη διατήρηση της κατάστασης, οι εφαρμογές ιστού χρησιμοποιούν συχνά περιόδους λειτουργίας. Οι περίοδοι λειτουργίας επιτρέπουν στον διακομιστή να αποθηκεύει πληροφορίες σχετικά με έναν χρήστη καθώς χρησιμοποιούν την εφαρμογή σε πολλαπλά αιτήματα. Αυτό μπορεί να περιλαμβάνει την κατάσταση σύνδεσης, στοιχεία καλαθιού αγορών ή άλλες προτιμήσεις. Τα δεδομένα της περιόδου σύνδεσης είναι προσωρινά και υπάρχουν για την επίσκεψη του χρήστη.

Πρόσθετα στοιχεία όπως διακομιστές εφαρμογών και βάσεις δεδομένων συνεργάζονται με διακομιστές ιστού για τη δημιουργία ισχυρών, επεκτάσιμων εφαρμογών ιστού. Οι διακομιστές εφαρμογών προσθέτουν επιπλέον λογική και υπηρεσίες που απαιτούνται πέρα από τον βασικό χειρισμό του πρωτοκόλλου HTTP. Οι βάσεις δεδομένων παρέχουν μόνιμη και ασφαλή αποθήκευση δεδομένων που παραμένει σε όλες τις περιόδους σύνδεσης των χρηστών.

3.1.3 Τεχνολογίες Εφαρμογών Ιστού Front-End

Οι τεχνολογίες front-end είναι ένα σύνολο τεχνολογιών που χρησιμοποιούνται για την ανάπτυξη της διεπαφής χρήστη ιστοσελίδων και εφαρμογών. Με τις τεχνολογίες front-end, οι προγραμματιστές δημιουργούν τη σχεδίαση και τη δομή μιας εφαρμογής ιστού ή μιας εφαρμογής για κινητά τηλέφωνα.

Το 'μπροστινό' μέρος μιας εφαρμογής ιστού αναφέρεται στον κώδικα από την πλευρά του πελάτη που εκτελείται στο πρόγραμμα περιήγησης και χειρίζεται τη διεπαφή χρήστη και τις αλληλεπιδράσεις με τον χρήστη. Μερικές από τις κρίσιμες τεχνολογίες front-end που χρησιμοποιούνται για την ανάπτυξη εφαρμογών ιστού

περιλαμβάνουν τις γλώσσες προγραμματισμού HTML, CSS και JavaScript. Αναλυτικότερα:

- Η HTML (HyperText Markup Language) παρέχει τη βασική δομή και το περιεχόμενο στις ιστοσελίδες. Τα στοιχεία HTML ορίζουν επικεφαλίδες, παραγράφους, εικόνες, συνδέσμους, φόρμες και άλλα μπλοκ περιεχομένου.
- Η CSS ελέγχει το οπτικό στυλ και τη διάταξη για τα στοιχεία της HTML. Με τη CSS (Cascading Style Sheets) είναι δυνατόν να προσθέτεις χρώμα, να αλλάζεις γραμματοσειρές, να προσαρμόζεις τα κενά και να δημιουργείς σχέδια με απόκριση για διάφορα μεγέθη συσκευών.
- Η JavaScript (JS) εκτελείται στο πρόγραμμα περιήγησης και επιτρέπει δυναμικές αλληλεπιδράσεις με τους χρήστες. Η JS μπορεί να ενημερώνει την HTML και τη CSS αμέσως, να επικυρώνει φόρμες, να ανιχνεύει ρυθμίσεις συσκευών, να καλεί API, να αποθηκεύει δεδομένα από την πλευρά του πελάτη και να παρέχει σύνθετες διεπαφές χρήστη.

Τα πλαίσια (frameworks) frontend είναι συλλογές προγραμμένου κώδικα και εργαλείων που βοηθούν τους προγραμματιστές να δημιουργήσουν τη διεπαφή χρήστη ενός ιστότοπου ή μιας εφαρμογής ιστού. Αυτά τα πλαίσια παρέχουν μια δομημένη προσέγγιση για την ανάπτυξη εφαρμογών ιστού, προσφέροντας ένα σύνολο επαναχρησιμοποιήσιμων στοιχείων, βιβλιοθηκών και βοηθητικών προγραμμάτων για τον εξορθολογισμό της διαδικασίας. Πλαίσια (frameworks) όπως το React, το Angular και το Vue έχουν γίνει εξαιρετικά δημοφιλή για την ανάπτυξη εφαρμογών ιστού front-end.

3.1.4 Τεχνολογίες εφαρμογών Ιστού Back-end

Η τεχνολογία Back-end αναφέρεται στις γλώσσες προγραμματισμού που χρησιμοποιούνται στην ανάπτυξη Ιστού για τη δημιουργία και τη διαχείριση ενός ιστότοπου ή μιας εφαρμογής. Οι τεχνολογίες Back-end είναι υπεύθυνες για την εργασία "παρασκηνίου", όπως η αποθήκευση δεδομένων, η διαχείριση του ελέγχου ταυτότητας χρήστη και η παροχή API για εφαρμογές front-end.

Το 'πίσω' μέρος (backend) μιας εφαρμογής ιστού περιλαμβάνει όλα τα στοιχεία από την πλευρά του διακομιστή που τροφοδοτούν την εφαρμογή. Ο διακομιστής ιστού είναι ένα πληροφοριακό σύστημα ικανό να παρέχει περιεχόμενο ιστού στους τελικούς

χρήστες μέσω του Διαδικτύου με τη βοήθεια ενός προγράμματος περιήγησης ιστού. Ο ρόλος των προγραμμάτων περιήγησης ιστού όπως ο Firefox, ο Chrome ή ο Internet Explorer είναι να βρίσκουν τον διακομιστή ιστού στον οποίο βρίσκονται τα δεδομένα του ιστοτοπού. Μόλις το πρόγραμμα περιήγησης 'βρει' τον διακομιστή, διαβάζει το αίτημα και επεξεργάζεται τις πληροφορίες.

Πιο συγκεκριμένα, όταν ο χρήστης αλληλεπιδρά με τη διεπαφή, η αλληλεπίδραση στέλνει ένα αίτημα στο 'πίσω' μέρος της εφαρμογής σε μορφή πρωτοκόλλου HTTP. Το πίσω μέρος της εφαρμογής επεξεργάζεται το αίτημα και επιστρέφει μια απάντηση. Όταν το backend επεξεργάζεται ένα αίτημα, συνήθως αλληλεπιδρά με τα ακόλουθα:

- Διακομιστές βάσεων δεδομένων για ανάκτηση ή τροποποίηση σχετικών δεδομένων
- Μικροϋπηρεσίες που εκτελούν ένα υποσύνολο των εργασιών που ζήτησε ο χρήστης
- API τρίτων για τη συλλογή πρόσθετων πληροφοριών ή την εκτέλεση πρόσθετων λειτουργιών

Το backend χρησιμοποιεί πολλά πρωτόκολλα επικοινωνίας και τεχνολογίες για να ολοκληρώσει ένα αίτημα. Επιπλέον, χειρίζεται χιλιάδες διαφορετικά αιτήματα ταυτόχρονα. Ακόμη, συνδυάζει τεχνικές συγχρονισμού και παραλληλισμού, όπως η διανομή αιτημάτων σε πολλούς διακομιστές, η προσωρινή αποθήκευση και η αντιγραφή δεδομένων.

Μερικές από τις γλώσσες προγραμματισμού που χρησιμοποιούνται για την ανάπτυξη κώδικα στο back-end μέρος μια εφαρμογής ιστού είναι οι παρακάτω:

- PHP - Μια ευρέως χρησιμοποιούμενη γλώσσα δέσμης ενεργειών ανοιχτού κώδικα κατάλληλη για ανάπτυξη ιστού. Ο κώδικας PHP εκτελείται στον διακομιστή για τη δημιουργία δυναμικού περιεχομένου ιστοσελίδας.
- Python - Μια δημοφιλής γλώσσα προγραμματισμού υψηλού επιπέδου που χρησιμοποιείται για τη δημιουργία εφαρμογών ιστού από την πλευρά του διακομιστή. Συνήθως, χρησιμοποιούνται, τα πλαίσια Python όπως το Django και το Flask.
- Java - Μια αντικειμενοστραφής γλώσσα προγραμματισμού σχεδιασμένη για τη δημιουργία ισχυρών και ασφαλών εφαρμογών ιστού. Η Java αξιοποιεί πλαίσια όπως το Spring και το Hibernate.

- JavaScript (Node.js) - Η JavaScript μπορεί επίσης να χρησιμοποιηθεί από την πλευρά του διακομιστή μέσω του Node.js για τη δημιουργία γρήγορων και επεκτάσιμων εφαρμογών δικτύου.
- C# - Μια αντικειμενοστραφής γλώσσα προγραμματισμού που χρησιμοποιείται από το πλαίσιο Asp .NET για την ανάπτυξη εφαρμογών ιστού.

Στην εποχή της ανάπτυξης εφαρμογών ιστού, το backend χρησιμεύει ως η μηχανή που τροφοδοτεί τη λειτουργικότητα και τη λογική πίσω από κάθε επιτυχημένη εφαρμογή ιστού. Η επιλογή των σωστών πλαισίων (frameworks) υποστήριξης είναι ζωτικής σημασίας για τη δημιουργία ισχυρών, επεκτάσιμων και αποτελεσματικών λύσεων ιστού που ανταποκρίνονται στις απαιτήσεις των σύγχρονων χρηστών. Τα πιο σημαντικά πλαίσια backend είναι τα παρακάτω:

- Django - Ένα πλαίσιο ιστού Python υψηλού επιπέδου που επιτρέπει την ταχεία ανάπτυξη και τον 'καθαρό' κώδικα. Ακολουθεί το αρχιτεκτονικό πρότυπο μοντέλο-όψη-πρότυπο (MVT).
- Ruby on Rails - Ένα πλαίσιο web ανοιχτού κώδικα γραμμένο σε γλώσσα προγραμματισμού Ruby που ακολουθεί το μοτίβο MVC. Το Rails δίνει έμφαση στη σύμβαση έναντι της διαμόρφωσης.
- Node.js - Παρέχει ένα περιβάλλον χρόνου εκτέλεσης JavaScript που χρησιμοποιεί ασύγχρονες εισόδους/εξόδους βάσει συμβάντων για τη δημιουργία επεκτάσιμων εφαρμογών ιστού.
- Laravel - Ένα πλαίσιο PHP που χρησιμοποιεί το μοτίβο MVC και περιλαμβάνει δρομολόγηση, έλεγχο ταυτότητας, περιόδους σύνδεσης και περισσότερες δυνατότητες.
- ASP .NET - Ένα πλαίσιο εφαρμογών ιστού που αναπτύχθηκε από τη Microsoft για να υλοποιεί το μοτίβο μοντέλου-προβολής-ελεγκτή (MVC).

3.1.5 Πλεονεκτήματα και μειονεκτήματα διαδικτυακών εφαρμογών

Οι τοπικές εφαρμογές υπολογιστών παραμένουν μέχρι και σήμερα οι πιο δημοφιλείς εφαρμογές και αυτό οφείλεται εν μέρη στο ότι οι τοπικές εφαρμογές ήταν και οι πρώτες εφαρμογές που δημιουργήθηκαν. Με την πάροδο του χρόνου όμως όλο και περισσότερες διαδικτυακές εφαρμογές κάνουν την εμφάνισή τους. Συγκρίνοντας τις δυο κατηγορίες εφαρμογών οι χρήστες θα εντοπίσουν τόσο θετικά όσο και αρνητικά

στοιχεία σε συνάρτηση με τις εργασίες που θέλουν η εφαρμογή τους να πραγματοποιηθεί.

Τα πλεονεκτήματα των εφαρμογών ιστού είναι τα παρακάτω:

1. **Άμεση πρόσβαση από οποιαδήποτε συσκευή:** Οι χρήστες των διαδικτυακών εφαρμογών έχουν άμεση πρόσβαση στις εφαρμογές που θέλουν να χρησιμοποιήσουν από οποιονδήποτε υπολογιστή ή άλλη συσκευή έχει internet χωρίς την εγκατάσταση κάποιου επιπρόσθετου λογισμικού. Η μόνη απαραίτητη εφαρμογή είναι ο περιηγητής διαδικτύου ο οποίος είναι προεγκατεστημένος σε όλα τα λειτουργικά συστήματα ακόμα και στις φορητές συσκευές αλλά και στα κινητά τηλέφωνα. Η ιδιότητα αυτή των διαδικτυακών εφαρμογών είναι ιδιαίτερα σημαντική για εφαρμογές με πολλούς χρήστες που στην περίπτωση της τοπικής εφαρμογής θα έπρεπε να εγκατασταθεί η εφαρμογή σε κάθε ένα υπολογιστή ξεχωριστά.
2. **Δυνατότητα χρήσης ανεξαρτήτως τοποθεσίας:** Ως συνέχεια του παραπάνω οι χρήστες των διαδικτυακών εφαρμογών μπορούν να τις χρησιμοποιούν ακόμα και αν δεν βρίσκονται στον χώρο εργασίας τους. Η δυνατότητα αυτή δίνει ευελιξία στους χρήστες ώστε να χρησιμοποιούν τις εφαρμογές οπουδήποτε αυτοί επιθυμούν επιτρέποντας τους ακόμα και να εργάζονται από απομακρυσμένες περιοχές.
3. **Συμβατές με όλα τα λειτουργικά συστήματα:** Ένα ακόμα πλεονέκτημα των διαδικτυακών εφαρμογών είναι ότι είναι συμβατές με όλα τα λειτουργικά συστήματα. Καθώς η εφαρμογή εκτελείται μέσω του περιηγητή του διαδικτύου και όχι στον υπολογιστή του χρήστη, την κάνει ικανή να εκτελείται σε όλα τα λειτουργικά συστήματα. Η ιδιότητα αυτή οφείλεται επίσης και στην προτυποποίηση των γλωσσών προγραμματισμού τις οποίες χρησιμοποιεί η εφαρμογή.
4. **Δεν καταναλώνουν πόρους:** Ως συνέχεια του παραπάνω και εφόσον οι διαδικτυακές εφαρμογές δεν εκτελούνται στον υπολογιστή του χρήστη δεν καταναλώνουν και πόρους από το σύστημα. Για τον λόγο αυτό οι εφαρμογές διαδικτύου είναι ιδιαίτερα ελαφριές για την υπολογιστική μονάδα.
5. **Δεν καταλαμβάνουν χώρο:** Οι εφαρμογές αυτές δεν καταλαμβάνουν καθόλου ή σχεδόν καθόλου χώρο στον δίσκο του χρήστη αφού το σύνολο της εφαρμογής είναι αποθηκευμένο στον εξυπηρετητή και μόνο κατά την χρήση της εφαρμογής μπορεί να υπάρχει μεταφορά δεδομένων προς την

υπολογιστική μονάδα του χρήστη στην περίπτωση που ο χρήστης το επιθυμεί.

6. **Γρήγορη αναβάθμιση:** Σημαντικό πλεονέκτημα συγκριτικά με τις τοπικές εφαρμογές, εμφανίζεται στις περιπτώσεις που η εφαρμογή χρειάζεται κάποια αναβάθμιση. Σε μια κλασική τοπική εφαρμογή η αναβάθμιση του συστήματος θα πρέπει να γίνει σε κάθε ένα υπολογιστή ξεχωριστά πράγμα που απαιτεί χρόνο και χρήμα. Αντίθετα σε μια διαδικτυακή εφαρμογή η αναβάθμιση πραγματοποιείται μόνο στον εξυπηρετητή που φιλοξενεί την εφαρμογή και ταυτόχρονα το αναβαθμισμένο πρόγραμμα είναι διαθέσιμο σε όλους τους χρήστες. Με τον τρόπο αυτό εξοικονομείται χρόνος ο οποίος είναι ιδιαίτερα πολύτιμος κυρίως για τις μεγάλες επιχειρήσεις ενώ ως συνέπεια του παραπάνω σημαντικά μειωμένο είναι και το κόστος της αναβάθμισης μιας και απαιτείται λιγότερο εργατικό δυναμικό για την διεκπεραίωση της αναβάθμισης.
7. **Νέο βελτιωμένο περιβάλλον:** Ένα ακόμα πλεονέκτημα των διαδικτυακών εφαρμογών είναι ότι πλέον με την εμφάνιση της HTML5 είναι δυνατόν ο δημιουργός της εφαρμογής να την εμπλουτίσει έτσι ώστε να είναι πιο φιλική, εύχρηστη και ευχάριστη προς τον χρήστη με εύκολο τρόπο. Παλαιότερα, οι εφαρμογές αυτές υστερούσαν στην εμφάνιση ωστόσο πλέον είναι ιδιαίτερα εύκολα να εμπλουτιστούν.
8. **Δυνατότητα χρήσης και εκτός διαδικτύου - ενδοδικτύου:** Ένα ακόμα πλεονέκτημα των σύγχρονων διαδικτυακών εφαρμογών (εφαρμογές με χρήση HTML5) είναι η δυνατότητα της εκτός διαδικτύου χρήσης μιας διαδικτυακής εφαρμογής με την προϋπόθεση ότι η εφαρμογή έχει κατασκευαστεί με ανάλογο τρόπο. Για παράδειγμα, αν για κάποιο λόγο η σύνδεση στο διαδίκτυο διακοπεί αυτό δεν επηρεάζει τον χρήστη, ο οποίος συνεχίζει να χρησιμοποιεί την εφαρμογή κανονικά. Αυτό επιτυγχάνεται από τον περιηγητή ο οποίος κρατάει ένα αντίγραφο από τα αρχεία, τα οποία είναι απαραίτητα για την εκτός δικτύου χρήση της εφαρμογής, στον υπολογιστή του χρήστη, και τα χρησιμοποιεί όταν αυτό κριθεί απαραίτητο. Η συγκεκριμένη δυνατότητα δεν είναι διαθέσιμη σε όλες τις εφαρμογές που χρησιμοποιούν HTML5 αλλά μόνο σε αυτές που έχει υπάρξει πρόβλεψη για χρήση της εφαρμογής και εκτός διαδικτύου ή ενδοδικτύου.

Τα μειονεκτήματα των εφαρμογών ιστού είναι τα παρακάτω:

1. **Μη πλήρης συμβατότητα των περιηγητών:** Ένα ακόμα μειονέκτημα που αφορά στην τελευταία έκδοση της HTML είναι η μη πλήρης συμβατότητα των περιηγητών με την έκδοση αυτή. Αν και τα πλεονεκτήματα και οι δυνατότητες της HTML5 είναι πολλά, αρκετοί από τους περιηγητές δεν είναι ακόμα πλήρως συμβατοί με αυτά. Έτσι, δεν γίνεται πλήρης χρήση των δυνατοτήτων αυτών, πράγμα που περιορίζει τους προγραμματιστές που έχουν αναλάβει ένα έργο. Επίσης, σε περίπτωση που δεν έχει προβλεφθεί η μη λειτουργία κάποιου χαρακτηριστικού της εφαρμογής σε κάποιον περιηγητή, αυτό μπορεί να δημιουργήσει προβλήματα στην εφαρμογή με αποτέλεσμα να μην λειτουργεί σωστά ή να μην λειτουργεί καθόλου. Για το λόγο αυτό ο κατασκευαστής της εφαρμογής με τον πελάτη πρέπει από κοινού να αποφασίζουν ποιος περιηγητής θα είναι ο προτεινόμενος για την εφαρμογή αλλά ταυτόχρονα να προβλέπεται και η περίπτωση χρήσης άλλων περιηγητών. Ένας καλός τρόπος για να ελεγχθεί η συμβατότητα του περιηγητή με την HTML5 είναι τα διάφορα διαδικτυακά τεστ που αξιολογούν τις δυνατότητές του.
2. **Άμεση αναβάθμιση:** Ένα ακόμα χαρακτηριστικό παράδειγμα πλεονεκτήματος και μειονεκτήματος ταυτόχρονα, αποτελεί και η αναβάθμιση της εφαρμογής. Στην περίπτωση της τοπικής εφαρμογής μια επιχείρηση μπορεί να αναβαθμίσει την εφαρμογή που χρησιμοποιεί όποτε αυτή το κρίνει αναγκαίο κρίνοντας το κόστος αναβάθμισης, την αξιοπιστία της νέας εφαρμογής αλλά και το χρόνο που θα χρειαστούν οι υπάλληλοι της ώστε να προσαρμοστούν στην νέα έκδοση. Αντίθετα στις διαδικτυακές εφαρμογές η αναβάθμιση γίνεται χωρίς πρώτα να ερωτηθούν όλοι οι χρήστες. Για παράδειγμα στην περίπτωση που η ερχόμενη αναβάθμιση μιας τοπικής εφαρμογής έχει σφάλματα τότε μπορεί κάποιος χρήστης (εταιρία) να μην πραγματοποιήσει την αναβάθμιση έως ότου διορθωθούν αυτά. Στην περίπτωση όμως της διαδικτυακής εφαρμογής ο χρήστης (εταιρία) δεν μπορεί να αποτρέψει την αναβάθμιση αυτή.

3.2 Αποκριτικές Εφαρμογές Ιστού

Η δημιουργία ενός ανταποκρινόμενου ιστότοπου είναι ένα ουσιαστικό βήμα για τη δυνατότητα παροχής περιεχομένου σε όλους όσους επισκέπτονται τον ιστότοπο από οποιαδήποτε συσκευή. Οι αποκριτικές εφαρμογές ιστού είναι μια από τις μεγαλύτερες τάσεις στην ανάπτυξη εφαρμογών ιστού.

Μια αποκριτική εφαρμογή ιστού είναι μια εφαρμογή που εκτελείται σε οποιαδήποτε συσκευή και ανταποκρίνεται στο μέγεθος της οθόνης στην οποία εμφανίζεται. Θα προσαρμόσει τη διάταξη και το περιεχόμενο με βάση τη συσκευή που χρησιμοποιείται, επιτρέποντας μια εξαιρετική εμπειρία χρήστη ανεξάρτητα από τη συσκευή από την οποία κάποιος έχει πρόσβαση στον ιστότοπο.



Εικόνα 3.2: Απεικόνιση εμφάνιση αποκριτικών εφαρμογών ιστού σε διάφορες κινητές συσκευές (Πηγή: (<https://www.ramotion.com/blog/responsive-web-application-development/>, last visit 31/5/2024))

Η αποκριτική εφαρμογή ιστού παρέχει την ίδια εμπειρία σε όλες τις συσκευές, από επιτραπέζιους υπολογιστές έως φορητές συσκευές. Μια αποκριτική ανάπτυξη εφαρμογών ιστού προσαρμόζεται στη συσκευή στην οποία προβάλλεται, γεγονός που διευκολύνει τους χρήστες να χρησιμοποιούν τον ιστότοπο ανεξάρτητα από τη συσκευή που χρησιμοποιούν. Ένας ιστότοπος που έχει βελτιστοποιηθεί για κινητές συσκευές θα εμφανίζεται σε διαφορετική μορφή από την αρχική του έκδοση. Θα έχει λιγότερα στοιχεία στη σελίδα και θα πρέπει να τοποθετούνται με τέτοιο τρόπο ώστε να είναι εύκολα προσβάσιμα από οπουδήποτε στην οθόνη χωρίς να χρειάζεται να γίνεται κύλιση ή μεγέθυνση ή σμίκρυνση. (<https://www.ramotion.com/blog/responsive-web-application-development/>)

3.2.1 Πλεονεκτήματα των Αποκριτικών Εφαρμογών Ιστού

Μερικά από τα πλεονεκτήματα των **Αποκριτικών Εφαρμογών Ιστού** είναι:

- Καλύτερη εμπειρία χρήστη

Η εμπειρία χρήστη βελτιώνεται λόγω του σχεδιασμού που αποκρίνεται. Οι χρήστες μπορούν εύκολα να έχουν πρόσβαση στις πληροφορίες που επιθυμούν σε οποιαδήποτε συσκευή χρησιμοποιούν. Αυτό είναι δυνατό λόγω της ευέλικτης διάταξης και της αλλαγής του μεγέθους των εικόνων και του περιεχομένου σε ιστοσελίδες με απόκριση.

- Μικρότερος χρόνος φόρτωσης

Ένας ανταποκρινόμενος ιστότοπος 'φορτώνει' γρηγορότερα από άλλους τύπους ιστότοπων, επειδή είναι 'ελαφρύς' σε σύγκριση με άλλες ιστοσελίδες που έχουν 'βαρύ' περιεχόμενο με γραφικά κ.λπ.

- Ευκολότερη βελτιστοποίηση μηχανών αναζήτησης (Search Engine Optimization)

Μια αποκριτική σχεδίαση για τον ιστότοπο διευκολύνει τις μηχανές αναζήτησης να ανιχνεύουν ολόκληρο τον ιστότοπο, καθώς όλο το περιεχόμενο βρίσκεται σε ένα μέρος και είναι εύκολο για τους ανιχνευτές να το διαβάσουν. Αυτό βοηθάει στην καλύτερη κατάταξή σε μηχανές αναζήτησης όπως το Google, το Yahoo, το Bing κ.λπ., κάτι που με τη σειρά του βοηθά στη μεγαλύτερη επισκεψιμότητα του ιστοτόπου.

- Εύκολη Συντήρηση

Οι αποκριτικές εφαρμογές web είναι εύκολο να διατηρηθούν, επειδή είναι προσβάσιμες από οποιαδήποτε συσκευή ή πρόγραμμα περιήγησης χωρίς προβλήματα. Έτσι, μία ενδεχόμενη αλλαγή στον ιστότοπο, δεν θα επηρεάσει άλλες συσκευές, όπως smartphone ή tablet, επειδή προσαρμόζονται αυτόματα ανάλογα με το μέγεθος και την ανάλυσή τους, ώστε να μην δημιουργούν καμία δυσκολία πρόσβασης τους.

- Αυξημένη ασφάλεια

Καθώς πολλοί άνθρωποι χρησιμοποιούν κινητές συσκευές ως το πρώτο σημείο επαφής τους με το Διαδίκτυο σήμερα, είναι σημαντικό κάθε διαδικτυακή δραστηριότητα να είναι ασφαλής. Με τις αποκριτικές εφαρμογές ιστού δεν απαιτούνται εγκαταστάσεις από τους χρήστες προτού αρχίσουν να χρησιμοποιούν τον ιστότοπο.

- Δεν χρειάζεται πολλές εκδόσεις

Οι αποκριτικές τεχνολογίες εφαρμογών ιστού δίνουν τη δυνατότητα δημιουργίας μιας έκδοσης της εφαρμογής που λειτουργεί σε όλες τις συσκευές, από smartphone και tablet έως επιτραπέζιους υπολογιστές και φορητούς υπολογιστές. Οι χρήστες δεν χρειάζεται πλέον να λαμβάνουν διαφορετικές εκδόσεις της εφαρμογής για κάθε τύπο συσκευής, γιατί υπάρχει μόνο μία έκδοση.

3.3 Μοντέλο Ανάπτυξης Εφαρμογών Ιστού MVC (Model – View – Controller)

Η αρχιτεκτονική MVC είναι ένας δημοφιλής τρόπος δημιουργίας εφαρμογών ιστού που δίνει τη δυνατότητα της επεκτασιμότητας, της επαναχρησιμοποίησης και της συντήρησης της εφαρμογής, δυνατότητες που είναι πολύ σημαντικές για την επιτυχία οποιασδήποτε διαδικτυακής εφαρμογής. Το μοτίβο αυτό αναπτύχθηκε στα μέσα της δεκαετίας του 1990 για να βοηθήσει τους προγραμματιστές να δημιουργήσουν εφαρμογές ιστού που δεν ήταν μόνο φιλικές προς το χρήστη αλλά και εύκολο να συντηρηθούν. Αυτό το μοτίβο εξακολουθεί να χρησιμοποιείται μέχρι σήμερα και αποτελεί μοντέλο αρχιτεκτονικής σε πολλά κοινά πλαίσια (frameworks) ιστού όπως για π.χ αυτό του Ruby on Rails, του Laravel και του ASP.NET MVC.

3.3.1 Σύντομη ιστορία του Πλαισίου MVC

Η ιστορία των διαδικτυακών εφαρμογών είναι μακρά, αλλά ξεκίνησε με την εισαγωγή του πρώτου προγράμματος περιήγησης ιστού το 1993. Αυτό ήταν το Netscape Navigator, το οποίο παρείχε στους χρήστες έναν εύκολο τρόπο πρόσβασης σε πληροφορίες στο Διαδίκτυο χωρίς να χρειάζεται να γνωρίζουν πώς να γράφουν κώδικα σε HTML ή οποιαδήποτε άλλη γλώσσα. Καθώς όλο και περισσότεροι άνθρωποι άρχισαν να έχουν πρόσβαση στο διαδίκτυο και να αλληλεπιδρούν στο διαδίκτυο, οι εταιρείες άρχισαν να παρατηρούν ότι χρειαζόνταν νέους τρόπους για να

δημιουργήσουν τους ιστότοπούς τους για να τους κάνουν πιο φιλικούς προς τον χρήστη.

Αυτό οδήγησε στην εισαγωγή του πλαισίου Model-View-Controller (MVC), το οποίο παρείχε έναν απλό τρόπο στους προγραμματιστές ιστού να δημιουργούν πολύπλοκες εφαρμογές. Το πλαίσιο MVC εισήχθη για πρώτη φορά από τον Trygve Reenskaug το 1979 ως ένας τρόπος διαχείρισης των αλληλεπιδράσεων μεταξύ χρηστών και υπολογιστών και εξακολουθεί να χρησιμοποιείται μέχρι σήμερα.

3.3.2 Αρχιτεκτονική MVC

Μία επισκόπηση υψηλού επιπέδου της αρχιτεκτονικής MVC είναι ότι το συγκεκριμένο πλαίσιο διαχωρίζει τη διεπαφή χρήστη (UI), την αποθήκευση δεδομένων και την επιχειρηματική λογική (business logic) σε τρία ξεχωριστά στοιχεία. Αυτό επιτρέπει στους προγραμματιστές να εργάζονται σε κάθε στοιχείο ανεξάρτητα, χωρίς να επηρεάζουν τα άλλα δύο μέρη της εφαρμογής.

Πιο συγκεκριμένα, το Model-View-Controller (MVC) είναι ένα αρχιτεκτονικό μοτίβο που διαχωρίζει μια εφαρμογή σε τρία κύρια λογικά στοιχεία: το μοντέλο (model), την προβολή (view) και τον ελεγκτή (controller). Κάθε ένα από αυτά τα στοιχεία είναι κατασκευασμένο για να χειρίζεται συγκεκριμένες πτυχές ανάπτυξης μιας εφαρμογής.

Μοντέλο (Model)

Το στοιχείο Μοντέλο στο μοτίβο σχεδίασης MVC (Model-View-Controller) αντιπροσωπεύει τα δεδομένα και την επιχειρηματική λογική (business logic) μιας εφαρμογής. Είναι υπεύθυνο για τη διαχείριση των δεδομένων της εφαρμογής, την επεξεργασία επιχειρηματικών κανόνων και την ανταπόκριση σε αιτήματα για πληροφορίες από άλλα στοιχεία, όπως η προβολή και ο ελεγκτής. Επίσης, παρέχει έναν τρόπο πρόσβασης στα δεδομένα από τη διεπαφή χρήστη. Αυτό περιλαμβάνει την αποθήκευση δεδομένων χρήστη, την ανάκτηση δεδομένων από τη βάση δεδομένων και την εκτέλεση άλλων εργασιών που είναι απαραίτητες για τη λειτουργικότητα της εφαρμογής.

Προβολή (View)

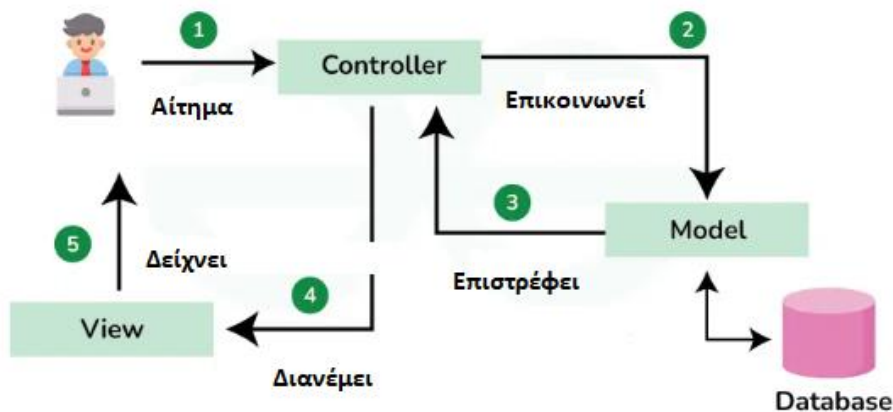
Η προβολή (view) είναι το σημείο όπου βρίσκεται η διεπαφή χρήστη της εφαρμογής. Λαμβάνει τα δεδομένα που παρέχονται από το μοντέλο και τα εμφανίζει στην οθόνη με τρόπο που οι χρήστες μπορούν να αλληλεπιδράσουν. Είναι σημαντικό να σημειωθεί ότι η προβολή δεν περιέχει καμία λογική. Εμφανίζει μόνο δεδομένα, χειρίζεται την εισαγωγή των χρηστών και αποστέλλει τα δεδομένα του χρήστη στον ελεγκτή.

Ελεγκτής (Controller)

Ο ελεγκτής (controller) είναι ένα στοιχείο που βρίσκεται μεταξύ του μοντέλου και της προβολής. Είναι υπεύθυνος για το χειρισμό της αλληλεπίδρασης του χρήστη με την προβολή, καθώς και για την εκτέλεση οποιασδήποτε λογικής απαραίτητης για την προετοιμασία των δεδομένων για εμφάνιση.

Ακολουθεί η λειτουργία του πλαισίου MVC σε μια διαδικτυακή εφαρμογή:

- 1) Ο χρήστης αλληλεπιδρά με την προβολή, όπως π.χ κάνοντας κλικ σε ένα κουμπί ή εισάγοντας κείμενο σε μια φόρμα.
- 2) Η προβολή λαμβάνει την είσοδο του χρήστη και την προωθεί στον ελεγκτή.
- 3) Ο ελεγκτής λαμβάνει την είσοδο χρήστη από την προβολή. Ερμηνεύει την είσοδο, εκτελεί οποιοσδήποτε απαραίτητες λειτουργίες (όπως η ενημέρωση του Μοντέλου) και αποφασίζει πώς θα απαντήσει.
- 4) Ο ελεγκτής ενημερώνει το μοντέλο με βάση την είσοδο χρήστη ή τη λογική της εφαρμογής.
- 5) Εάν αλλάξει το μοντέλο, ειδοποιεί την προβολή.
- 6) Η προβολή ζητά δεδομένα από το μοντέλο για να ενημερώσει την οθόνη του χρήστη.
- 7) Ο ελεγκτής ενημερώνει την προβολή με βάση τις αλλαγές στο Μοντέλο.
- 8) Η προβολή αποδίδει την ενημερωμένη διεπαφή χρήστη με βάση τις αλλαγές που έγιναν από τον ελεγκτή. (<https://www.geeksforgeeks.org/mvc-design-pattern/>)



Εικόνα 3.3: Διαγραμματική απεικόνιση λειτουργίας MVC (Πηγή:
<https://www.geeksforgeeks.org/mvc-design-pattern/>)

3.3.3 Πλεονεκτήματα του μοτίβου σχεδίασης MVC

Η αρχιτεκτονική MVC βοηθά στην οργάνωση της εφαρμογής ιστού, με αποτέλεσμα να γίνεται πιο διαχειρίσιμη. Επιπλέον, διαχωρίζει την επιχειρηματική λογική από τη λογική παρουσίασης, γεγονός που καθιστά πολύ εύκολη την προσθήκη νέων λειτουργιών ή την αλλαγή των υπαρχουσών χωρίς να επηρεάζεται ολόκληρη η εφαρμογή.

Τα πλεονεκτήματα ανάπτυξης εφαρμογών ιστού με το πλαίσιο MVC είναι τα παρακάτω:

‘Καθαρός’ και οργανωμένος κώδικας

Το μοτίβο σχεδίασης MVC βοηθά στο να γραφτεί ‘καθαρός’ και οργανωμένος κώδικας. Ο ελεγκτής, όπως ήδη έχει αναφερθεί, χειρίζεται όλα τα αιτήματα και τις απαντήσεις, και είναι υπεύθυνος για την ανάκτηση δεδομένων από μια βάση δεδομένων ή την εκτέλεση άλλων εργασιών. Αυτό καθιστά πολύ εύκολη τη συντήρηση και την επέκταση της εφαρμογής.

Εύκολο στη δοκιμή και τον εντοπισμό σφαλμάτων

Με το μοτίβο σχεδίασης MVC, είναι εύκολος ο εντοπισμός και η διόρθωση σφαλμάτων της εφαρμογής. Αυτό συμβαίνει επειδή ο ελεγκτής μπορεί εύκολα να απομονωθεί από τα άλλα στοιχεία και να γίνουν δοκιμές που δεν θα επηρεάσουν τα υπόλοιπα στοιχεία του μοτίβου.

Βελτιωμένη παραγωγικότητα και συνεργασία της ομάδας

Το μοτίβο σχεδίασης MVC είναι ένας πολύ καλός τρόπος για την βελτίωση της παραγωγικότητας και της συνεργασίας της ομάδας, επειδή δίνει τη δυνατότητα του διαμοιρασμού των εργασιών μεταξύ διαφορετικών ομάδων, οι οποίες στη συνέχεια μπορούν να εργαστούν παράλληλα. Για παράδειγμα, μια ομάδα θα μπορούσε να επικεντρωθεί στη σύνταξη κώδικα για το στοιχείο μοντέλου και μια άλλη ομάδα θα μπορούσε να φροντίσει για την κωδικοποίηση για το στοιχείο προβολής.

Ωστόσο, υπάρχουν και κάποια μειονεκτήματα ανάπτυξης εφαρμογών ιστού με το πλαίσιο MVC, που είναι τα παρακάτω:

Αυξημένη πολυπλοκότητα

Το κύριο μειονέκτημα του MVC είναι η αυξημένη πολυπλοκότητα. Καθώς η εφαρμογή 'μεγαλώνει', αυξάνεται και ο αριθμός των αρχείων που πρέπει να διατηρηθούν. Αυτό δυσκολεύει τους προγραμματιστές να παρακολουθούν κάθε αρχείο και να διασφαλίζουν ότι είναι πάντα ενημερωμένοι με τις πιο πρόσφατες αλλαγές στη βάση κώδικα.

Στην πράξη αυτό σημαίνει ότι, οι προγραμματιστές θα πρέπει να αφιερώσουν περισσότερο χρόνο πλοηγώντας μεταξύ διαφορετικών αρχείων και προσπαθώντας να βρουν πού χρησιμοποιείται συγκεκριμένος κώδικας.

Απότομη καμπύλη εκμάθησης

Η καμπύλη εκμάθησης για το MVC είναι αρκετά απότομη. Αυτό μπορεί να είναι πρόβλημα εάν προσπαθεί να δημιουργήσει μια εφαρμογή μια μικρή ομάδα ή μία ομάδα που μόλις πρόσφατα άρχισε να χρησιμοποιεί αυτήν την τεχνολογία. Χρειάζεται χρόνος για να μάθουν οι προγραμματιστές πώς λειτουργούν όλα τα διαφορετικά κομμάτια, συμπεριλαμβανομένων της δρομολόγησης, των ελεγκτών και των προβολών.

Πιθανά προβλήματα απόδοσης

Οι εφαρμογές MVC μπορεί να υποφέρουν από προβλήματα απόδοσης όταν δεν έχουν κατασκευαστεί σωστά. Εάν ο κώδικας δεν είναι καλά γραμμένος, μπορεί να προκαλέσει προβλήματα απόδοσης και να επιβραδύνει ολόκληρη την εφαρμογή.

Περιορισμένος έλεγχος στη διεπαφή χρήστη

Το MVC δεν είναι πολύ ευέλικτο όσον αφορά τη διεπαφή χρήστη. Ο έλεγχος είναι πολύ περιορισμένος σχετικά με την εμφάνιση της εφαρμογής.

Μερικά από τα πιο δημοφιλή MVC Frameworks είναι τα παρακάτω:

Ruby on Rails

Το Ruby on Rails είναι ένα πλαίσιο ανάπτυξης ιστού ανοιχτού κώδικα γραμμένο σε γλώσσα προγραμματισμού Ruby που δημιουργήθηκε από τον David Heinemeier Hansson. Παρέχει μια σειρά από χαρακτηριστικά που διευκολύνουν την ανάπτυξη και τη συντήρηση πολύπλοκων εφαρμογών web. Είναι χτισμένο πάνω από την αντικειμενοστραφή γλώσσα προγραμματισμού Ruby και ακολουθεί το αρχιτεκτονικό μοτίβο model–view–controller (MVC).

Django

Το Django είναι πλαίσιο εφαρμογών ιστού ανοιχτού κώδικα, γραμμένο σε Python. Ακολουθεί την αρχιτεκτονική MVC και παρέχει ένα ολοκληρωμένο σύνολο στοιχείων για την ταχεία ανάπτυξη πολύπλοκων ιστότοπων που βασίζονται σε βάσεις δεδομένων. Το Django αναπτύχθηκε από το Ίδρυμα Λογισμικού Django (DSF) και έχει άδεια χρήσης βάσει της άδειας Apache, έκδοση 2.0.

Laravel

Το Laravel είναι ένα πλαίσιο PHP ανοιχτού κώδικα για ανάπτυξη ιστού. Δημιουργήθηκε από τον Taylor Otwell και προοριζόταν για την ανάπτυξη και το σχεδιασμό εφαρμογών ιστού σύμφωνα με το αρχιτεκτονικό μοτίβο model–view–controller (MVC) χρησιμοποιώντας κυρίως τη γλώσσα προγραμματισμού PHP. Η Laravel στοχεύει να κάνει τη διαδικασία ανάπτυξης ευχάριστη για τους προγραμματιστές που τη χρησιμοποιούν.

Spring MVC

Το πλαίσιο Spring MVC είναι ένα πλαίσιο εφαρμογών ιστού Java ανοιχτού κώδικα που παρέχει ένα ισχυρό σύνολο δυνατοτήτων για την ανάπτυξη εφαρμογών MVC εταιρικής ποιότητας. Έχει σχεδιαστεί για να είναι εύκολο στη χρήση, και είναι εξαιρετικά ελεγχόμενο και επεκτάσιμο.

3.4 Αντικειμενοστραφής Προγραμματισμός

Η ιδέα του **αντικειμενοστραφούς προγραμματισμού** ή της αντικειμενοστραφούς σχεδίασης έχει τις ρίζες της σε μία πολλή απλοϊκή ιδέα. Ένα πρόγραμμα περιγράφει "ενέργειες" (επεξεργασία) που εφαρμόζονται πάνω σε δεδομένα. Ένα βασικό ερώτημα που τέθηκε ήταν αν η φιλοσοφία, η δομή του προγράμματος ήταν προτιμότερο να στηρίζεται στις "ενέργειες" ή στα δεδομένα. Η απάντηση σε αυτό το ερώτημα προσδιορίζει και τη βασική διαφορά ανάμεσα στις παραδοσιακές προγραμματιστικές τεχνικές και στην αντικειμενοστραφή προσέγγιση.

Οι αντικειμενοστραφείς γλώσσες προγραμματισμού (Java, Eiffel, Smalltalk και φυσικά C++) δίνουν έμφαση στα δεδομένα παρά στον κώδικα. Το πρόγραμμα αναπτύσσεται γύρω από τα δεδομένα (data-centric) τα οποία ορίζουν από μόνα τους τον τρόπο με τον οποίο διαχειρίζονται (Μαργαρίτης & Τσιομπίκας, 2004). Έτσι, εκλαμβάνει ως πρωτεύοντα δομικά στοιχεία ενός προγράμματος τα δεδομένα, από τα οποία δημιουργούνται με κατάλληλη μορφοποίηση τα **αντικείμενα** (objects). Στα προγράμματα, ένα αντικείμενο είναι η ομαδοποίηση κώδικα και δεδομένων, τα οποία χειριζόμαστε ενιαία. Τα δεδομένα αποτελούν τα χαρακτηριστικά και οι ενέργειες καθορίζουν τη συμπεριφορά ενός αντικειμένου.

Ο αντικειμενοστραφής προγραμματισμός μοντελοποιείται με βάση την παρατήρηση ότι στον πραγματικό κόσμο, τα αντικείμενα αποτελούνται από πολλά μικρότερα αντικείμενα. Αλλά όμως η δυνατότητα συνδυασμού αντικειμένων είναι μόνο μία γενική αρχή του αντικειμενοστραφούς προγραμματισμού. Περιλαμβάνει επίσης αρχές και χαρακτηριστικά που κάνουν τη δημιουργία και τη χρήση αντικειμένων ευκολότερη και πιο ευέλικτη. Το σημαντικότερο από αυτά τα χαρακτηριστικά είναι η **κλάση** (Lemay & Cadenhead, 2003/1999).

Μία κλάση είναι ένα πρότυπο που χρησιμοποιείται για τη δημιουργία πολλαπλών αντικειμένων με παρόμοια χαρακτηριστικά. Οι κλάσεις περιέχουν όλα τα χαρακτηριστικά ενός συγκεκριμένου συνόλου αντικειμένων. Όταν γράφεται ένα πρόγραμμα σε μία αντικειμενοστραφή γλώσσα προγραμματισμού, δεν ορίζονται μεμονωμένα αντικείμενα. Αντί αυτών, ορίζονται κλάσεις αντικειμένων.

Τα πιο σημαντικά χαρακτηριστικά του αντικειμενοστραφούς προγραμματισμού είναι:

- η κληρονομικότητα

Δημιουργείται ένα νέο αντικείμενο παίρνοντας ως βάση ένα άλλο ήδη υπάρχον.

Το νέο αντικείμενο θα έχει τα χαρακτηριστικά του παλιού ενώ θα μπορεί να τα

τροποποιήσει, να τα επεκτείνει και να προσθέσει καινούρια για να καλύψει συγκεκριμένες ανάγκες.

- ο πολυμορφισμός

Αντικείμενα που ανήκουν σε παρόμοιες κλάσεις μπορούν να έχουν κοινό τρόπο προσπέλασης, με αποτέλεσμα ο χρήστης να μπορεί να τα χειριστεί με τον ίδιο τρόπο χωρίς να χρειάζεται να μάθει νέες διαδικασίες.

- η ενθυλάκωση

Οι διαδικασίες κρύβονται από το χρήστη και τα ίδια τα δεδομένα προσδιορίζουν τους τρόπους διαχείρισης τους.

3.5 Βάσεις Δεδομένων

Σε ένα ολοκληρωμένο σύστημα MVC υπάρχει μία βάση δεδομένων για την αποθήκευση και τη διαχείριση της πληροφορίας.

Μια βάση δεδομένων είναι μια οργανωμένη συλλογή δομημένων πληροφοριών, για τις οποίες είναι δυνατή η ανάκτησή τους μέσω της εκτέλεσης ερωτημάτων.

Στη συνέχεια θα ακολουθήσει μία σύντομη περιγραφή της θεωρίας των βάσεων δεδομένων.

3.5.1 Τύποι βάσεων δεδομένων

Υπάρχουν πολλοί διαφορετικοί τύποι βάσεων δεδομένων. Η καλύτερη βάση δεδομένων εξαρτάται από τον τρόπο με τον οποίο σκοπεύει κάποιος να χρησιμοποιήσει τα δεδομένα. Ακολουθεί μία σύντομη περιγραφή των διάφορων ειδών βάσεων δεδομένων.

Σχεσιακές βάσεις δεδομένων

Μια σχεσιακή βάση δεδομένων (relational database) αποτελεί ένα τύπο συστήματος διαχείρισης βάσεων δεδομένων (DBMS) που οργανώνει και αποθηκεύει δεδομένα με δομημένο τρόπο ως ένα σύνολο πινάκων με στήλες (columns) και σειρές (rows) και με προκαθορισμένες σχέσεις μεταξύ τους.

Τα δεδομένα είναι συνήθως δομημένα σε πολλούς πίνακες που μπορούν να ενωθούν μέσω ενός πρωτεύοντος κλειδιού (primary key) ή ενός εξωτερικού κλειδιού (foreign key). Καθώς, τα δεδομένα σε μια σχεσιακή βάση δεδομένων αποθηκεύονται με τρόπο που επιτρέπει τη δημιουργία σχέσεων μεταξύ διαφορετικών συνόλων δεδομένων, την καθιστά ένα πολύ δυνατό εργαλείο για την αποτελεσματική διαχείριση και αναζήτηση δομημένων δεδομένων.

Πιο συγκεκριμένα, οι πίνακες χρησιμοποιούνται για τη διατήρηση πληροφοριών σχετικά με τα στοιχεία που θα αναπαρασταθούν στη βάση δεδομένων. Κάθε σειρά στον πίνακα είναι μια εγγραφή (record) με ένα μοναδικό αναγνωριστικό που ονομάζεται πρωτεύον κλειδί (primary key), και οι σειρές μεταξύ πολλών πινάκων μπορούν να συσχετιστούν χρησιμοποιώντας ξένα κλειδιά (foreign key). Κάθε στήλη στον πίνακα περιέχει ένα συγκεκριμένο τύπο δεδομένων και σε κάθε πεδίο αποθηκεύεται η πραγματική τιμή ενός χαρακτηριστικού.

Το σύνολο των ορισμών των πινάκων καθώς και των περιορισμών και άλλων κανόνων ονομάζεται σχήμα (schema).

Αντικειμενοστραφείς βάσεις δεδομένων

Η αντικειμενοστραφής βάση δεδομένων είναι ένας τύπος συστήματος διαχείρισης βάσεων δεδομένων που αποθηκεύει και ανακτά δεδομένα με τη μορφή αντικειμένων, τα οποία είναι περιπτώσεις κλάσεων ή δομών δεδομένων που ορίζονται σε μια αντικειμενοστραφή γλώσσα προγραμματισμού. Σε αντίθεση με τις παραδοσιακές σχεσιακές βάσεις δεδομένων, οι αντικειμενοστραφείς βάσεις δεδομένων επιτρέπουν την αποθήκευση πολύπλοκων τύπων δεδομένων, όπως εικόνες, έγγραφα και αρχεία πολυμέσων, καθώς και τις σχέσεις μεταξύ αντικειμένων.

Κάθε αντικείμενο στη βάση δεδομένων έχει χαρακτηριστικά και συμπεριφορές που σχετίζονται με αυτό, οι οποίες ορίζονται από την κλάση του. Τα αντικείμενα μπορούν να αποθηκευτούν, να ανακτηθούν και να διαχειριστούν χρησιμοποιώντας αντικειμενοστρεφείς τεχνικές προγραμματισμού, διευκολύνοντας την εργασία με πολύπλοκες δομές δεδομένων.

Οι αντικειμενοστρεφείς βάσεις δεδομένων χρησιμοποιούν αντικειμενοστρεφείς γλώσσες ερωτημάτων, όπως π.χ είναι η γλώσσα Object Query Language (OQL), για την ανάκτηση δεδομένων. Αυτές οι γλώσσες επιτρέπουν πολύπλοκα ερωτήματα που μπορούν να διαπεράσουν ιεραρχίες αντικειμένων και να εκτελέσουν λειτουργίες σε αντικείμενα.

Κατανεμημένες βάσεις δεδομένων

Μια κατανεμημένη βάση δεδομένων αποτελείται από δύο ή περισσότερα αρχεία, που βρίσκονται σε διαφορετικές τοποθεσίες. Η βάση δεδομένων μπορεί να αποθηκευτεί σε πολλαπλούς υπολογιστές, που βρίσκονται στην ίδια φυσική τοποθεσία, ή να διασκορπιστεί σε διαφορετικά δίκτυα.

Αποθήκες δεδομένων

Ένα κεντρικό αποθετήριο δεδομένων, μια αποθήκη δεδομένων, είναι ένας τύπος βάσης δεδομένων που έχει σχεδιαστεί ειδικά για γρήγορη υποβολή ερωτημάτων και ανάλυση.

Βάσεις δεδομένων NoSQL

Μια βάση δεδομένων NoSQL, ή μια μη σχεσιακή βάση δεδομένων, επιτρέπει την αποθήκευση και τον χειρισμό μη δομημένων και ημιδομημένων δεδομένων. Στο σημείο αυτό αξίζει να αναφερθεί ο ορισμός των μη δομημένων ή ημιδομημένων δεδομένων.

Τα μη δομημένα ή ημιδομημένα δεδομένα είναι δεδομένα που δεν είναι οργανωμένα σύμφωνα με ένα προκαθορισμένο μοντέλο ή δομή δεδομένων. Ονομάζονται συχνά ποιοτικά δεδομένα επειδή δεν μπορούν να αναλυθούν ή να υποβληθούν σε επεξεργασία με τους παραδοσιακούς τρόπους χρησιμοποιώντας τις συνήθεις μεθόδους που χρησιμοποιούνται για τα δομημένα δεδομένα. Επίσης, είναι δύσκολο να αναλυθούν και η κατανόηση τους συχνά περιλαμβάνει την εξέταση μεμονωμένων τμημάτων δεδομένων για να διακρίνει κανείς πιθανά χαρακτηριστικά.

Βάσεις δεδομένων γραφήματος

Μια βάση δεδομένων γραφήματος αποθηκεύει δεδομένα με βάση τις οντότητες και τις σχέσεις μεταξύ των οντοτήτων.

Ένα παράδειγμα βάσης δεδομένων γραφήματος είναι η βάση δεδομένων OLTP. Μια OLTP είναι μια γρήγορη βάση δεδομένων analytics που έχει σχεδιαστεί για μεγάλο αριθμό συναλλαγών που εκτελούνται από πολλαπλούς χρήστες.

Αυτές είναι μόνο μερικές από τις δεκάδες μορφές βάσεων δεδομένων που χρησιμοποιούνται σήμερα. Εκτός από τους διάφορους τύπους βάσεων δεδομένων,

οι αλλαγές στις προσεγγίσεις ανάπτυξης της τεχνολογίας και οι έντονες εξελίξεις, όπως το cloud και η αυτοματοποίηση, προωθούν τις βάσεις δεδομένων σε εντελώς νέες κατευθύνσεις. Μερικές από τις πιο πρόσφατες βάσεις δεδομένων περιλαμβάνουν:

Βάσεις δεδομένων ανοιχτού κώδικα

Ένα σύστημα βάσης δεδομένων ανοιχτού κώδικα είναι ένα σύστημα βάσης δεδομένων ανοιχτού κώδικα του οποίου ο πηγαίος κώδικας είναι «ανοιχτός». Αυτές οι βάσεις δεδομένων μπορεί να είναι βάσεις δεδομένων SQL ή NoSQL.

Βάσεις δεδομένων στο Cloud

Μια βάση δεδομένων στο cloud είναι μια συλλογή δεδομένων, είτε δομημένων είτε μη δομημένων, που βρίσκεται σε μια ιδιωτική, δημόσια ή υβριδική υπολογιστική πλατφόρμα cloud. Υπάρχουν δύο τύποι μοντέλων βάσεων δεδομένων στο cloud: το παραδοσιακό και το database as a service (DBaaS).

Βάση δεδομένων πολλαπλών μοντέλων

Οι βάσεις δεδομένων πολλαπλών μοντέλων συνδυάζουν διάφορους τύπους μοντέλων βάσεων δεδομένων σε ένα ενιαίο, ενοποιημένο υπόβαθρο. Αυτό σημαίνει ότι μπορούν να φιλοξενήσουν διάφορους τύπους δεδομένων.

Βάση δεδομένων Document/JSON

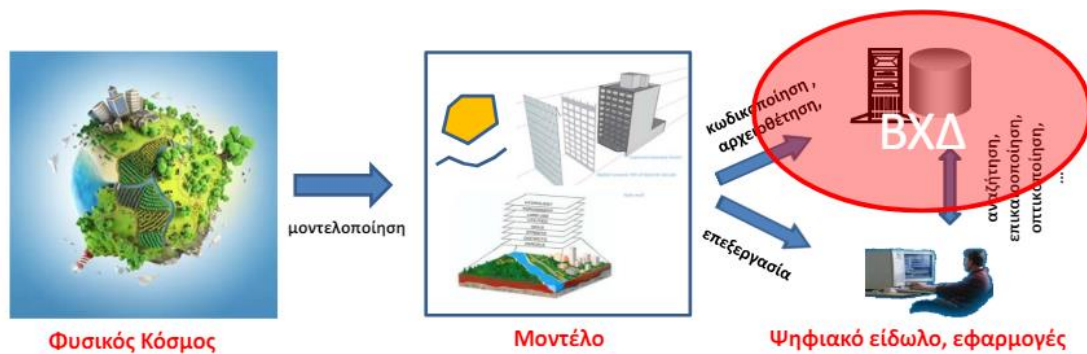
Σχεδιασμένες για την αποθήκευση, την ανάκτηση και τη διαχείριση πληροφοριών με εγγραφοστραφή προσανατολισμό, οι βάσεις δεδομένων εγγράφων είναι ένας σύγχρονος τρόπος αποθήκευσης δεδομένων σε μορφή JSON αντί για γραμμές και στήλες.

Αυτόνομες βάσεις δεδομένων

Ο νεότερος και πιο πρωτοποριακός τύπος βάσης δεδομένων, οι αυτόνομες βάσεις δεδομένων βασίζονται στο cloud και χρησιμοποιούν τη μηχανική μάθηση για αυτοματοποίηση της ρύθμισης, της ασφάλειας, των αντιγράφων ασφαλείας, των ενημερώσεων και άλλων εργασιών διαχείρισης ρουτίνας που εκτελούνται παραδοσιακά από τους διαχειριστές της βάσης δεδομένων.

3.5.2 Σχεδιασμός Βάσεων δεδομένων

Το πρώτο βήμα για την ανάπτυξη ενός συστήματος βάσεων δεδομένων είναι η συλλογή και ανάλυση των απαιτήσεων (requirement collection and analysis). Κατά τη διάρκεια αυτού του βήματος, οι σχεδιαστές της βάσης δεδομένων συζητούν με τους χρήστες για να κατανοήσουν και να καταγράψουν τα είδη των δεδομένων το οποία πρόκειται να καταχωρηθούν στη βάση. Παράλληλα με τις απαιτήσεις σχετικά με τα δεδομένα, καταγράφονται και οι λειτουργικές απαιτήσεις, δηλαδή οι λειτουργίες και οι εφαρμογές που θα υποστηρίζει το σύστημα διαχείρισης βάσεων δεδομένων. Το βήμα αυτό περιλαμβάνει συνεργασία με τους χρήστες και είναι δύσκολο να τυποποιηθεί. Τελικός στόχος του σχεδιασμού ενός συστήματος βάσεων δεδομένων είναι να μοντελοποιηθεί ο πραγματικός κόσμος με την επιλογή των χαρακτηριστικών ενδιαφέροντος, την κωδικοποίησή τους σε δομές δεδομένων (ή κλάσεις) και τη συλλογή – αποθήκευση των δεδομένων σε βάσεις δεδομένων με απώτερο σκοπό τη διατύπωση ερωτημάτων.



Εικόνα 3.4: Απεικόνιση μοντελοποίησης του πραγματικού κόσμου σε βάσεις δεδομένων

Η διαχείριση μιας βάσης δεδομένων γίνεται από ένα ολοκληρωμένο πρόγραμμα λογισμικού βάσης δεδομένων, γνωστό ως σύστημα διαχείρισης βάσεων δεδομένων (DBMS). Ένα DBMS λειτουργεί ως η διασύνδεση μεταξύ της βάσης δεδομένων και των τελικών χρηστών ή των προγραμμάτων της, επιτρέποντας στους χρήστες να ανακτούν, να ενημερώνουν και να διαχειρίζονται τον τρόπο οργάνωσης και βελτιστοποίησης των πληροφοριών. Ένα DBMS διευκολύνει επίσης την εποπτεία και τον έλεγχο των βάσεων δεδομένων, καθιστώντας δυνατές μια ποικιλία διαχειριστικών λειτουργιών, όπως παρακολούθηση απόδοσης, συντονισμό και δημιουργία αντιγράφων ασφαλείας και ανάκτηση.

3.5.3 Εννοιολογικός, Λογικός και Φυσικός σχεδιασμός στις βάσεις δεδομένων

Ο εννοιολογικός σχεδιασμός της βάσης δεδομένων (conceptual database design) αναφέρεται στη δημιουργία του εννοιολογικού σχήματος της βάσης δεδομένων χρησιμοποιώντας ένα υψηλού επιπέδου εννοιολογικό μοντέλο. Το εννοιολογικό σχήμα καθορίζει τις δομές που θα χρησιμοποιηθούν για να καταχωρηθούν τα δεδομένα καθώς και τους περιορισμούς όπως αυτοί έχουν καθοριστεί από την περιγραφή του προβλήματος. Η περιγραφή των δομών και των περιορισμών γίνεται συνήθως χρησιμοποιώντας ως εννοιολογικό μοντέλο το μοντέλο Οντοτήτων-Συσχετίσεων, το οποίο θα περιγραφεί στη συνέχεια. Στόχος του εννοιολογικού σχεδιασμού είναι να δημιουργηθεί μια περιγραφή που να είναι απλή ώστε να είναι κατανοητή από τους χρήστες αλλά να έχει ταυτόχρονα και την απαιτούμενη ακρίβεια ώστε να είναι δυνατή η άμεση μετατροπή της σε ένα μοντέλο υλοποίησης.

Στο σημείο αυτό, θα πρέπει να αποσαφηνιστεί ο όρος οντότητα και ο όρος συσχέτιση. Οντότητα (entity) είναι ένα αντικείμενο ενδιαφέροντος στον πραγματικό κόσμο το οποίο ξεχωρίζει από τα υπόλοιπα. Μια οντότητα λειτουργεί αφαιρετικά σε έναν πολύπλοκο τομέα. Οντότητες μπορεί να είναι άνθρωποι, μέρη, αντικείμενα, γεγονότα, έννοιες κλπ. Στιγμιότυπο (instance) μιας οντότητας είναι μια συγκεκριμένη περίπτωση ενός τύπου οντότητας.

Συσχέτιση είναι η σύνδεση δύο ή περισσότερων τύπων οντοτήτων που παρουσιάζει ενδιαφέρον για σχεδιασμό. Με συσχετίσεις μπορούν να συνδέονται και χαρακτηριστικά οντοτήτων.

Στο επόμενο βήμα, αφού επιλεγεί το ΣΔΒΔ το οποίο θα χρησιμοποιηθεί για την υλοποίηση της βάσης δεδομένων, το εννοιολογικό σχήμα μετασχηματίζεται από το υψηλού επιπέδου εννοιολογικό μοντέλο στο μοντέλο δεδομένων υλοποίησης που υποστηρίζει το ΣΔΒΔ. Το βήμα αυτό καλείται και λογικός σχεδιασμός (logical design) της βάσης δεδομένων. Η απεικόνιση του εννοιολογικού σχήματος στο λογικό σχήμα γίνεται με την εφαρμογή μιας σειράς κανόνων – βημάτων, όπως για π.χ είναι η ανάλυση των σύνθετων ιδιοτήτων σε στοιχειώδεις, η επιλογή του πεδίου που θα είναι το πρωτεύον κλειδί σε μία σχέση, η απεικόνιση κάθε τύπου συσχετίσεων κ.α.

Το λογικό σχεδιασμό ακολουθεί το βήμα του φυσικού σχεδιασμού (physical design) κατά το οποίο προσδιορίζονται οι εσωτερικές δομές αποθήκευσης των δεδομένων, ορίζονται ευρετήρια και καθορίζεται η οργάνωση των αρχείων. Στόχος αυτού του βήματος είναι η καλύτερη απόδοση του συστήματος. Για το σκοπό αυτό, εξετάζονται οι απαιτήσεις από τη λειτουργία του συστήματος, όπως οι αναμενόμενες ερωτήσεις

και τροποποιήσεις των δεδομένων, έτσι ώστε να γίνει πιο αποδοτική η προσπέλαση στα σχετικά δεδομένα.

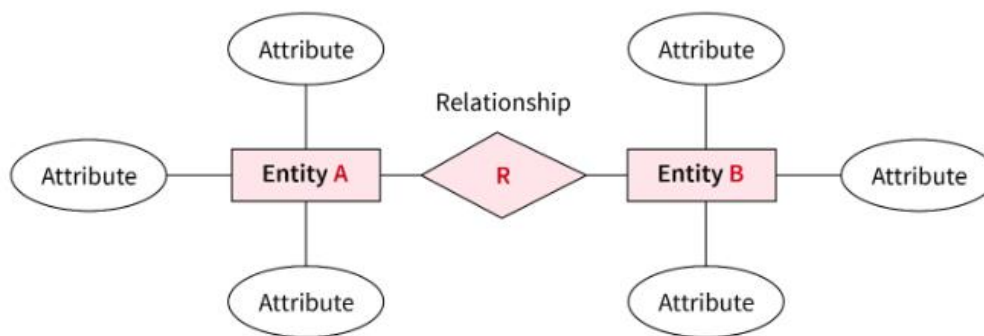
3.5.4 Το μοντέλο Οντοτήτων-Συσχετίσεων

Το μοντέλο Οντοτήτων-Συσχετίσεων ΟΣ (Entity-Relationship ER Model) είναι ένα εννοιολογικό μοντέλο που χρησιμοποιείται για την περιγραφή του σχήματος μιας βάσης δεδομένων. Τα δύο βασικά δομικά στοιχεία του μοντέλου ΟΣ είναι οι οντότητες (entities) και οι συσχετίσεις (relationships). Οι οντότητες μοντελοποιούν αντικείμενα, πρόσωπα ή έννοιες του φυσικού κόσμου, ενώ οι συσχετίσεις μοντελοποιούν σχέσεις μεταξύ οντοτήτων. Τόσο οι οντότητες όσο και οι συσχετίσεις έχουν γνωρίσματα ή χαρακτηριστικά (attributes) που τις περιγράφουν. Τα γνωρίσματα αφορούν στην πληροφορία που θέλουμε να διατηρήσουμε για κάθε μία από τις οντότητες ή τις συσχετίσεις.

Κατά το σχεδιασμό, ορίζονται τους τύπους οντοτήτων και του τύπους συσχετίσεων που αποτελούν το σχήμα της βάσης δεδομένων. Ένας τύπος οντοτήτων (entity type) ορίζει ένα σύνολο οντοτήτων που έχουν τα ίδια γνωρίσματα και περιγράφεται από ένα όνομα και από ένα σύνολο γνωρισμάτων. Όμοια, ένας τύπος συσχέτισης (relationship type) περιγράφει ένα σύνολο συσχετίσεων ανάμεσα σε τύπους οντοτήτων και περιγράφεται επίσης από ένα όνομα, τους σχετιζόμενους τύπους οντοτήτων και τα πιθανά γνωρίσματα. Για ένα τύπο συσχετίσεων, ο λόγος πληθικότητας, δείχνει σε πόσες συσχετίσεις μια οντότητα μπορεί να συμμετέχει. Οι περιπτώσεις είναι:

- 1-1 (ένα προς ένα)
Αντιστοιχίζεται μια οντότητα ενός τύπου με μια οντότητα ενός άλλου τύπου.
- 1-N (ένα προς πολλά)
Αντιστοιχίζεται μια οντότητα ενός τύπου με πολλά στιγμιότυπα ενός άλλου τύπου.
- M-N (πολλά προς πολλά)
Αντιστοιχίζονται πολλά στιγμιότυπα ενός τύπου με πολλά στιγμιότυπα ενός άλλου τύπου

Το μοντέλο ΟΣ είναι ένα γραφικό μοντέλο και το σχήμα μιας βάσης δεδομένων με χρήση του μοντέλου ΟΣ καλείται και διάγραμμα ΟΣ (ER diagram). Στο διάγραμμα ΟΣ, οι τύποι οντοτήτων σχεδιάζονται ως ορθογώνια παραλληλόγραμμα, οι τύποι συσχετίσεων ως ρόμβοι και τα γνωρίσματα ως ελλείψεις.



Εικόνα 3.5: Διαγραμματική απεικόνιση του μοντέλου Οντοτήτων-Συσχετίσεων

3.5.5 Χωρικές βάσεις δεδομένων

Μία χωρική βάση δεδομένων είναι μία κεντρική αποθηκευτική μονάδα για χωρικά δεδομένα. Αποθηκεύει τη γεωμετρία και τις ιδιότητες των χωρικών δεδομένων, καθώς και τις σχέσεις που υπάρχουν μεταξύ τους. Η ανάκτηση της πληροφορίας στην περίπτωση των χωρικών βάσεων δεδομένων γίνεται με την εκτέλεση χωρικών ερωτημάτων.

Οι χωρικές βάσεις δεδομένων ακολουθούν το ενισχυμένο σχεσιακό μοντέλο (Extended Relational Model) ή αντικειμενο-σχεσιακό (object-relational) μοντέλο που διατηρεί τα χαρακτηριστικά του σχεσιακού μοντέλου και το επεκτείνει με αντικειμενοστραφείς έννοιες. Οι βασικές λειτουργίες των χωρικών βάσεων δεδομένων είναι οι παρακάτω:

- Εκτέλεση διαδραστικών ερωτημάτων
- Αναζήτηση αποτελεσμάτων σε συγκεκριμένες περιοχές
- Ανάλυση δεδομένων με βάση την τοποθεσία
- Ανάλυση ροών κίνησης
- Εύρεση κοντινότερων σημείων παρουσίας ή βέλτιστης δρομολόγησης
- Επεξεργασία δεδομένων σε χάρτες

Με τον όρο χωρικά δεδομένα συνήθως λέγονται τα γεωγραφικά ή τα χαρτογραφικά δεδομένα που συσχετίζονται με μια σειρά από περιγραφικά δεδομένα τα οποία και τα χαρακτηρίζουν μοναδικά.

Οι χωρικές βάσεις δεδομένων δίνουν τη δυνατότητα αποθήκευσης τύπων χωρικών δεδομένων. Ο ένας τύπος είναι τα διανυσματικά χωρικά δεδομένα που αναπαριστώνται με τρεις βασικούς τύπους γεωμετριών, τα σημεία, τις γραμμές και τα

πολύγωνα. Ο δεύτερος τύπος είναι η ψηφιδωτή δομή δεδομένων που χρησιμοποιείται σε περιπτώσεις που το χωρικό φαινόμενο που αποτυπώνεται χαρακτηρίζεται ως συνεχής μεταβλητή, όπως π.χ. είναι το υψόμετρο του εδάφους ή η κατανομή του θορύβου.

Αναλυτικότερα, στα διανυσματικά ανήκουν τα γεωμετρικά και τα γεωγραφικά χωρικά δεδομένα. Τα γεωμετρικά αναπαριστούν δεδομένα με βάση προβολές σε καρτεσιανό επίπεδο, όπου οι υπολογισμοί πάνω σε ορισμένες 'γεωμετρίες' πραγματοποιούνται σε συναρτήσεις που εφαρμόζονται στις συντεταγμένες που δηλώνονται στο καρτεσιανό επίπεδο. Ενώ τα γεωγραφικά χωρικά δεδομένα, αναπαριστούν δεδομένα σε μία σφαίρα.

Επιπροσθέτως, οι χωρικές βάσεις δεδομένων υποστηρίζουν τα Γεωγραφικά Πληροφοριακά Συστήματα (ΓΠΣ), τα οποία είναι υπολογιστικά συστήματα και εργαλεία σχεδιασμένα για να υποστηρίζουν τη συλλογή, διαχείριση, επεξεργασία, ανάλυση, μοντελοποίηση και απεικόνιση δεδομένων που αναφέρονται στο χώρο και μπορεί να μεταβάλλονται στο χρόνο.

3.5.6 Χωρικά ερωτήματα

Ένα χωρικό ερώτημα είναι ένας ειδικός τύπος ερωτήματος βάσης δεδομένων που υποστηρίζεται από χωρικές βάσεις δεδομένων και συστήματα γεωγραφικών πληροφοριών (GIS). Αυτός ο τύπος ερωτήματος έχει σχεδιαστεί ειδικά για να λειτουργεί με δεδομένα που σχετίζονται με χωρικά χαρακτηριστικά όπως π.χ είναι οι συντεταγμένες, συγκεκριμένες γεωμετρίες και άλλες λεπτομέρειες με γεωγραφική αναφορά. Τα χωρικά ερωτήματα παρέχουν τη δυνατότητα ειδικής αναζήτησης και ανάλυσης χωρικών δεδομένων με βάση τις χωρικές σχέσεις.

Επιπροσθέτως, ένα χωρικό ερώτημα είναι μια μέθοδος επικοινωνίας που επιτρέπει στον χρήστη να αλληλεπιδρά με γεωχωρικά δεδομένα που είναι αποθηκευμένα σε μια βάση δεδομένων. Η θεμελιώδης λειτουργία ενός χωρικού ερωτήματος είναι να ανακτήσει ένα συγκεκριμένο υποσύνολο χωρικών δεδομένων μέσα από τη βάση δεδομένων με βάση ορισμένα κριτήρια. Αυτά τα κριτήρια μπορούν να οριστούν και να τροποποιηθούν από τον χρήστη και μπορεί να εξαρτώνται από χωρικές σχέσεις όπως π.χ είναι η απόσταση (distance), η τομή (intersection), η γειτνίαση (adjacency) και άλλες.

Τα χωρικά ερωτήματα αποτελούν ζωτικό μέρος οποιουδήποτε συστήματος χωρικής βάσης δεδομένων ή GIS καθώς αποτελούν το κύριο μέσο ανάλυσης και χρήσης των

χωρικών δεδομένων. Το χωρικό ερώτημα είναι ένα ισχυρό εργαλείο και χρησιμοποιείται σε ευρείας κλίμακας εφαρμογές, όπως π.χ είναι ο πολεοδομικός σχεδιασμός, οι περιβαλλοντικές μελέτες, η γεωγραφία, ο σχεδιασμός μεταφορών και πολλές άλλες.

Υπάρχουν κυρίως τρεις τύποι χωρικών ερωτημάτων, πιο συγκεκριμένα:

- Ερωτήματα εγγύτητας (Nearness queries): Αναζητούνται αντικείμενα που βρίσκονται κοντά σε μία καθορισμένη τοποθεσία.
- Ερωτήματα περιοχής (Region queries): Αναζητούνται αντικείμενα που βρίσκονται σε συγκεκριμένες χωρικές περιοχές.
- Ερωτήματα ένωσης / τομής (Union / Intersection queries): Αναζητούνται αντικείμενα τα οποία βρίσκονται σε ενώσεις ή τομές περιοχών.

3.5.7 Γλώσσα προγραμματισμού SQL

Η γλώσσα προγραμματισμού SQL είναι μία γλώσσα υπολογιστών στις βάσεις δεδομένων, που σχεδιάστηκε για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational Database Management System, RDBMS). Η γλώσσα περιλαμβάνει δυνατότητες ανάκτησης και ενημέρωσης δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα.

Η γλώσσα προγραμματισμού SQL υποδιαιρείται σε τέσσερις ξεχωριστές γλώσσες, τη DDL, τη DML, τη DCL και τη TCL.

Η Data Definition Language (DDL) περιλαμβάνει εντολές που χρησιμοποιούνται για τον ορισμό της δομής της βάσης δεδομένων (schema). Μερικά παραδείγματα τέτοιων εντολών είναι η:

- CREATE – χρησιμοποιείται για τη δημιουργία αντικειμένων στη βάση δεδομένων
- ALTER - χρησιμοποιείται για την αλλαγή της δομής
- DROP - χρησιμοποιείται για τη διαγραφή αντικειμένων από τη βάση
- TRUNCATE - διαγράφει όλες τις εγγραφές από έναν πίνακα και ελευθερώνει όλο το χώρο που είχε διατεθεί για αυτές
- COMMENT - προσθέτει σχόλια

Η Data Manipulation Language (DML) περιλαμβάνει εντολές για τη διαχείριση των δεδομένων των αντικειμένων της δομής. Μερικά παραδείγματα τέτοιων εντολών είναι η:

- **SELECT** - ανακτά δεδομένα από τη βάση δεδομένων
- **INSERT** - εισάγει δεδομένα σε κάποιον πίνακα
- **UPDATE** - επικαιροποιεί υπάρχοντα δεδομένα σε πίνακα
- **DELETE** - διαγράφει όλες τις εγγραφές ενός πίνακα, αλλά ο χώρος παραμένει δεσμευμένος

Η Data Control Language (DCL) περιλαμβάνει εντολές όπως είναι η:

- **GRANT** - δίνει στους χρήστες δικαιώματα πρόσβασης στη ΒΔ
- **REVOKE** – αποσύρει τα δικαιώματα πρόσβασης που δόθηκαν με την εντολή GRANT

Η Transaction Control Language (TCL) περιλαμβάνει εντολές που χρησιμοποιούνται για τη διαχείριση (μονιμοποίηση, αναίρεση) αλλαγών που έγιναν με εντολές DML. Επιτρέπει την ομαδοποίηση εντολών προς εκτέλεση, όπως είναι η:

- **COMMIT** - αποθηκεύει ό,τι έχει γίνει
- **SAVEPOINT** - σημειώνει ένα σημείο «συναλλαγής», όπου μπορεί αργότερα να επανέλθει η ροή εντολών με **ROLLBACK**
- **ROLLBACK** - αναιρεί τις αλλαγές που έγιναν από το τελευταίο **COMMIT** και έπειτα

4. Ανάπτυξη Εφαρμογής

Στο κεφάλαιο αυτό θα περιγραφεί ο σχεδιασμός της χωρικής βάσης και τα στάδια μέχρι την ολοκλήρωση της ανάπτυξης της εφαρμογής.

4.1 Εργαλεία Ανάπτυξης της Εφαρμογής

Στο κεφάλαιο αυτό, πρόκειται να περιγραφούν τα εργαλεία ανάπτυξης της εφαρμογής ιστού.

Μοτίβο Σχεδίασης Asp .Net Core MVC

Το βασικό εργαλείο ανάπτυξης της εφαρμογής ήταν το ASP.NET MVC Core. Πιο συγκεκριμένα, είναι ένα πλαίσιο ανάπτυξης εφαρμογών ιστού από τη Microsoft που βασίζεται στην αρχιτεκτονική MVC για τη διευκόλυνση της δημιουργίας εφαρμογών ιστού για επιτραπέζιους υπολογιστές και κινητές συσκευές. Κατά κύριο λόγο, είναι ένα πλαίσιο, 'ελαφρύ' μεταξύ όλων των άλλων τεχνολογιών ανάπτυξης της Microsoft και είναι επίσης ανοιχτού κώδικα.

Οι κύριοι λόγοι ανάπτυξης εφαρμογών ιστού με το πλαίσιο Asp .Net Core MVC είναι ότι:

- Βοηθά στην αξιόπιστη δοκιμή εφαρμογών, οδηγώντας στην εξάλειψη ευπαθειών, σφαλμάτων και δυσλειτουργιών.
- Παρέχει συμβατότητα μεταξύ πλατφορμών, βοηθώντας στην εκτέλεση λογισμικού σε Windows, iOS και Linux.
- Μπορεί να χρησιμοποιηθεί για τη δημιουργία API, τη σύνδεση με συστήματα τρίτων και την εκτέλεση λειτουργιών CRUD (Create, Read, Update, Delete)
- Τέλος, η 'ελαφριά' αρχιτεκτονική βοηθά στην εξοικονόμηση κόστους εγκατάστασης και συντήρησης.

Entity Framework (EF)

Το δεύτερο εργαλείο ανάπτυξης της εφαρμογής ήταν το Entity Framework, το οποίο είναι ένα πλαίσιο αντικείμενο - σχεσιακής σχεδίασης. Αναλυτικότερα, είναι ένα σύνολο τεχνολογιών που υποστηρίζει την ανάπτυξη εφαρμογών λογισμικού προσανατολισμένων στα δεδομένα. Οι προγραμματιστές εφαρμογών που προσανατολίζονται στα δεδομένα πρέπει να επιτύχουν δύο πολύ διαφορετικούς στόχους. Ο πρώτος στόχος είναι να μοντελοποιήσουν τις οντότητες, τις σχέσεις και

τη λογική των προβλημάτων και ο δεύτερος στόχος είναι να συνεργαστούν με τις μηχανές δεδομένων που χρησιμοποιούνται για την αποθήκευση και την ανάκτηση των δεδομένων. Τα δεδομένα μπορούν να εκτείνονται σε πολλαπλά συστήματα αποθήκευσης, το καθένα με τα δικά του πρωτόκολλα. Ακόμη και οι εφαρμογές που λειτουργούν με ένα ενιαίο σύστημα αποθήκευσης πρέπει να εξισορροπούν τις απαιτήσεις του συστήματος αποθήκευσης με τις απαιτήσεις της σύνταξης αποτελεσματικού και συντηρήσιμου κώδικα εφαρμογής.

Γλώσσα προγραμματισμού C#

Τα δύο προαναφερόμενα πλαίσια (frameworks) έχουν αναπτυχθεί στην γλώσσα προγραμματισμού C#, επομένως όλη ανάπτυξη του backend μέρους της εφαρμογής έγινε με τη συγκεκριμένη γλώσσα.

Η C# είναι μια γενικής χρήσης, σύγχρονη και αντικειμενοστραφής γλώσσα προγραμματισμού που προφέρεται ως "C sharp". Αναπτύχθηκε από τη Microsoft με επικεφαλής τον Anders Hejlsberg και την ομάδα του στο πλαίσιο ανάπτυξης της πρωτοβουλίας .Net και εγκρίθηκε από την Ευρωπαϊκή Ένωση Κατασκευαστών Υπολογιστών (ECMA) και τον Διεθνή Οργανισμό Προτύπων (ISO). Η C# μοιάζει πολύ συντακτικά με τη Java και είναι εύκολη η εκμάθησή της για τους χρήστες που έχουν γνώση C, C++ ή Java. Επίσης, είναι μία από τις πιο δημοφιλείς γλώσσες που χρησιμοποιείται για την ανάπτυξη επιτραπέζιων (desktop) εφαρμογών και εφαρμογών ιστού (web). (<https://www.geeksforgeeks.org/introduction-to-c-sharp/>, last visit 4/6/2024)

Γλώσσες προγραμματισμού Frontend

Για τη δημιουργία της δομής της διεπαφής χρήστη, αναπτύχθηκε κώδικας στις γλώσσες προγραμματισμού HTML, CSS και Javascript. Οι γλώσσες αυτές έχουν περιγραφεί στο προηγούμενο κεφάλαιο του τεύχους.

Microsoft Visual Studio

Η ανάπτυξη του κώδικα της εφαρμογής έγινε σε ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment) από τη Microsoft, και συγκεκριμένα στο Microsoft Visual Studio (MVS).

Το MVS χρησιμοποιείται για την ανάπτυξη προγραμμάτων υπολογιστών, καθώς και ιστοσελίδων, εφαρμογών ιστού, υπηρεσιών ιστού και εφαρμογών για κινητά. Το Visual Studio χρησιμοποιεί πλατφόρμες ανάπτυξης λογισμικού της Microsoft όπως το Windows API, τα Windows Forms, το Windows Presentation Foundation,

το Windows Store και το Microsoft Silverlight. Μπορεί να παράγει τόσο εγγενή όσο και διαχειριζόμενο κώδικα. Επιπροσθέτως, υποστηρίζει 36 διαφορετικές γλώσσες προγραμματισμού και επιτρέπει τον εντοπισμό σφαλμάτων σε οποιαδήποτε γλώσσα προγραμματισμού, υπό την προϋπόθεση ότι υπάρχει μια συγκεκριμένη υπηρεσία για τη γλώσσα.

Microsoft Sql Server

Ο SQL Server είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων από τη Microsoft. Το σύστημα έχει σχεδιαστεί και κατασκευαστεί για να διαχειρίζεται και να αποθηκεύει πληροφορίες. Υποστηρίζει διάφορες λειτουργίες επιχειρηματικής ευφυΐας, λειτουργίες ανάλυσης και επεξεργασίας συναλλαγών.

Βιβλιοθήκη Leaflet

Η Leaflet είναι μια μοντέρνα βιβλιοθήκη ανοιχτού κώδικα, γραμμένη σε JavaScript, για τη δημιουργία χαρτών φιλικών για φορητές συσκευές. Αναπτύσσεται από τον Vladimir Agafonkin μαζί με μια ομάδα ανεξάρτητων προγραμματιστών. Η βιβλιοθήκη “ζυγίζει” περίπου 30 KB συμπιεσμένου κώδικα JavaScript, και περιλαμβάνει τα χαρακτηριστικά που οι περισσότεροι προγραμματιστές χρειάζονται για την δημιουργία διαδικτυακών χαρτών.

Η Leaflet είναι σχεδιασμένη με στόχο την απλότητα, τις επιδόσεις και την χρηστικότητα. Λειτουργεί χωρίς προβλήματα σε όλες τις πλατφόρμες, χρησιμοποιώντας την συμβατότητα των λογισμικών πλοήγησης με τις τεχνολογίες HTML5 και CSS3. Μπορεί να επεκταθεί με πρόσθετα, διαθέτει καλογραμμένο και εύκολο στη χρήση πηγαίο κώδικα, διευκολύνοντας όσους θέλουν να συνεισφέρουν. (https://live.osgeo.org/archive/10.5/el/overview/leaflet_overview.html, last visit 15/6/2024)

4.2 Προσέγγιση του εννοιολογικού μοντέλου της χωρικής βάσης δεδομένων

Η εφαρμογή έχει αναπτυχθεί ως βασικό εργαλείο για την καταγραφή της ανασκαφικής διαδικασίας σε πραγματικό χρόνο και ενθαρρύνεται, αν όχι απαιτείται, η δυνατότητα της χρήσης της απευθείας στο πεδίο. Η πληροφορία καταγράφεται με συγκεκριμένο τρόπο, ώστε να εξασφαλίζεται η ομοιογένεια στα στοιχεία που λαμβάνουν υπόψη τυχόν διαφορετικοί ανασκαφείς στον ίδιο αρχαιολογικό χώρο και

να διευκολύνεται η ομαδοποίηση, διαχείριση και τελικά αξιοποίηση των εκάστοτε δεδομένων. Αν και σε αυτό το στάδιο η εφαρμογή δεν δίνει τη δυνατότητα εμφάνισης σε μεμονωμένα κινητά ή σταθερά ευρήματα, καθώς ο αρχικός σχεδιασμός της αποσκοπεί πρωτίστως στην επιτόπου καταγραφή της ανασκαφικής διαδικασίας, εκτιμάται πως περιέχει τα απαραίτητα στοιχεία για τον ειδικό ερευνητή, ώστε να ανασυντεθεί, κατά το δυνατόν, το πλέγμα των διεργασιών που προηγήθηκαν στο πεδίο.

Τέλος, αν και η παραδοσιακή μέθοδος χειρόγραφης καταγραφής σε ανασκαφικό ημερολόγιο φαίνεται ότι μπορεί να υποκατασταθεί από αυτό το ψηφιακό μέσο, συστήνεται η ταυτόχρονη χρήση των δύο για λόγους ασφάλειας αλλά και λειτουργικότητας.

4.2.1 Σχεδιασμός Εννοιολογικού μοντέλου της χωρικής βάσης της εφαρμογής

Με απώτερο σκοπό την οργάνωση, την αποθήκευση και τη διαχείριση της περιγραφικής και χωρικής πληροφορίας μια ανασκαφικής βάσης δεδομένων, στόχος του σχεδιασμού του εννοιολογικού μοντέλου της βάσης ήταν να μοντελοποιηθεί ένα αρχαιολογικό ημερολόγιο χερσαίας ανασκαφής στα κύρια χαρακτηριστικά του, καθώς και στις σχέσεις μεταξύ τους.

Το βήμα αυτό απαιτούσε τη συνεργασία μηχανικού Γεωπληροφορικής, ως ειδικού στην ανάπτυξη τέτοιων πληροφοριακών συστημάτων και ενός Αρχαιολόγου, ως ειδικού στη συγγραφή στο αρχαιολογικό ημερολόγιο της ανασκαφικής διαδικασίας. Ακολουθήθηκε η τεχνική της μοντελοποίησης των κύριων χαρακτηριστικών μιας ανασκαφής, όπως επίσης η μοντελοποίηση εννοιών που ήταν επιθυμητό εκ των προτέρων να έχουν προκαθορισμένες τιμές, έτσι ώστε να ομαδοποιούνται, να διαχειρίζονται και τελικά να αξιοποιούνται καλύτερα τα δεδομένα. Στο σημείο αυτό τα ερωτήματα που θα έπρεπε να απαντηθούν ήταν τα εξής:

- ποιες οντότητες θα περιλαμβάνει η βάση;
- ποιες είναι οι σχέσεις ανάμεσα σε διαφορετικές οντότητες;
- ποιες είναι οι ιδιότητες των οντοτήτων και των σχέσεων;
- ποια είναι τα πεδία τιμών των ιδιοτήτων;
- ποιους περιορισμούς πρέπει να ικανοποιούν οι οντότητες;
- ποια χωρικά ερωτήματα θα πρέπει να 'απαντάει' η βάση δεδομένων;

Πιο συγκεκριμένα, από την πλευρά του Αρχαιολόγου δημιουργήθηκαν σε ένα αρχείο excel τρεις πίνακες, οι οποίοι εμπεριείχαν στοιχεία των τριών βασικών θεματικών της ανασκαφικής διαδικασίας. Η πρώτη θεματική αφορούσε στοιχεία καταγραφής αρχαιολογικής επίχωσης, η δεύτερη θεματική αφορούσε στοιχεία για την καταγραφή αρχιτεκτονικών καταλοίπων και η τρίτη θεματική περιείχε στοιχεία που χαρακτηρίζουν τα ειδικά ευρήματα μιας ανασκαφής.

Αναλυτικότερα, από τον πίνακα που περιείχε στοιχεία αρχαιολογικής επίχωσης προέκυψαν οι κύριες οντότητες **Ανασκαφικά τετράγωνα, Χώροι, Στρώσεις και Ανασκαφικές Ομάδες.**

ΘΕΣΗ:				ΧΩΡΟΣ:				ΑΝΑΣΚΑΦΗ				ΑΡΙΘΜΟΣ ΑΝΑΣΚΑΦΙΚΗΣ			
ΗΜΕΡΟΜΗΝΙΑ:				ΤΕΤΡΑΓΩΝΟ:				ΕΠΙΦΑΝΕΙΑΚΗ ΠΕΡΙΣΣΥΛΟΓΗ				ΕΝΟΤΗΤΑΣ/ΟΜΑΔΑΣ			
ΕΠΙΧΩΣΗ:				Υψόμετρο				ΠΕΡΙΓΡΑΦΗ				πρ.ομάδα			
1. ΧΡΩΜΑ		2. ΣΥΣΤΑΣΗ		3. ΣΥΝΕΚΤΙΚΟΤΗΤΑ		ΑΡΙΘΜΟΣ ΣΤΡΩΜΑΤΟΣ (ΛΑΤΙΝΙΚΗ ΑΡΙΘΜΗΣΗ)									
4Α. ΕΓΚΛΕΙΣΜΑΤΑ		4Β. ΕΙΔΟΣ		4Γ. ΣΥΧΝΟΤΗΤΑ											
ΝΑΙ-ΟΧΙ															
5. ΠΙΘΑΝΟΤΗΤΑ ΔΙΑΤΑΡΑΧΗΣ		ΚΑΜΙΑ		ΧΑΜΗΛΗ		ΜΕΤΡΙΑ		ΥΨΗΛΗ							
ΠΛΕΓΜΑ ΑΝΑΣΚΑΦΙΚΩΝ ΕΝΟΤΗΤΩΝ/ΟΜΑΔΩΝ				ΕΥΡΗΜΑΤΑ				ΑΡΙΘΜΟΙ ΣΑΚΩΝ (ΣΚ) & ΕΙΔΙΚΩΝ ΕΥΡΗΜΑΤΩΝ (ΕΕ): (123-127 και ΕΕ2-ΕΕ4)							
ΑΝΩ				ΧΩΡΙΣ		ΚΕΡΑΜΙΚΗ		ΟΣΤΑ		ΟΣΤΡΕΑ		ΛΑΞΕΥΜΕΝΑ		ΜΕΤΑΛΛΙΚΑ	
ΠΑΡΟΝ				ΚΑΡΒΟΥΝΟ		ΔΕΙΓΜΑ ΧΩΜΑΤΟΣ		ΧΩΜΑ ΠΙΑ ΕΠΙΠΛΕΥΣΗ		ΟΡΓΑΝΙΚΑ		ΑΡΧΙΤΕΚΤΟΝΙΚΑ			
ΚΑΤΩ				ΆΛΛΟ:											
Φωτογραφίες		(αριθμοί)		ΠΑΡΑΤΗΡΗΣΕΙΣ:				Χρονολόγηση:							
Σχέδια		(αριθμοί)						Συντάκτης:							

Εικόνα 6: Απεικόνιση πίνακα με τα βασικά χαρακτηριστικά καταγραφής της αρχαιολογικής επίχωσης

Επίσης, δημιουργήθηκαν και κάποιες δευτερεύουσες οντότητες, που αποτέλεσαν χαρακτηριστικά γνωρίσματα των κύριων οντοτήτων, αφού ήταν αναγκαίο να έχουν προκαθορισμένες τιμές. Αυτές είναι η οντότητα **Είδος Επίχωσης, Κωδικός Χρώματος, Σύστημα Χρώματος, Χρωματικός Τόνος, Σύσταση, Συνεκτικότητα, Έδαφος, Είδος Εγκλεισμάτων, Συχνότητα, Πιθανότητα Διατάραξης, Μέθοδος Περισυλλογής και Είδος Ευρημάτων Ανασκαφικής Ομάδας.**

ΕΠΙΧΩΣΗ: (ΛΥΤ)		ΕΠΙΦΑΝΕΙΑΚΗ ΕΠΕΙΣΑΚΤΗ ΑΡΧΑΙΟΛΟΓΙΚΗ ΔΙΑΤΑΡΑΓΜΕΝΗ		3. ΣΥΝΕΚΤΙΚΟΤΗΤΑ		σκληρή μαλακή χαλαρή σταθερή μέτρια			
1. ΧΡΩΜΑ		Ανοικτό Σκούρο		ερυθρωπάγκρι γκριζωπό καστανό καστανωπό κίτρινο κιτρινωπό κυανό πρασινωπό ερυθρό λευκωπό λευκό κυανωπό		2. ΣΥΣΤΑΣΗ		ιλιώδης αργιλώδης αμμώδης ιλυαργιλώδης ιλυαμμώδης αργιλοιλώδης αργιλοαμμώδης αμμοιλώδης αμμοαργιλώδης άλλο	
Κωδικός		Munsell RGB HEX Άλλο							
4Β. ΕΙΔΟΣ		Λίθοι Άνθρακες Κονιάματα Βότσαλα/Λατύπες Άλλο							

Εικόνα 4.1: Απεικόνιση ενός μέρους των οντοτήτων προκαθορισμένων τιμών

Ο δεύτερος πίνακας αφορούσε στοιχεία για την καταγραφή των αρχιτεκτονικών καταλοίπων. Οι κύριες οντότητες που προέκυψαν από αυτόν τον πίνακα ήταν η οντότητα **Τοίχος** και η οντότητα **Κατασκευή**. Οι δευτερεύουσες οντότητες στην περίπτωση αυτή ήταν η οντότητα **Υλικό Δόμησης**, **Κατάσταση Διατήρησης Τοίχου**, **Κατάσταση Διατήρησης Κατασκευής**, **Κατάσταση Διατήρησης** και **Περίγραμμα**.

ΘΕΣΗ:		ΧΩΡΟΣ:		ΑΡΙΘΜΟΣ ΑΝΑΣΚΑΦΙΚΗΣ	
ΗΜΕΡΟΜΗΝΙΑ:		ΤΕΤΡΑΓΩΝΟ:		ΕΝΟΤΗΤΑΣ/ΟΜΑΔΑΣ	(αρ.ομάδας)
ΤΟΙΧΟΣ	(αρ.)	ΥΨΟΜΕΤΡΟ:		ΣΥΝΤΕΤΑΓΜΕΝΕΣ:	
ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΣ:					
ΔΙΑΣΤΑΣΕΙΣ (Μ.)		ΥΛΙΚΟ ΔΟΜΗΣΗΣ	ΚΑΤΑΣΤΑΣΗ ΔΙΑΤΗΡΗΣΗΣ	ΠΑΡΑΤΗΡΗΣΕΙΣ	
ΜΗΚΟΣ		ΛΙΘΟΙ	ΚΑΛΗ		
ΠΛΑΤΟΣ		ΧΩΜΑ	ΜΕΤΡΙΑ		
ΥΨΟΣ		ΞΥΛΟ	ΚΑΚΗ		
ΧΡΟΝΟΛΟΓΗΣΗ		ΠΗΛΟΣ	ΑΠΟΚΛΙΣΗ		
		ΆΛΛΟ	ΆΛΛΟ		
Φωτογραφίες	(αριθμοί)	Συντάκτης:			
Σχέδια	(αριθμοί)				
ΚΑΤΑΣΚΕΥΗ	(αρ.)	ΥΨΟΜΕΤΡΟ:		ΣΥΝΤΕΤΑΓΜΕΝΕΣ:	
ΠΕΡΙΓΡΑΜΜΑ					
ΤΕΤΡΑΓΩΝΟ	ΟΡΘΟΓΩΝΙΟ	ΚΥΚΛΙΚΟ	ΩΘΕΙΔΕΣ	ΑΚΑΝΟΝΙΣΤΟ	ΆΛΛΟ:
ΔΙΑΣΤΑΣΕΙΣ (Μ.)		ΥΛΙΚΟ ΔΟΜΗΣΗΣ	ΚΑΤΑΣΤΑΣΗ ΔΙΑΤΗΡΗΣΗΣ	ΠΑΡΑΤΗΡΗΣΕΙΣ	
ΜΗΚΟΣ		ΛΙΘΟΙ	ΚΑΛΗ		
ΠΛΑΤΟΣ		ΧΩΜΑ	ΜΕΤΡΙΑ		
ΥΨΟΣ		ΞΥΛΟ	ΚΑΚΗ		
ΔΙΑΜΕΤΡΟΣ		ΠΗΛΟΣ	ΘΡΑΥΣΜΑΤΑ		
ΧΡΟΝΟΛΟΓΗΣΗ		ΆΛΛΟ	ΆΛΛΟ		
Φωτογραφίες	(αριθμοί)	Συντάκτης:			
Σχέδια	(αριθμοί)				

Εικόνα 4.2: Απεικόνιση πίνακα με τα στοιχεία καταγραφής των αρχιτεκτονικών καταλοίπων

Ο τρίτος πίνακας περιείχε στοιχεία που αφορούν την καταγραφή των ειδικών ευρημάτων μιας ανασκαφής. Η κύρια οντότητα που προέκυψε από αυτόν τον πίνακα είναι η οντότητα **Ειδικά Ευρήματα**, ενώ οι δευτερεύουσες είναι το **Είδος Ευρημάτων** και η **Κατάσταση Διατήρησης**.

ΘΕΣΗ:		ΧΩΡΟΣ:		ΑΡΙΘΜΟΣ ΑΝΑΣΚΑΦΙΚΗΣ	
ΗΜΕΡΟΜΗΝΙΑ:		ΤΕΤΡΑΓΩΝΟ:		ΕΝΟΤΗΤΑΣ/ΟΜΑΔΑΣ	(αρ.ομάδας)
Ειδικό Εύρημα	(αρ.)	ΥΨΟΜΕΤΡΟ:		ΣΥΝΤΕΤΑΓΜΕΝΕΣ:	
ΕΙΔΟΣ		ΥΛΙΚΟ		ΠΑΡΑΤΗΡΗΣΕΙΣ	
ΔΙΑΣΤΑΣΕΙΣ (Μ.)		ΚΑΤΑΣΤΑΣΗ ΔΙΑΤΗΡΗΣΗΣ			
ΜΗΚΟΣ		ΠΗΛΟΣ	ΑΚΕΡΑΙΟ		
ΠΛΑΤΟΣ		ΛΙΘΟΣ	ΗΜΙΣΥ		
ΥΨΟΣ		ΜΕΤΑΛΛΟ	ΕΝΑ ΤΡΙΤΟ		
ΧΡΟΝΟΛΟΓΗΣΗ		ΟΡΓΑΝΙΚΟ	ΔΥΟ ΤΡΙΤΑ		
		ΆΛΛΟ	ΆΛΛΟ		
ΠΕΡΙΓΡΑΦΗ					
Φωτογραφίες	(αριθμοί)	Συντάκτης:			
Σχέδια	(αριθμοί)				

Εικόνα 4.3: Απεικόνιση πίνακα με τα στοιχεία καταγραφής των χαρακτηριστικών των ειδικών ευρημάτων.

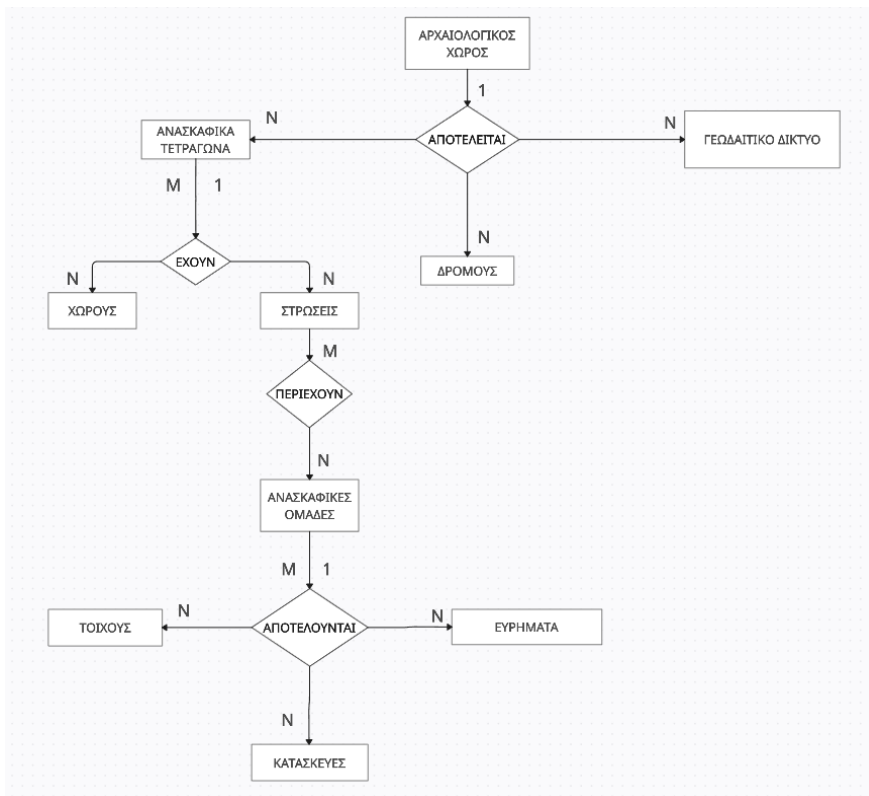
Τέλος, για λόγους πληρότητας δημιουργήθηκαν 2 επιπλέον οντότητες, οι οποίες είναι η οντότητα **Δρόμοι** και η οντότητα **Γεωδαιτικό Δίκτυο**.

4.2.2 Απεικόνιση του σχήματος της βάσης μέσω του διαγράμματος οντοτήτων - συσχετίσεων

Το επόμενο στάδιο του σχεδιασμού του εννοιολογικού μοντέλου ήταν ο καθορισμός των συσχετίσεων μεταξύ των οντοτήτων και η αποτύπωση τους σε ένα διάγραμμα οντοτήτων – συσχετίσεων.

Ο καθορισμός των σχέσεων μεταξύ των οντοτήτων ήταν μία διαδικασία που επαναλήφθηκε αρκετές φορές μέχρι την οριστική διαγραμματική απεικόνιση του σχήματος της βάσης. Βασική επιδίωξη αυτής της διαδικασίας ήταν η όσο το δυνατόν καλύτερη συσχέτιση των οντοτήτων, έτσι ώστε να διευκολυνθεί η διαχείριση και η αξιοποίηση των δεδομένων.

Ακολουθεί το διάγραμμα οντοτήτων – συσχετίσεων (E – R Diagram) του σχήματος της βάσης δεδομένων.



Εικόνα 4.4: Απεικόνιση διαγράμματος οντοτήτων – συσχετίσεων της χωρικής βάσης της εφαρμογής

Στο παραπάνω διάγραμμα επιλέχθηκε να απεικονιστούν οι κύριες οντότητες του σχήματος της βάσης, διότι στην πραγματικότητα πρόκειται για ένα μεγάλο και πολύπλοκο διάγραμμα που αν αποτύπωνε όλες τις οντότητες και τις σχέσεις μεταξύ τους θα ήταν εξαιρετικά δύσκολη η ανάγνωση και η ερμηνεία του από την πλευρά της ειδικότητας των αρχαιολόγων που δεν έχουν το θεωρητικό υπόβαθρο των βάσεων δεδομένων.

Με αυτόν τον τρόπο απεικονίστηκε μία απλή περιγραφή του σχήματος της βάσης, η οποία ταυτόχρονα είχε και την απαιτούμενη ακρίβεια, ώστε να μετατραπεί σε ένα μοντέλο υλοποίησης.

4.3 Σχεδιασμός Λογικού και Φυσικού μοντέλου της χωρικής βάσης της εφαρμογής

Ο λογικός και φυσικός σχεδιασμός της χωρικής βάσης δεδομένων πραγματοποιήθηκε στο Σύστημα Διαχείρισης της βάσης δεδομένων, όπου στην περίπτωση αυτή είναι ο SQL Server.

Αρχικά, δημιουργήθηκαν οι πίνακες αναπτύσσοντας κώδικα στην γλώσσα προγραμματισμού SQL. Με αυτή τη διαδικασία ορίστηκαν τα παρακάτω:

- Οι τύποι των μεταβλητών των πεδίων των πινάκων.
- Τα πεδία τιμών των μεταβλητών των πεδίων των πινάκων.
- Τα πρωτεύοντα και τα δευτερεύοντα κλειδιά των πινάκων για τις συσχετίσεις μεταξύ των οντοτήτων που έχουν λόγο πληθυκότητας 1-N.

```

Db.sql - KATERINA...ADESKPC\User (68)
create table ExcavationUnit
(
  Id int IDENTITY(1,1) PRIMARY KEY,
  Name Varchar(50),
  DateExcavationUnit DateTime,
  WithoutFinding Varchar(10),
  Ceramic Varchar(10),
  Bone Varchar(10),

  Oyster Varchar(10),

  Carved Varchar(10),
  WearingOut Varchar(10),
  Metallic Varchar(10),
  Charcoal Varchar(10),
  SoilSample Varchar(10),
  SoilFloatation Varchar(10),

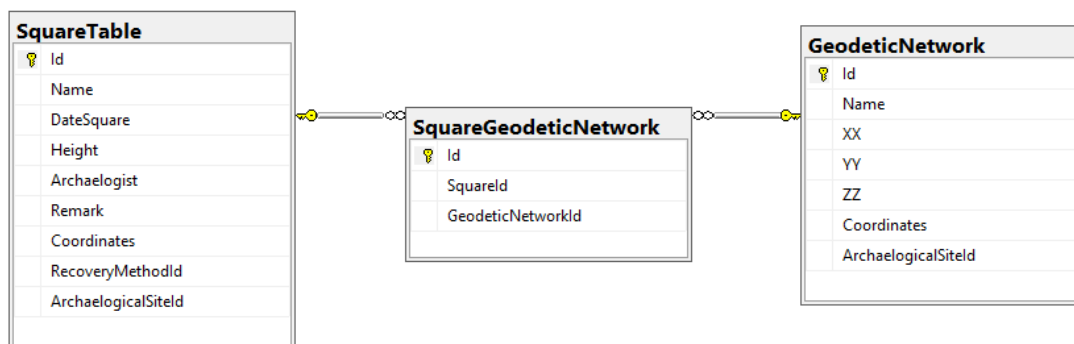
  Organics Varchar(10),
  Architectural Varchar(10),
  Num Varchar(10),
  Description Varchar(50),
  Coordinates Varchar(2000),
  DateFrom Varchar(50),
  Archaeologist Varchar(50),
  LocusId int ,

  CONSTRAINT FK10 FOREIGN KEY (LocusId)
  REFERENCES Locus (Id)
);

```

Εικόνα 4.5: Απεικόνιση δημιουργίας πίνακα στον SQL Server.

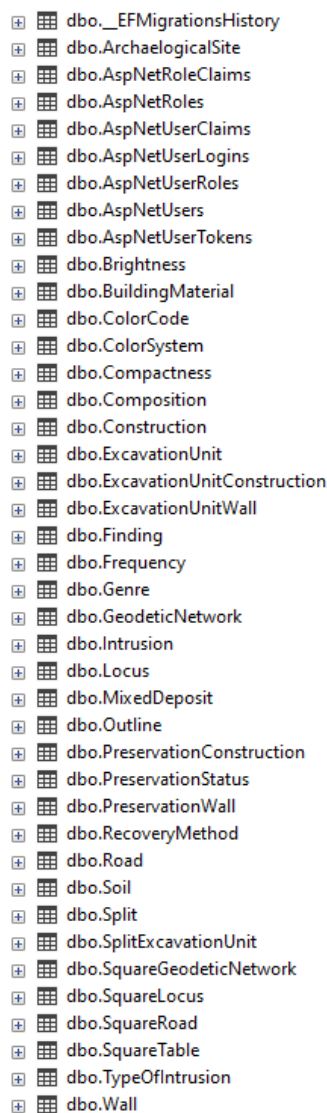
Στις περιπτώσεις των συσχετίσεων μεταξύ των οντοτήτων που ο λόγος πληθυκότητας ήταν M-N, δημιουργήθηκε ένας ενδιάμεσος πίνακας που έχει ως πεδία τα πρωτεύοντα κλειδιά των δύο αρχικών. Αυτή είναι μία πρακτική που γίνεται, για να μετατραπεί η σχέση από M-N σε 1-N μέσω του ενδιάμεσου πίνακα. Ακολουθεί στιγμιότυπο εικόνας τέτοιας περίπτωσης πινάκων.



Εικόνα 4.6: Διαγραμματική απεικόνιση σχέσης πινάκων M-N στον SQL Server.

Κατά τον σχεδιασμό του εννοιολογικού μοντέλου της βάσης, είχε προαποφασιστεί κάποιες οντότητες να έχουν προκαθορισμένες τιμές, έτσι ώστε να υπάρχει ομοιογένεια στα δεδομένα και να διευκολύνεται η ανάκτηση της πληροφορίας τους κατά την εκτέλεση των χωρικών ερωτημάτων. Σε αυτές τις οντότητες ανήκουν η οντότητα **‘Απόχρωση’**, **‘Υλικό Δόμησης’**, **Κωδικός Χρώματος’**, **‘Σύστημα Χρώματος’**, **‘Συνεκτικότητα’**, **‘Σύσταση’**, **‘Συχνότητα’**, **‘Είδος’**, **‘Εγκλεισμα’**, **‘Διατάραξη’**, **‘Περίγραμμα’**, **‘ Κατάσταση Διατήρησης Κατασκευής’**, **‘Κατάσταση Διατήρησης’**, **‘ Κατάσταση Διατήρησης Τοίχου’** και **‘Είδος Εγκλείσματος’**.

Μετά την ολοκλήρωση της παραπάνω διαδικασίας δημιουργήθηκαν οι πίνακες της βάσης στο σύστημα διαχείρισης.

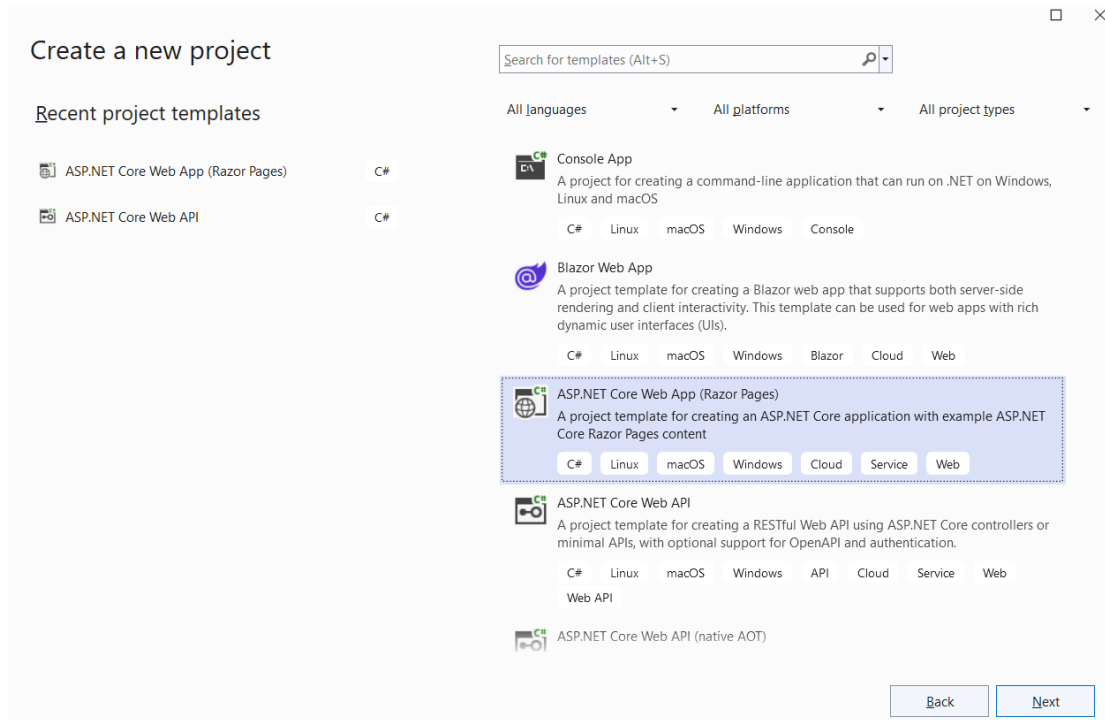


- dbo._EFMigrationsHistory
- dbo.ArchaeologicalSite
- dbo.AspNetRoleClaims
- dbo.AspNetRoles
- dbo.AspNetUserClaims
- dbo.AspNetUserLogins
- dbo.AspNetUserRoles
- dbo.AspNetUsers
- dbo.AspNetUserTokens
- dbo.Brightness
- dbo.BuildingMaterial
- dbo.ColorCode
- dbo.ColorSystem
- dbo.Compactness
- dbo.Composition
- dbo.Construction
- dbo.ExcavationUnit
- dbo.ExcavationUnitConstruction
- dbo.ExcavationUnitWall
- dbo.Finding
- dbo.Frequency
- dbo.Genre
- dbo.GeodeticNetwork
- dbo.Intrusion
- dbo.Locus
- dbo.MixedDeposit
- dbo.Outline
- dbo.PreservationConstruction
- dbo.PreservationStatus
- dbo.PreservationWall
- dbo.RecoveryMethod
- dbo.Road
- dbo.Soil
- dbo.Split
- dbo.SplitExcavationUnit
- dbo.SquareGeodeticNetwork
- dbo.SquareLocus
- dbo.SquareRoad
- dbo.SquareTable
- dbo.TypeOfIntrusion
- dbo.Wall

Εικόνα 4.7: Απεικόνιση πινάκων βάσης δεδομένων στο σύστημα διαχείρισης της βάσης.

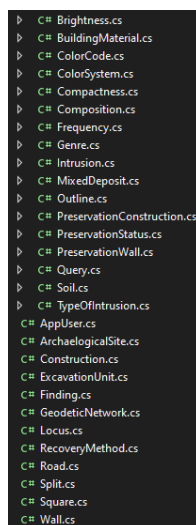
4.4 Δημιουργία περιβάλλοντος εργασίας

Ο κώδικας της εφαρμογής αναπτύχθηκε στο Microsoft Visual Studio, αφού πρώτα είχε δημιουργηθεί το αντίστοιχο project “ASP.NET Core Web App”.



Εικόνα 4.8: Απεικόνιση δημιουργίας project στο Microsoft Visual Studio.

Έπειτα, δημιουργήθηκαν τα μοντέλα – κλάσεις στο Microsoft Visual Studio, τα οποία αντιστοιχήθηκαν με τη βοήθεια του Entity Framework με τους πίνακες της βάσης δεδομένων που είχαν δημιουργηθεί κατά την υλοποίηση του λογικού μοντέλου στο σύστημα διαχείρισης της βάσης δεδομένων.



Εικόνα 4.9: Απεικόνιση μοντέλων (models) στο Microsoft Visual Studio.

Η επικοινωνία μεταξύ του Microsoft Visual Studio με το σύστημα διαχείρισης της βάσης δεδομένων τον SQL Server, έγινε με την παραμετροποίηση ενός "ConnectionString", στο οποίο δηλώθηκε ο τοπικός server του συστήματος διαχείρισης της βάσης δεδομένων.

```
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=KATERINADESKPC;Database=ExcavationGeoDB;Trusted_Connection=True;MultipleActiveResultSets=true;TrustServerCertificate=True;"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

Εικόνα 4.10: Απεικόνιση ConnectionString στο Microsoft Visual Studio.

Με αυτόν τον τρόπο, δημιουργήθηκε το περιβάλλον εργασίας για την ανάπτυξη του κώδικα.

4.5 Περιγραφή λειτουργιών εφαρμογής

Το πληροφοριακό σύστημα που αναπτύχθηκε περιλαμβάνει εκτενέστατη ανάπτυξη λογισμικού που ήταν απαραίτητη για την υλοποίηση του. Ακολουθήθηκε ένα αρθρωτό μοντέλο ανάπτυξης, ενσωματώνοντας και διασυνδέοντας όλες τις επιμέρους ενότητες που υλοποιήθηκαν.

Το πληροφοριακό σύστημα αναλύεται σε έξι (6) βασικές ενότητες, οι οποίες περιλαμβάνουν με τη σειρά τους κάποια επιμέρους υποσυστήματα. Οι ενότητες αυτές είναι:

- Ενότητα 01 (E01): Υποσύστημα Λειτουργιών CRUD
- Ενότητα 02 (E02): Υποσύστημα Χαρτογραφικού υποβάθρου
- Ενότητα 03 (E03): Υποσύστημα Χωρικών Ερωτημάτων
- Ενότητα 04 (E04): Υποσύστημα Αποθήκευσης Πολυμέσων
- Ενότητα 05 (E05): Υποσύστημα Αυθεντικοποίησης Χρήστη
- Ενότητα 06 (E06): Υποσύστημα Διαχείρισης Πρόσβαση

4.5.1 Υποσύστημα Λειτουργιών CRUD (E01)

Το υποσύστημα Λειτουργιών CRUD είναι το σημαντικότερο υποσύστημα της εφαρμογής, διότι επιτρέπει τη διαχείριση πληροφορίας σε μία σχεσιακή βάση δεδομένων από μη εξοικειωμένους χρήστες με τις βάσεις δεδομένων, όπως π.χ είναι οι Αρχαιολόγοι.

Το ακρωνύμιο CRUD σημαίνει δημιουργία (Create), ανάγνωση (Read), ενημέρωση (Update) και διαγραφή (Delete). Αυτές είναι οι τέσσερις βασικές λειτουργίες της αποθήκευσης της πληροφορίας σε μία βάση δεδομένων. Κάθε γράμμα στο ακρωνύμιο μπορεί να αναφέρεται σε όλες τις συναρτήσεις που εκτελούνται σε εφαρμογές σχεσιακής βάσης δεδομένων. Πιο συγκεκριμένα, η λειτουργία:

- CREATE: Εκτελείται με τη συνάρτηση INSERT για τη δημιουργία νέας εγγραφής σε σχεσιακές βάσεις δεδομένων.
- READ: Διαβάζει τις εγγραφές του πίνακα με τη συνάρτηση SELECT με βάση το πρωτεύον κλειδί που σημειώνεται στην παράμετρο εισόδου.
- UPDATE: Εκτελείται με τη συνάρτηση UPDATE στον πίνακα με βάση το καθορισμένο πρωτεύον κλειδί για μια εγγραφή εντός του όρου 'WHERE' της πρότασης.
- DELETE: Διαγράφει με τη συνάρτηση DELETE μια καθορισμένη σειρά στον όρο 'WHERE' της πρότασης.

Μια βασική εφαρμογή CRUD που χρησιμοποιεί αρχιτεκτονική MVC περιλαμβάνει τη δημιουργία ενός μοντέλου για τα δεδομένα, μιας προβολής για τη διεπαφή χρήστη και ενός ελεγκτή για τον χειρισμό των λειτουργιών εισαγωγής δεδομένων από τον χρήστη. Ο χρήστης έχει τη δυνατότητα να δημιουργεί νέες εγγραφές, να διαβάζει τις υπάρχουσες, να τις ενημερώσει και να τις διαγράφει μέσω της διεπαφής, με τον ελεγκτή να χειρίζεται τις αλληλεπιδράσεις μεταξύ της προβολής και του μοντέλου. Αυτή η αρχιτεκτονική παρέχει μια δομημένη και αρθρωτή προσέγγιση στην ανάπτυξη εφαρμογών ιστού, καθιστώντας ευκολότερη τη διαχείριση της εφαρμογής με την πάροδο του χρόνου.

Επιλέχθηκε να δημιουργηθούν λειτουργίες CRUD στις κύριες οντότητες της βάσης δεδομένων της εφαρμογής, οι οποίες είναι η οντότητα '**Αρχαιολογικός Χώρος**', η οντότητα '**Ανασκαφικά Τετράγωνα**', η οντότητα '**Δρόμος**', η οντότητα '**Γεωδαιτικό**

Δίκτυο', η οντότητα 'Χώρος', η οντότητα 'Ανασκαφική Ομάδα', η οντότητα 'Στρώση', η οντότητα 'Ευρήματα', η οντότητα 'Τοίχοι' και η οντότητα 'Κατασκευές'.

Η υλοποίηση του υποσυστήματος προγραμματιστικά έγινε με την εφαρμογή της αρχιτεκτονικής του μοτίβου σχεδίασης Asp.Net Core MVC μέσω της δημιουργίας των κύριων χαρακτηριστικών του μοτίβου, που είναι το μοντέλο (model), ο ελεγκτής (controller), η κλάση Repository και η προβολή (page).

Το πρώτο βήμα ήταν η δημιουργία του μοντέλου. Με αυτόν τον τρόπο ορίζουμε μία κλάση με τα πεδία της, που θα αποτελεί την περιγραφή ενός πίνακα της βάσης δεδομένων και θα δημιουργεί αντικείμενα ίδιου τύπου.

```
namespace ExcavationGeo.Models
{
    61 references
    public class ArchaeologicalSite
    {
        30 references
        public int Id { get; set; }
        13 references
        public string Name { get; set; }
        10 references
        public string Description { get; set; }
        10 references
        public string Coordinates { get; set; }
        9 references
        public double Embadon { get; set; }
        5 references
        public ArchaeologicalSite(int id, string name, string description, string coordinates, double embadon)
        {
            Id = id;
            Name = name;
            Description = description;
            Coordinates = coordinates;
            Embadon = embadon;
        }
        7 references
        public ArchaeologicalSite() { }
    }
}
```

Εικόνα 4.11: Απεικόνιση κώδικα δημιουργίας του μοντέλου *ArchaeologicalSite* της εφαρμογής στο Microsoft Visual Studio

Έπειτα, δημιουργήθηκε ο ελεγκτής (controller) με σκοπό την αρχικοποίηση της κλάσης Repository.

```
namespace ExcavationGeo.Controllers
{
    0 references
    public class ArchaeologicalSiteController : Controller
    {
        0 references
        public IActionResult Index()
        {
            List<ArchaeologicalSite> archaeologicalSites = new List<ArchaeologicalSite>();
            ArchaeologicalSiteRepository archaeologicalSiteRepository = new ArchaeologicalSiteRepository();
            archaeologicalSites = archaeologicalSiteRepository.GeArchaeologicalSites();
            return View(archaeologicalSites);
        }
    }
}
```

Εικόνα 4.12: Απεικόνιση κώδικα δημιουργίας ελεγκτή του μοντέλου *ArchaeologicalSite* της εφαρμογής στο Microsoft Visual Studio.

Ακολούθησε, η δημιουργία της κλάσης Repository, η οποία έχει τον ρόλο να αντλεί δεδομένα από τη βάση δεδομένων, όπως επίσης να ενημερώνει, να διαγράφει και να εισάγει δεδομένα. Στη συγκεκριμένη κλάση, δημιουργήθηκαν οι συναρτήσεις:

- AddSite (). Η συνάρτηση αυτή δημιουργεί ένα νέο αντικείμενο της συγκεκριμένης οντότητας. Ακολουθεί ο κώδικας σε στιγμιότυπο εικόνας.

```
public void AddSite(ArchaeologicalSite pr)
{
    SqlCommand cmd = new SqlCommand("INSERT INTO [ExcavationGeoDB].[dbo].[ArchaeologicalSite] " +
        "(Name ,Description ,Coordinates ,Embadon) values (@Name , @Description , @Coordinates , @Embadon );");

    _connection.Open();

    cmd.Parameters.AddWithValue("@Name", pr.Name);
    cmd.Parameters.AddWithValue("@Description", pr.Description);
    cmd.Parameters.AddWithValue("@Coordinates", pr.Coordinates);
    cmd.Parameters.AddWithValue("@Embadon", pr.Embadon);
    cmd.CommandType = System.Data.CommandType.Text;
    cmd.Connection = _connection;

    Console.WriteLine("Error Generated. Details: " + cmd.CommandText);

    cmd.ExecuteNonQuery();
}
```

Εικόνα 4.13: Απεικόνιση κώδικα συνάρτησης AddSite () της εφαρμογής στο Microsoft Visual Studio

- GetArchaeologicalSite(). Η συνάρτηση αυτή διαβάζει του αρχαιολογικούς χώρους που είναι αποθηκευμένοι στη βάση δεδομένων της εφαρμογής. Ακολουθεί ο κώδικας σε στιγμιότυπο εικόνας.

```
public List<ArchaeologicalSite> GetArchaeologicalSites()
{
    List<ArchaeologicalSite> archaeologicalSites = new List<ArchaeologicalSite>();
    SqlCommand cmd = new SqlCommand("SELECT Id ,Name ,Description ,Coordinates , " +
        "Embadon FROM [ExcavationGeoDB].[dbo].[ArchaeologicalSite]");
    cmd.CommandType = CommandType.Text;
    cmd.Connection = _connection;
    SqlDataAdapter adapter = new SqlDataAdapter(cmd);

    DataTable dt = new DataTable();

    adapter.Fill(dt);

    foreach (DataRow dr in dt.Rows)
    {
        archaeologicalSites.Add(new ArchaeologicalSite(
            Convert.ToInt32(dr["Id"]),
            dr["Name"].ToString(),
            dr["Description"].ToString(),
            dr["Coordinates"].ToString(),
            Convert.ToDouble(dr["Embadon"])
        ));
    }
    return archaeologicalSites;
}
```

Εικόνα 4.14: Απεικόνιση κώδικα συνάρτησης GetArchaeologicalSite() της εφαρμογής στο Microsoft Visual Studio

- DeleteArchaeologicalSite(). Η συνάρτηση αυτή διαγράφει τα αντικείμενα της συγκεκριμένης οντότητας. Ακολουθεί ο κώδικας σε στιγμιότυπο εικόνας.

```

public void DeleteArchaeologicalSite(String id)
{
    SqlDataReader dr;

    SqlCommand cmd1 = new SqlCommand("Delete from [ExcavationGeoDB].[dbo].[ArchaeologicalSite] " +
        "where id = '" + Convert.ToInt32(id) + "';");

    try
    {
        _connection.Open();
        cmd1.CommandType = System.Data.CommandType.Text;
        cmd1.Connection = _connection;

        Console.WriteLine("Error Generated. Details: " + cmd1.CommandText);

        cmd1.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
    }
}

```

Εικόνα 4.15: Απεικόνιση κώδικα συνάρτησης DeleteArchaeologicalSite() της εφαρμογής στο Microsoft Visual Studio

- UpdateArchaeologicalSite (). Η συνάρτηση αυτή ενημερώνει τα αντικείμενα της συγκεκριμένης οντότητας. Ακολουθεί ο κώδικας σε στιγμιότυπο εικόνας.

```

public void UpdateArchaeologicalSite(ArchaeologicalSite pr)
{
    SqlCommand cmd = new SqlCommand("Update [ExcavationGeoDB].[dbo].[ArchaeologicalSite] set Name =@Name , " +
        "Description=@Description , " +
        "Coordinates= @Coordinates ,Embadon=@Embadon where id = '" + Convert.ToInt32(pr.Id) + "';");

    _connection.Open();
    cmd.Parameters.AddWithValue("@Name", pr.Name);
    cmd.Parameters.AddWithValue("@Description", pr.Description);
    cmd.Parameters.AddWithValue("@Coordinates", pr.Coordinates);
    cmd.Parameters.AddWithValue("@Embadon", pr.Embadon);
    cmd.CommandType = System.Data.CommandType.Text;
    cmd.Connection = _connection;

    Console.WriteLine("Error Generated. Details: " + cmd.CommandText);

    cmd.ExecuteNonQuery();
}

```

Εικόνα 4.16: Απεικόνιση κώδικα συνάρτησης UpdateArchaeologicalSite () της εφαρμογής στο Microsoft Visual Studio

Στη συνέχεια, δημιουργήθηκε η διεπαφή χρήστη μέσω της δημιουργίας των προβολών (pages). Οι προβολές εκτός από κώδικα σε γλώσσα προγραμματισμού HTML που αφορά τη μορφοποίησή τους, έχουν και κώδικα που διαχειρίζεται τις

επιλογές του χρήστη (προβολή, εισαγωγή, διαγραφή, ενημέρωση) και αναλόγως καλεί την αντίστοιχη συνάρτηση ή και νέα προβολή - σελίδα, εάν απαιτείται. Ακολουθεί στιγμιότυπο μίας προβολής της οντότητας 'Αρχαιολογικός Χώρος'.

ΚΑΤΑΧΩΡΗΣΗ ΝΕΟΥ ΑΡΧΑΙΟΛΟΓΙΚΟΥ ΧΩΡΟΥ					
ΘΕΣΗ	ΠΕΡΙΓΡΑΦΗ	ΣΥΝΤΕΤΑΓΜΕΝΕΣ Χ(m),Υ(m) στο ΕΓΣΑ'87 (πολύγωνο)	ΕΜΒΑΔΟΝ (m2)		
Αρχαιολογικός χώρος 1	Σύντομη Περιγραφή	614674.7425,4412089.2722,614670.5408,4412121.7798,614688.4320,4412126.9946	1762,328	Επεξεργασία	Διαγραφή
ΑΡΧΑΙΟΛΟΓΙΚΟΣ ΧΩΡΟΣ 2	ΑΡΧΑΙΟΛΟΓΙΚΟΣ ΧΩΡΟΣ 2	610190.76,4417678.70,610824.44,4417428.67,610565.15,4417251.40	200	Επεξεργασία	Διαγραφή

Εικόνα 4.17: Απεικόνιση μίας προβολής της οντότητας 'Αρχαιολογικός Χώρος' στη διεπαφή χρήστη της εφαρμογής.

Η παραπάνω εικόνα περιέχει κάποια στατικά και κάποια δυναμικά στοιχεία της προβολής. Στα στατικά στοιχεία ανήκουν τα πεδία του πίνακα, τα οποία δημιουργήθηκαν με τον παρακάτω κώδικα. Ακολουθεί στιγμιότυπο εικόνας του κώδικα.

```
<table id="table">
  <tr>
    <th>ΘΕΣΗ</th>
    <th>ΠΕΡΙΓΡΑΦΗ</th>
    <th>ΣΥΝΤΕΤΑΓΜΕΝΕΣ Χ(m),Υ(m) στο ΕΓΣΑ'87 (πολύγωνο)</th>
    <th>ΕΜΒΑΔΟΝ (m2)</th>
    <th></th>
    <th></th>
  </tr>
```

Εικόνα 4.18: Απεικόνιση κώδικα δημιουργίας πίνακα προβολής της οντότητας 'Αρχαιολογικός Χώρος' στη διεπαφή χρήστη της εφαρμογής.

Στα δυναμικά στοιχεία ανήκουν τα διάφορα 'κουμπιά' που καλούν άλλες προβολές της οντότητας 'Αρχαιολογικός Χώρος'. Αναλυτικότερα:

1) Το 'κουμπί' ΚΑΤΑΧΩΡΗΣΗ ΝΕΟΥ ΑΡΧΑΙΟΛΟΓΙΚΟΥ ΧΩΡΟΥ, δημιουργείται μέσω του κώδικα:

```
<a asp-page="/CreateArchaeologicalSite" class="btn btn-primary">ΚΑΤΑΧΩΡΗΣΗ ΝΕΟΥ ΑΡΧΑΙΟΛΟΓΙΚΟΥ ΧΩΡΟΥ</a>
```

Στην παραπάνω εντολή καλείται και μία νέα σελίδα – προβολή, η CreateArchaeologicalSite, η οποία είναι μία φόρμα εισαγωγής των αντικειμένων της οντότητας. Ακολουθεί στιγμιότυπο εικόνας του κώδικα που την υλοποιεί.

```

@page
@model ExcavationGeo.Pages.CreateArchaeologicalSiteModel
@{
}

<h1>ΑΡΧΑΙΟΛΟΓΙΚΟΣ ΧΩΡΟΣ</h1>

<form method="post">

  <div class="form-group">
    <label asp-for="ArchaeologicalSite.Id" class="control-label">Id</label>
    <input asp-for="ArchaeologicalSite.Id" class="form-control" pattern="[0-9]*" disabled />
    <span asp-validation-for="ArchaeologicalSite.Id" class="text-danger"></span>
  </div>

  <div class="form-group">
    <label asp-for="ArchaeologicalSite.Name" class="control-label">ΘΕΣΗ </label>
    <input asp-for="ArchaeologicalSite.Name" class="form-control" maxlength="50" required />
    <span asp-validation-for="ArchaeologicalSite.Name" class="text-danger"></span>
  </div>

  <div class="form-group">
    <label asp-for="ArchaeologicalSite.Description" class="control-label">ΠΕΡΙΓΡΑΦΗ</label>
    <input asp-for="ArchaeologicalSite.Description" class="form-control" maxlength="255" required />
    <span asp-validation-for="ArchaeologicalSite.Description" class="text-danger"></span>
  </div>

  <div class="form-group">
    <label asp-for="ArchaeologicalSite.Coordinates" class="control-label">ΣΥΝΤΕΤΑΓΜΕΝΕΣ Χ(μ),Υ(μ) στο ΕΓΣΑ'87 (πολύγωνο)</label>
    <input asp-for="ArchaeologicalSite.Coordinates" class="form-control" pattern="^[0-9,]*$" maxlength="2500" required />
    <span asp-validation-for="ArchaeologicalSite.Coordinates" class="text-danger"></span>
  </div>

  <div class="form-group">
    <label asp-for="ArchaeologicalSite.Embadon" class="control-label">ΕΜΒΑΔΟΝ (m2)</label>
    <input asp-for="ArchaeologicalSite.Embadon" class="form-control" pattern="[0-9,]*" />
    <span asp-validation-for="ArchaeologicalSite.Embadon" class="text-danger"></span>
  </div>

  <button type="submit" class="btn btn-primary">Καταχώρηση</button>
</form>

```

Εικόνα 4.19: Απεικόνιση κώδικα της προβολής CreateArchaeologicalSite της οντότητας 'Αρχαιολογικός Χώρος' στο Microsoft Visual Studio.

Το αποτέλεσμα της εκτέλεσης του παραπάνω κώδικα είναι η παρακάτω διεπαφή χρήστη:

ΑΡΧΑΙΟΛΟΓΙΚΟΣ ΧΩΡΟΣ

id
0

ΘΕΣΗ

ΠΕΡΙΓΡΑΦΗ

ΣΥΝΤΕΤΑΓΜΕΝΕΣ Χ(μ),Υ(μ) στο ΕΓΣΑ'87 (πολύγωνο)

ΕΜΒΑΔΟΝ (m2)
0

Καταχώρηση

Εικόνα 4.20: Απεικόνιση διεπαφής χρήστη καταχώρησης νέων αντικειμένων της οντότητας 'Αρχαιολογικός Χώρος'.

2) Το 'κουμπί' Επεξεργασία, δημιουργείται μέσω του κώδικα:

```

<td>
  <a asp-page="/EditArchaeologicalSite" asp-route-id="@item.Id" class="btn btn-primary">Επεξεργασία</a>
</td>

```

Στην παραπάνω εντολή καλείται και μία νέα σελίδα – προβολή, η EditArchaeologicalSite, η οποία είναι μία φόρμα εισαγωγής επεξεργασίας των

αντικειμένων της οντότητας. Η διαφορά της με την προβολή 'CreateArchaeologicalSite' είναι ότι σε αυτή την περίπτωση εμφανίζεται η φόρμα συγκεκριμένου αντικειμένου συμπληρωμένη, ώστε να τροποποιηθούν οι εγγραφές των πεδίων και να αποθηκευτούν εκ νέου. Αυτό επιτυγχάνεται με τον παρακάτω κώδικα:

```
<h1>ΑΡΧΑΙΟΛΟΓΙΚΟΣ ΧΩΡΟΣ</h1>

<form method="post">
  <input type="hidden" asp-for="ArchaeologicalSite.Id" />

  <div class="form-group">
    <label asp-for="ArchaeologicalSite.Id" class="control-label"></label>
    <input asp-for="ArchaeologicalSite.Id" disabled class="form-control" pattern="[0-9]*" />
    <span asp-validation-for="ArchaeologicalSite.Id" class="text-danger"></span>
  </div>

  <div class="form-group">
    <label asp-for="ArchaeologicalSite.Name" class="control-label">ΘΕΣΗ</label>
    <input asp-for="ArchaeologicalSite.Name" class="form-control" maxlength="50" required />
    <span asp-validation-for="ArchaeologicalSite.Name" class="text-danger"></span>
  </div>

  <div class="form-group">
    <label asp-for="ArchaeologicalSite.Description" class="control-label">ΠΕΡΙΓΡΑΦΗ</label>
    <input asp-for="ArchaeologicalSite.Description" class="form-control" maxlength="255" required />
    <span asp-validation-for="ArchaeologicalSite.Description" class="text-danger"></span>
  </div>

  <div class="form-group">
    <label asp-for="ArchaeologicalSite.Coordinates" class="control-label">ΣΥΝΤΕΤΑΓΜΕΝΕΣ Χ(μ),Υ(μ) στο ΕΓΣΑ'87 (πολύγωνο)</label>
    <input asp-for="ArchaeologicalSite.Coordinates" class="form-control" pattern="[0-9,.]*$" maxlength="2000" required />
    <span asp-validation-for="ArchaeologicalSite.Coordinates" class="text-danger"></span>
  </div>

  <div class="form-group">
    <label asp-for="ArchaeologicalSite.Embadon" class="control-label">ΕΜΒΑΔΟΝ (m2)</label>
    <input asp-for="ArchaeologicalSite.Embadon" class="form-control" pattern="[0-9]*" required />
  </div>

  <button type="submit" class="btn btn-primary">Save</button>
</form>
```

Εικόνα 4.21: Απεικόνιση κώδικα της προβολής *EditArchaeologicalSite* της οντότητας 'Αρχαιολογικός Χώρος' στο Microsoft Visual Studio.

3) Το κουμπί 'Διαγραφή' δημιουργείται μέσω του κώδικα:

```
<td>
  <a asp-page="/ArchaeologicalSite" onclick="return confirm
  ('Διαγραφή Αρχαιολογικού χώρου');"
  asp-page-handler="Delete" asp-route-id="@item.Id">
    <button type="submit" class="btn btn-danger">Διαγραφή</button>
  </a>
</td>
```

Στη συγκεκριμένη περίπτωση δεν καλείται μία νέα – προβολή σελίδα, αλλά μία συνάρτηση η *DeleteArchaeologicalSite(id)*, η οποία περιγράφηκε παραπάνω.

4.5.2 Υποσύστημα Χαρτογραφικού υποβάθρου (E02)

Το υποσύστημα του χαρτογραφικού υποβάθρου είναι το δεύτερο πιο σημαντικό υποσύστημα του πληροφοριακού συστήματος μετά το υποσύστημα των CRUD λειτουργιών.

Βασικό ζητούμενο σε μία χωρική βάση δεδομένων είναι η οπτικοποίηση της σε κάποιο χαρτογραφικό υπόβαθρο. Αυτό βοηθά τους χρήστες να εξάγουν κάποια πρώτα συμπεράσματα για την πληροφορία της βάσης δεδομένων.

Για τη μορφοποίηση του χάρτη αξιοποιήθηκε η βιβλιοθήκη ανοιχτού κώδικα Leaflet, στην οποία ενσωματώθηκε το ανοιχτού κώδικα έργο χαρτογράφησης World Imagery(Map Server) της ESRI, αποφεύγοντας με αυτόν τον τρόπο τη χρήση ακριβών συνδρομητικών μοντέλων.

```
var mymap = L.map('mapid').setView([39.389, 24.620], 8);

L.tileLayer('https://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}', {
  attribution: 'geo'
}).addTo(mymap);
```

Εικόνα 4.22: Απεικόνιση κώδικα της εισαγωγής χάρτη στο Microsoft Visual Studio.

Στις απαιτήσεις του αρχικού σχεδιασμού της βάσης δεδομένων ήταν η χωρικής της πληροφορία να αποτυπώνεται στο Ελληνικό Γεωδαιτικό Σύστημα Αναφοράς (ΕΓΣΑ 87), έτσι ώστε να επιτυγχάνεται η διασύνδεσή της με άλλα χαρτογραφικά υπόβαθρα του Ελληνικού χώρου. Το γεωδαιτικό σύστημα του επιλεγμένου χαρτογραφικού υποβάθρου της ESRI είναι το Παγκόσμιο Γεωδαιτικό Σύστημα Αναφοράς (WGS84), γι' αυτό προέκυψε η ανάγκη της μετατροπής του ΕΓΣΑ 87 στο WGS84.

Η υλοποίηση αυτής της απαίτησης προγραμματιστικά έγινε με τη δημιουργία μιας συνάρτησης μετατροπής των συντεταγμένων του ΕΓΣΑ 87 σε συντεταγμένες στο WGS84. Ακολουθεί ο κώδικας υλοποίησης:

```
function convertCoords(egsa87Coordinates) {

    proj4.defs("EPSG:2100", "+proj=tmerc +lat_0=0 +lon_0=24 +k=0.9996 +x_0=500000 +y_0=0 +ellps=GRS80 +towgs84=-199.87,74.79,246.62,-0.01,-0.02,-0.03,0 +units=m +no_defs");
    proj4.defs("EPSG:4326", "+proj=longlat +datum=WGS84 +no_defs");

    // Convert EGSA87 to WGS84
    var wgs84Coordinates = proj4("EPSG:2100", "EPSG:4326", egsa87Coordinates);

    return wgs84Coordinates;
}
```

Στην προσπάθεια να είναι πιο ευανάγνωστος ο χάρτης ενσωματώθηκε η λειτουργία 'pop up'. Με τη λειτουργία αυτή, εμφανίζονται αναδυόμενα παράθυρα πάνω στο

χάρτη, τα οποία απεικονίζουν κάποια επιλεγμένη περιγραφική πληροφορία της συγκεκριμένης οντότητας. Ακολουθεί στιγμιότυπο εικόνας του κώδικα υλοποίησης.

```
var ρορυρΗτμλ = '<br>';
ρορυρΗτμλ += '<table>';
ρορυρΗτμλ += '<tbody>';

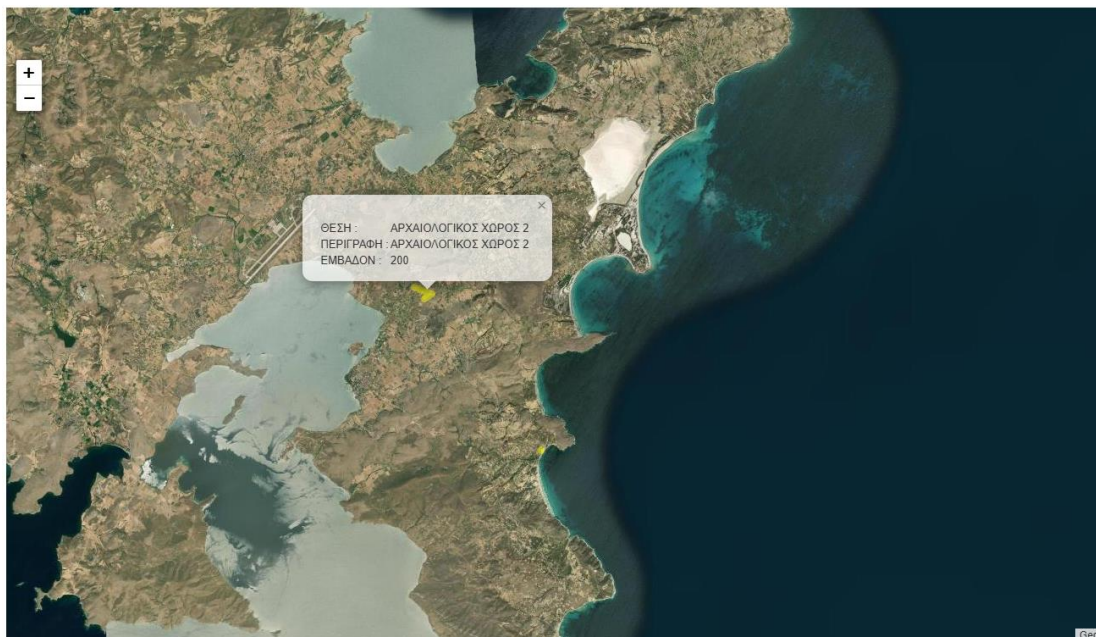
ρορυρΗτμλ += '<tr>';
ρορυρΗτμλ += '<td>ΘΕΣΗ :</td>';
ρορυρΗτμλ += '<td>' + temp.name + '</td>';
ρορυρΗτμλ += '</tr>';

ρορυρΗτμλ += '<tr>';
ρορυρΗτμλ += '<td>ΠΕΡΙΓΡΑΦΗ :</td>';
ρορυρΗτμλ += '<td>' + temp.description + '</td>';
ρορυρΗτμλ += '</tr>';

ρορυρΗτμλ += '<tr>';
ρορυρΗτμλ += '<td>ΕΜΒΑΔΟΝ :</td>';
ρορυρΗτμλ += '<td>' + temp.embadon + '</td>';
ρορυρΗτμλ += '</tr>';
```

Εικόνα 4.23: Απεικόνιση κώδικα της λειτουργίας ρορ up της οντότητας 'Αρχαιολογικός Χώρος' στο Microsoft Visual Studio.

Τελός, το αποτέλεσμα της αποτύπωσης της χωρικής πληροφορίας της βάσης δεδομένων στο χάρτη, απεικονίζεται στην παρακάτω εικόνα.



Εικόνα 4.24: Απεικόνιση χάρτη και της λειτουργίας ρορ up της οντότητας 'Αρχαιολογικός Χώρος' στη διεπαφή χρήστη.

4.5.3 Υποσύστημα Χωρικών Ερωτημάτων (E03)

Το Υποσύστημα των χωρικών ερωτημάτων είναι το βασικότερο υποσύστημα αλληλεπίδρασης του χρήστη με τα γεωχωρικά δεδομένα που είναι αποθηκευμένα στη βάση δεδομένων. Είναι μία μέθοδος επικοινωνίας, αφού με την εκτέλεση του ανακτάται ένα συγκεκριμένο υποσύνολο των δεδομένων βάσει κριτηρίων που έχουν οριστεί κατά τον σχεδιασμό της βάσης δεδομένων.

Το υποσύστημα δημιουργήθηκε με τέτοιο τρόπο, ώστε να εξαγεί αποτελέσματα αυτόματα μετά την καταχώρηση νέων δεδομένων. Με αυτό τον τρόπο, ενημερώνεται το αποτέλεσμα του χωρικού ερωτήματος, χωρίς την επέμβαση του χρήστη.

Όσον αφορά την υλοποίησή του τεχνικά, ακολουθήθηκε το μοτίβο σχεδίασης MVC και στην περίπτωση αυτή, με τον ελεγκτή (controller) να εκτελεί συναρτήσεις που στην ουσία είναι τα χωρικά ερωτήματα και τις προβολές να οπτικοποιούν τα αποτελέσματα αυτών των ερωτημάτων.

Ενδεικτικά, θα παρουσιαστεί ένα χωρικό ερωτήμα.

-Δείξε μου τα Ευρηματα που είναι λιθινα , σε ποιες ανασκαφικές ομάδες ανηκουν, και αυτές οι ανασκαφικές ομάδες σε ποια ανσκαφικά τετραγωνα ανηκουν.

Η εκτέλεση του χωρικού ερωτήματος γίνεται με τη συνάρτηση GetQueryA().

Ακολουθεί στιγμιότυπο εικόνας του κώδικα.

```
1 reference
public List<Query> GetQueryA()
{
    List<Query> queryA = new List<Query>();
    SqlCommand cmd = new SqlCommand("SELECT          f.Name as a,          g.Name as b,          ex.Name as c,          sq.Name as d          " +
        "FROM [excavationgeodb].[dbo].[Finding] as f, [excavationgeodb].[dbo].[Genre] as g , " +
        "[excavationgeodb].[dbo].[ExcavationUnit] as ex          , " +
        "[excavationgeodb].[dbo].[SquareTable] as sq          " +
        "where f.GenreId = 2 and f.GenreId = g.Id and f.ExcavationUnitId = ex.Id and ex.SquareTableId = sq.Id ;");

    cmd.CommandType = CommandType.Text;
    cmd.Connection = _connection;
    SqlDataAdapter adapter = new SqlDataAdapter(cmd);

    DataTable dt = new DataTable();

    adapter.Fill(dt);

    foreach (DataRow dr in dt.Rows)
    {
        queryA.Add(new Query(
            dr["a"].ToString(),
            dr["b"].ToString(),
            dr["c"].ToString(),
            dr["d"].ToString()
        ));
    }
    return queryA;
}
```

Εικόνα 4.25: Απεικόνιση του κώδικα της συνάρτησης GetQueryA() στο Microsoft Visual Studio.

Το αποτέλεσμα οπτικοποιείται μέσω της προβολής που έχει δημιουργηθεί για το συγκεκριμένο χωρικό ερώτημα. Ακολουθεί στιγμιότυπο εικόνας του κώδικα.

```

Δείξε μου τα Ευρηματα που είναι λιθινα , σε ποιες ανασκαφικές ομάδες ανηκουν,
και αυτές οι ανασκαφικές ομάδες σε ποια ανασκαφικά τετραγωνα ανηκουν.
<table id="table">
  <tr>
    <th>ΕΥΡΗΜΑ</th>
    <th>ΥΛΙΚΟ</th>
    <th>ΑΝΑΣΚΑΦΙΚΗ ΟΜΑΔΑ</th>
    <th>ΤΕΤΡΑΓΩΝΟ</th>
  </tr>
  @foreach (var item in Model.QueryA)
  {
    <tr>
      <td>
        @Html.DisplayFor(modelItem => item.result1)
      </td>
      <td>
        @Html.DisplayFor(modelItem => item.result2)
      </td>
      <td>
        @Html.DisplayFor(modelItem => item.result3)
      </td>
      <td>
        @Html.DisplayFor(modelItem => item.result4)
      </td>
    </tr>
  }

```

Εικόνα 4.26: Απεικόνιση του κώδικα της προβολής του χωρικού ερωτήματος στο Microsoft Visual Studio.

4.5.4 Υποσύστημα Αποθήκευσης Πολυμέσων (E04)

Το συγκεκριμένο υποσύστημα δημιουργήθηκε για να ενισχύσει τα μέσα τεκμηρίωσης των ανασκαφικών δεδομένων με την αποθήκευση πολυμέσων, όπως π.χ είναι οι φωτογραφίες και τα βίντεο.

Πιο συγκεκριμένα, δίνει τη δυνατότητα φόρτωσης αρχείου στη βάση δεδομένων, είτε αυτό είναι φωτογραφία, βίντεο, ήχος, αρχείο σχεδίου κ.λ.π., όπως επίσης παρέχει τη δυνατότητα αποθήκευσης μιας διεύθυνσης της ιστοσελίδας Youtube. Αυτή είναι μία δημοφιλής πρακτική αποθήκευσης των βίντεο, λόγω του ότι προκειται συνήθως για αρχεία που έχουν αυξημένες απαιτήσεις αποθηκευτικού χώρου. Επιπροσθέτως, με τη συγκεκριμένη πρακτική μπορεί ο χρήστης να αποθηκεύσει περισσότερα του ενός βίντεο.

Ο κώδικας υλοποίησης της φόρτωσης αρχείου στη βάση δεδομένων είναι ο παρακάτω:

```

var file = Path.Combine(_environment.ContentRootPath, "uploads", Upload.FileName);
using (var fileStream = new FileStream(file, FileMode.Create))
{
    await Upload.CopyToAsync(fileStream);

    Finding.Image = Upload.FileName;
}

```



```

        findingRepository.UpdateFindingFile(Finding);

        return RedirectToPage("./Findings");
    }

```

Ο παραπάνω κώδικας υλοποιείται μόνο όταν γίνει έλεγχος στο πεδίο της εγγραφής ότι έχει τιμή διάφορη του κενού, εμφανίζεται ένα κουμπί λήψης αρχείου. Ο κώδικας που υλοποιεί αυτό τον έλεγχο είναι ο παρακάτω:

```

<td>
    @if (item.Image != "-")
    {
        <a asp-page="./Findings"
          asp-page-handler="Download" asp-route-id="@item.Id">
            <button type="submit"></button>
        </a>

        <a asp-page="./Findings"
          asp-page-handler="DeleteImage" asp-route-id="@item.Id">
            <button type="submit"></button>
        </a>
    } else
    {
        <a asp-page="./UploadImageFinding" asp-route-id="@item.Id"> </a>
    }
</td>

```

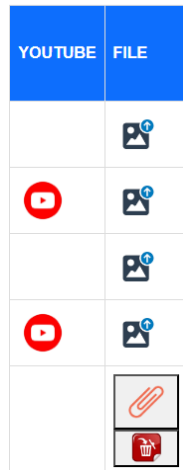
Στη διεπαφή χρήστη εμφανίζεται στη στήλη 'FILE' είτε ένα κουμπί φόρτωσης αρχείου είτε στις περιπτώσεις που υπάρχουν εγγραφές ένα 'κουμπί' ότι υπάρχει συνημμένο αρχείο και ένα 'κουμπί' για την διαγραφή του.

Όσον αφορά την αποθήκευση της διεύθυνσης Youtube στη βάση δεδομένων, προγραμματιστικά πρώτα γίνεται έλεγχος, εάν το πεδίο που πρόκειται να γίνει η αποθήκευση της διεύθυνσης έχει τιμή διάφορη της παύλας (-). Στην περίπτωση που ισχύει η παραπάνω συνθήκη εμφανίζεται ένα 'κουμπί' που αποτελεί υπερσύνδεσμο και οδηγεί στη διεύθυνση (url) που έχει θέσει ο χρήστης. Ακολουθεί ο κώδικας υλοποίησης:

```

<td>
    @if (item.Youtube != "-")
    {
        <a href="@Html.DisplayFor(modelItem => item.Youtube)" target="_blank" data-
        toggle="modal"> 
        </a>
    }
</td>

```



Εικόνα 4.27: Απεικόνιση φόρτωσης αρχείου και αποθήκευσης διεύθυνσης Youtube στη διεπαφή χρήστη.

4.5.5 Υποσύστημα Αυθεντικοποίησης Χρήστη (E05)

Το υποσύστημα αυτό αποτελεί μία από τις κρίσιμότερες λειτουργίες της εφαρμογής. Μέσω αυτού, οι χρήστες του συστήματος έχουν διαβαθμισμένη και ασφαλή πρόσβαση σε λειτουργίες και πληροφορίες του συστήματος, ενώ ταυτόχρονα διασφαλίζεται ότι μόνο εξουσιοδοτημένοι χρήστες θα έχουν πρόσβαση στο σύστημα. Το υποσύστημα αυτό παρέχει στους χρήστες του συστήματος τις παρακάτω υπηρεσίες:

- **Εγγραφή νέου λογαριασμού:** Η διαδικασία αυτή περιλαμβάνει την καταχώρηση νέου χρήστη στο σύστημα.
- **Είσοδος στο λογαριασμό:** Η διαδικασία αυτή περιλαμβάνει την επαλήθευση της ταυτότητας του χρήστη και την είσοδο στον προσωπικό του λογαριασμό.
- **Αποσύνδεση:** Η διαδικασία αυτή περιλαμβάνει την αποσύνδεση από τον λογαριασμό μετά την ολοκλήρωση των εργασιών.
- **Επαναφορά κωδικού πρόσβασης:** Η διαδικασία αυτή περιλαμβάνει την ανάκτηση του κωδικού πρόσβασης σε περίπτωση που ο χρήστης κάνει τέτοιο αίτημα
- **Αλλαγή κωδικού πρόσβασης:** Η διαδικασία αυτή περιλαμβάνει τη δυνατότητα που έχει ο χρήστης να αλλάξει τον κωδικό πρόσβασης για λόγους ασφαλείας.

- **Πρόταση ισχυρού κωδικού πρόσβασης:** Η διαδικασία αυτή περιλαμβάνει την υπηρεσία πρότασης ενός πολύπλοκου κωδικού πρόσβασης για αυξημένη ασφάλεια.

Η υλοποίηση του υποσυστήματος έγινε με τη βοήθεια του μοτίβου σχεδίασης Asp .Net Core MVC, το οποίο περιέχει κάποιες έτοιμες ρουτίνες αυθεντικοποίησης χρήστη. Επιπροσθέτως, παρέχει διεπαφή χρήστη για την επιλογή των επιθυμητών λειτουργιών αυθεντικοποίησης, όπως φαίνεται στην παρακάτω εικόνα.

The screenshot shows the 'Add Identity' dialog box. At the top, there is a title bar with a close button (X). Below the title, the text reads: 'Select an existing layout page, or specify a new one: /Areas/Identity/Pages/Account/Manage/_Layout.cshtml (Leave empty if it is set in a Razor _viewstart file)'. There is a text input field with a dropdown arrow and a plus sign. Below this, there is a checkbox for 'Override all files'. The main section is titled 'Choose files to override' and contains a grid of checkboxes for various files. The files are organized into three columns. The first column includes files like Account\StatusMessage, Account\ConfirmEmailChange, Account\ForgotPasswordConfirmation, etc. The second column includes Account\AccessDenied, Account\ExternalLogin, Account\Lockout, etc. The third column includes Account\ConfirmEmail, Account\ForgotPassword, Account>Login, etc. At the bottom, there are three dropdown menus: 'DbContext class' (set to ApplicationDbContext (WebApplication2.Data)), 'Database provider' (set to Configured from the selected DbContext), and 'User class'. There are 'Add' and 'Cancel' buttons at the bottom right.

Εικόνα 4.28: Απεικόνιση διεπαφής χρήστη για την εγκατάσταση υπηρεσιών αυθεντικοποίησης χρήστη

Τέλος, με την ολοκλήρωση της παραπάνω διαδικασίας, δημιουργείται ο αντίστοιχος πίνακας στο σύστημα διαχείρισης της βάσης δεδομένων, ο οποίος είναι ο 'Asp.NetUserLogins'.

4.5.6 Υποσύστημα Διαχείρισης Πρόσβασης (E06)

Το υποσύστημα διαχείρισης πρόσβασης είναι ένα μοντέλο διαχείρισης της πρόσβασης στην πληροφορία της βάσης δεδομένων, το οποίο βασίζεται στον καθορισμό των ρόλων που μπορεί να έχει ένας χρήστης και των αντίστοιχων δικαιωμάτων πρόσβασης που αυτοί οι ρόλοι φέρουν.

Οι ρόλοι ορίζουν τις διαφορετικές θέσεις ή λειτουργίες ενός χρήστη στην εφαρμογή, βάσει των δικαιωμάτων πρόσβασης που μπορεί να εκτελέσει ένας ρόλος.

Το υποσύστημα αυτό επιτρέπει την αποτελεσματική διαχείριση της πρόσβασης στην εφαρμογή, αφού διευκολύνει τη διαχείριση των ρόλων και των δικαιωμάτων, εξασφαλίζοντας την ασφάλεια και την αποτελεσματικότητα του συστήματος πρόσβασης.

Το υποσύστημα διαχείρισης πρόσβασης της εφαρμογής περιέχει 3 επίπεδα πρόσβασης, ως εξής:

- **Administrator:** Βασικός διαχειριστής του συστήματος με πρόσβαση σε εργαλεία διαχείρισης της εφαρμογής και ο μόνος εξουσιοδοτημένος χρήστης, ο οποίος μπορεί να δίνει ρόλους.
- **Editor:** Χρήστες με πρόσβαση σε όλα τα δεδομένα της εφαρμογής με δυνατότητα προσθαφαίρεσης περιεχομένου.
- **Member:** Χρήστες με μόνη δυνατότητα αυτής της θέασης του χαρτογραφικού υποβάθρου της εφαρμογής.

Αρχικά, η υλοποίηση του υποσυστήματος προγραμματιστικά έγινε με τη δημιουργία των ρόλων στην κύρια κλάση της εφαρμογής (Program.cs).

```
using (var scope = app.Services.CreateScope())
{
    var roleManager = scope.ServiceProvider.GetRequiredService<RoleManager<IdentityRole>>();
    var roles = new[] { "admin", "member", "editor" };

    foreach (var role in roles)
    {
        if (!await roleManager.RoleExistsAsync(role))
        {
            await roleManager.CreateAsync(new IdentityRole(role));
        }
    }
}
```

Εικόνα 4.29: Απεικόνιση αποσπασματος κώδικα δημιουργίας των ρόλων χρήστη της εφαρμογής στο Microsoft Visual Studio.

Στη συνέχεια, εφαρμόστηκε η αρχιτεκτονική του μοτίβου σχεδίασης Asp.Net Core MVC με τη δημιουργία των κύριων χαρακτηριστικών του μοτίβου, που είναι το μοντέλο (model), ο ελεγκτής (controller), η κλάση Repository και η προβολή (page) για τη συγκεκριμένη οντότητα 'User'.

Το πρώτο βήμα ήταν η δημιουργία του μοντέλου. Με αυτόν τον τρόπο ορίζουμε μία κλάση με τα πεδία της, που θα αποτελεί την περιγραφή ενός πίνακα της βάσης δεδομένων και θα δημιουργεί αντικείμενα ίδιου τύπου.

```
using Microsoft.AspNetCore.Identity;

namespace ExcavationGeo.Models
{
    9 references
    public class AppUser : IdentityUser
    {
        4 references
        public string UserId { get; set; }

        2 references
        public string Email { get; set; }

        4 references
        public string RoleName { get; set; }

        2 references
        public string FullName { get; set; }

        2 references
        public string Status { get; set; }

        1 reference
        public AppUser(string userId, string email, string roleName, string fullName, string status)
        {
            UserId = userId;
            Email = email;
            RoleName = roleName;
            FullName = fullName;
            Status = status;
        }
    }
}
```

Εικόνα 4.30: Απεικόνιση κώδικα δημιουργίας του μοντέλου User της εφαρμογής στο Microsoft Visual Studio.

Έπειτα, δημιουργήθηκε ο ελεγκτής (controller) με σκοπό την αρχικοποίηση της κλάσης Repository.

```
namespace ExcavationGeo.Controllers
{
    0 references
    public class UserController : Controller
    {
        0 references
        public IActionResult Index()
        {
            List<AppUser> appUsers = new List<AppUser>();

            AppUserRepository appUserRepository = new AppUserRepository();

            appUsers = appUserRepository.GetUsers();

            return View(appUsers);
        }
    }
}
```

Εικόνα 4.31: Απεικόνιση κώδικα δημιουργίας ελεγκτή του μοντέλου User της εφαρμογής στο Microsoft Visual Studio.

Ακολούθησε, η δημιουργία της κλάσης Repository, η οποία έχει τον ρόλο να αντλεί δεδομένα από τη βάση δεδομένων, όπως επίσης να ενημερώνει, να διαγράφει και να εισάγει δεδομένα.

```
1 reference
public void RemoveRole(String id)
{
    try
    {
        SqlDataReader dr;

        SqlCommand cmd1 = new SqlCommand("delete FROM [ExcavationGeoDB].[dbo].[AspNetUserRoles] where UserId = '" + id + "'");

        _connection.Open();
        cmd1.CommandType = System.Data.CommandType.Text;
        cmd1.Connection = _connection;

        Console.WriteLine("Error Generated. Details: " + cmd1.CommandText);

        dr = cmd1.ExecuteReader();
    }
    catch (Exception ex) { }
}
```

Εικόνα 4.32: Απεικόνιση κώδικα της συνάρτησης 'RemoveRole' του μοντέλου User της εφαρμογής στο Microsoft Visual Studio.

Το τελευταίο βήμα ήταν η δημιουργία της προβολής, στην οποία γίνεται προσθήκη ή αφαίρεση ρόλου από τον διαχειριστή.

```
<td>

    @if (item.RoleName == "editor")
    {
        <a asp-page="./AppUsers" onclick="return confirm
('Αφαίρεση Ρόλου');">
            asp-page-handler="Delete" asp-route-id="@item.UserId">
            <button type="submit" class="btn btn-danger">Αφαίρεση Ρόλου</button>
        </a>
    }
    @if (item.RoleName == "") {
        <a asp-page="./AppUsers" onclick="return confirm
('Προσθήκη Ρόλου');">
            asp-page-handler="Add" asp-route-id="@item.UserId">
            <button type="submit" class="btn btn-info">Προσθήκη Ρόλου</button>
        </a>
    }

</td>
```

Εικόνα 4.33: Απεικόνιση αποσπάσματος κώδικα της προβολής στον οποίο γίνεται προσθήκη η αφαίρεση ρόλου στο μοντέλο User της εφαρμογής στο Microsoft Visual Studio.

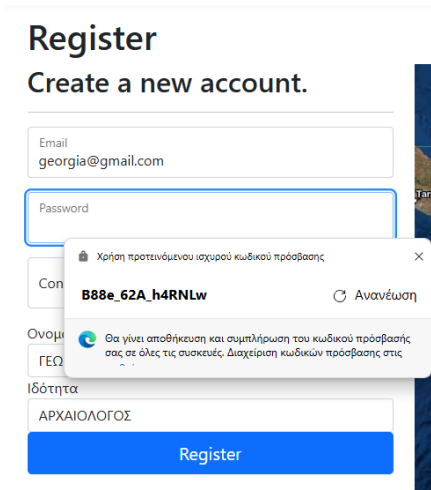
4.6 Συνδυασμός Υποσυστημάτων – Τελική Διεπαφή Χρήστη

Βασική επιδίωξη της δημιουργίας των υποσυστημάτων ήταν διασύνδεση τους, έτσι ώστε να προκύψει μία διεπαφή χρήστη, η οποία θα είναι λειτουργική και κατανοητή από τους χρήστες.

Στη συνέχεια, θα ακολουθήσει μία περιγραφή των λειτουργιών της εφαρμογής μέσω της διεπαφής χρήστη που έχει δημιουργηθεί.

1) **Εγγραφή Χρήστη:** Με αυτή την ενέργεια καλείται το υποσύστημα Αυθεντικοποίησης Χρήστη.

Κατά την είσοδο του ένας επισκέπτης μπορεί να κάνει εγγραφή χρήστη και η εφαρμογή να του προτείνει έναν ισχυρο κωδικό πρόσβασης.



The screenshot shows a 'Register' form with the following elements:

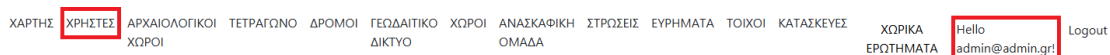
- Header: Register, Create a new account.
- Email field: georgia@gmail.com
- Password field: (empty)
- Generated Password: B88e_62A_h4RNLw (with 'Ανανέωση' button)
- Notification: 'Χρήση προτεινόμενου ισχυρού κωδικού πρόσβασης' and 'Θα γίνει αποθήκευση και συμπλήρωση του κωδικού πρόσβασης σας σε όλες τις συσκευές. Διαχείριση κωδικών πρόσβασης στις...'
- Other fields: Ονομα (GEO), Ιδιότητα (ΑΡΧΑΙΟΛΟΓΟΣ)
- Register button

Εικόνα 4.34: Απεικόνιση εγγραφής χρήστη στην εφαρμογή.

Ο χρήστης όταν εγγράφεται στην εφαρμογή έχει τη δυνατότητα μόνο θέασης της χωρικής πληροφορίας της εφαρμογής.

2) **Προσθήκη ρόλου από τον Διαχειριστή:** Με αυτή την ενέργεια καλείται το υποσύστημα Διαχείρισης Πρόσβασης.

Ο Διαχειριστής θα πρέπει να συνδεθεί στην εφαρμογή με τα διαπιστευτήρια του και να δώσει ρόλο στο χρήστη. Αρχικά, θα πρέπει να μεταβεί στην καρτέλα «ΧΡΗΣΤΕΣ».



Εικόνα 4.35: Απεικόνιση του μενού της εφαρμογής.

Η διεπαφή του μενού «ΧΡΗΣΤΕΣ» περιέχει έναν πίνακα, στον οποίο απεικονίζονται τα στοιχεία των εγγεγραμμένων χρηστών, καθώς και ο ρόλος που κατέχουν. Στη συνέχεια, ο Διαχειριστής θα τροποποιήσει το ρόλο του τελευταίου εγγεγραμμένου χρήστη και από 'member' θα του προσδώσει ρόλο 'editor'.

The screenshot shows a web application interface with a navigation menu at the top. A modal dialog box is open in the center, titled "To localhost:5296 λέει Προσθήκη Ρόλου". The dialog has two buttons: "OK" and "Άκυρο". Below the dialog, the "Χρήστες" (Users) table is visible. The table has columns for Email, Ρόλος (Role), Ονοματεπώνυμο (Full Name), Ιδιότητα (Specialty), and an action column. The user "georgia@gmail.com" is highlighted in grey, and the "editor" role is being assigned to them.

Email	Ρόλος	Ονοματεπώνυμο	Ιδιότητα	
ab@cd.gr		edgs	dsgdsa	Προσθήκη Ρόλου
admin@admin.gr	admin			
georgia@gmail.com		ΓΕΩΡΓΙΑ ΤΑΜΠΑΚΟΠΟΥΛΟΥ	ΑΡΧΑΙΟΛΟΓΟΣ	Προσθήκη Ρόλου
ta@gmail.com	editor	ΑΝΑΣΤΑΣΙΟΣ ΠΑΠΑΔΟΠΟΥΛΟΣ	ΑΡΧΑΙΟΛΟΓΟΣ	Αφαίρεση Ρόλου

Εικόνα 4.36: Απεικόνιση διαδικασίας προσθήκη ρόλου από τον Διαχειριστή.

Μετά την ολοκλήρωση τροποποίησης του ρόλου του χρήστη από τον Διαχειριστή, ο χρήστης έχει τη δυνατότητα θέασης και διαχείρισης της πληροφορίας της βάσης δεδομένων.

The screenshot shows the "Χρήστες" (Users) table after the role change. The user "georgia@gmail.com" now has the role "editor" (highlighted in red) and the "Αφαίρεση Ρόλου" (Remove Role) button is visible. The user "ta@gmail.com" also has the role "editor" and the "Αφαίρεση Ρόλου" button.

Email	Ρόλος	Ονοματεπώνυμο	Ιδιότητα	
ab@cd.gr		edgs	dsgdsa	Προσθήκη Ρόλου
admin@admin.gr	admin			
georgia@gmail.com	editor	ΓΕΩΡΓΙΑ ΤΑΜΠΑΚΟΠΟΥΛΟΥ	ΑΡΧΑΙΟΛΟΓΟΣ	Αφαίρεση Ρόλου
ta@gmail.com	editor	ΑΝΑΣΤΑΣΙΟΣ ΠΑΠΑΔΟΠΟΥΛΟΣ	ΑΡΧΑΙΟΛΟΓΟΣ	Αφαίρεση Ρόλου

Εικόνα 4.37: Απεικόνιση πίνακα χρηστών μετά την τροποποίηση ρόλου χρήστη από τον Διαχειριστή.

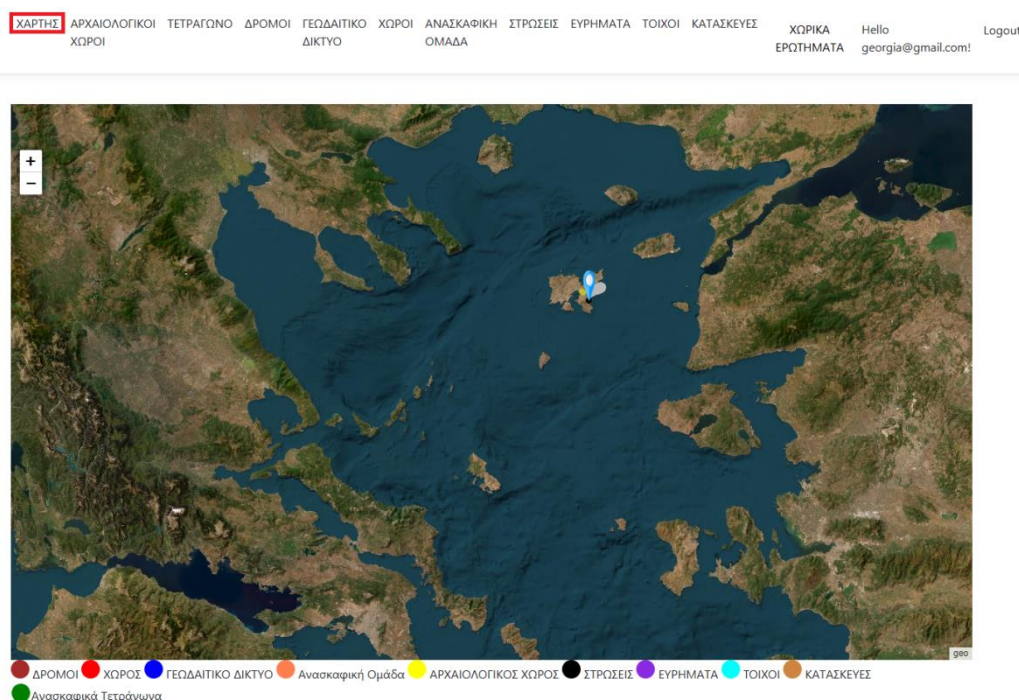
3) **Σύνδεση Χρήστη:** Με αυτή την ενέργεια είναι διαθέσιμα προς χρήση τα υποσυστήματα Λειτουργιών CRUD, Χαρτογραφικού Υποβάθρου, Χωρικών Ερωτημάτων και Αποθήκευσης Πολυμέσων.

The screenshot shows the user management interface with a navigation menu. The user "georgia@gmail.com" is highlighted in red, and the "Logout" button is visible next to the user's name.

Εικόνα 4.38: Απεικόνιση σύνδεσης χρήστη στο μενού της εφαρμογής.

4) **Ετικέτα Χάρτης:** Με αυτή την ενέργεια καλείται το υποσύστημα του Χαρτογραφικού υποβάθρου της εφαρμογής, το οποίο στην περίπτωση αυτή αποτυπώνει όλη την πληροφορία της χωρικής βάσης στο χάρτη.

Ο χάρτης διαθέτει υπόμνημα με διαφορετικό χρώμα για την κάθε οντότητα και λειτουργία pop up.



Εικόνα 4.39: Απεικόνιση διεπαφής χρήστη της ετικέτας «Χάρτης» της εφαρμογής.

5) **Ετικέτα Ευρήματα:** Με αυτή την ενέργεια καλείται το υποσύστημα των Λειτουργιών CRUD σε συνδυασμό με το υποσύστημα του Χαρτογραφικού Υποβάθρου και το υποσύστημα Αποθήκευσης Πολυμέσων.

Το υποσύστημα Λειτουργιών CRUD ενεργοποιεί τις εντολές «Καταχώρηση Νέου Ευρήματος», «Επεξεργασία» και «Διαγραφή», όταν αυτές επιλεγούν από το χρήστη. Με την ενεργοποίηση της εντολής «Καταχώρηση Νέου Ευρήματος», εμφανίζεται μία διεπαφή χρήστη που έχει τη διάταξη συμπληρωσης στοιχείων φόρμας. Τα πεδία της φόρμας είτε έχουν προκαθορισμένο πεδίο τιμών είτε όχι. Τα πεδία με προκαθορισμένο πεδίο τιμών εμφανίζονται ως dropdown λίστες και ο χρήστης έχει τη δυνατότητα να επιλέξει μία από αυτές τις τιμές. Επίσης, ως dropdown λίστες εμφανίζονται και οι οντότητες που έχουν κάποια καθορισμένη σχέση με το προς συμπλήρωση αντικείμενο της οντότητας.

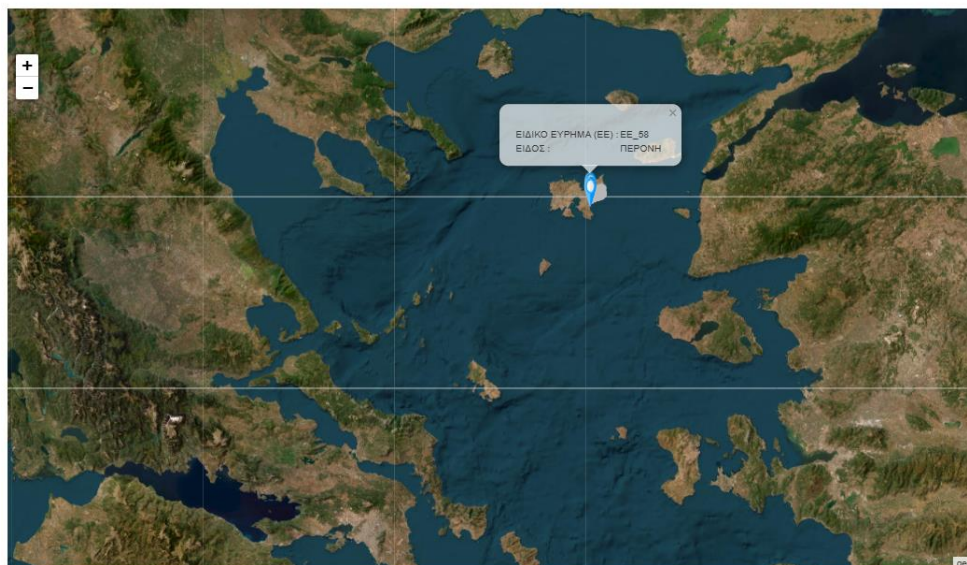
ΕΥΡΗΜΑΤΑ

Id	0
ΕΙΔΙΚΟ ΕΥΡΗΜΑ (ΕΕ)	ΕΕ_58
*ΗΜΕΡΟΜΗΝΙΑ	02/07/2024 22:47
ΕΙΔΟΣ	ΠΕΡΟΝΗ
ΥΨΟΜΕΤΡΟ	10.52
ΣΥΝΤΕΤΑΓΜΕΝΕΣ Χ(m),Υ(m) στο ΕΓΣΑ'87:	614278.189,4415346.525
ΜΗΚΟΣ σε (cm)	1.2
*ΠΛΑΤΟΣ σε (cm)	1,0
*ΥΨΟΣ σε (cm)	0,5
ΧΡΟΝΟΛΟΓΗΣΗ	Η Χρονολόγηση του ειδικού ευρήματος
ΥΛΙΚΟ	Μέταλλο
	Πηλός
	Λίθος
	Μέταλλο
	Οργανικό
	Άλλο
*ΚΑΤΑΣΤΑΣΗ ΔΙΑΤΗΡΗΣΗΣ	Ακέραιο
*ΑΝΑΣΚΑΦΙΚΗ ΟΜΑΔΑ	Ανασκαφική Ομάδα 3
Youtube	https://www.youtube.com/watch?v=HBcdfFIQqww
	Καταχώρηση

Εικόνα 4.40: Απεικόνιση συμπλήρωσης φόρμας με την καταχώρηση νέου ευρήματος στη διεπαφή χρήστη της εφαρμογής.

Με την καταχώρηση του νέου ευρήματος εκτελείται και το υποσύστημα του Χαρτογραφικού Υποβάθρου, αφού η προβολή έχει δημιουργηθεί με τέτοιο τρόπο ώστε να αποτυπώνεται το νέο εύρημα στο χάρτη και να συμπληρώνεται η εγγραφή του στον πίνακα με τις εγγραφές των ευρημάτων που ακολουθεί.

ΕΥΡΗΜΑΤΑ



Εικόνα 4.41: Απεικόνιση αποτύπωσης νέου ευρήματος στο χάρτη της εφαρμογής.

ΚΑΤΑΧΩΡΗΣΗ ΝΕΟΥ ΕΥΡΗΜΑΤΟΣ																	
ΕΙΔΙΚΟ ΕΥΡΗΜΑ (ΕΕ)	ΗΜΕΡΟΜΗΝΙΑ	ΕΙΔΟΣ	ΥΨΟΜΕΤΡΟ	ΣΥΝΤΕΤΑΓΜΕΝΕΣ Χ(μ),Υ(μ) στο ΕΓΣΑ'87	ΜΗΚΟΣ σε (cm)	ΠΛΑΤΟΣ σε (cm)	ΥΨΟΣ σε (cm)	ΧΡΟΝΟΛΟΓΗΣΗ	ΥΛΙΚΟ	ΚΑΤΑΣΤΑΣΗ ΔΙΑΤΗΡΗΣΗΣ	ΠΑΡΑΤΗΡΗΣΕΙΣ	ΣΥΝΤΑΚΤΗΣ	Ανασκαφική Ομάδα	YOUTUBE	FILE		
EE_1	2/6/2022 12:00:00	Σφονδύλι	16	614685.4706,4412091.9069	1,5	1	3,5	Χρονολόγηση του ΕΕ	Λίθος	Ακέραιο	Παρατηρήσεις για το ΕΕ	Κώστας Παπαδόπουλος	Ανασκαφική Ομάδα 1				Επεξεργασία Διαγραφή
EE_2	11/6/2022 12:00:00	Σφονδύλι	15,95	614685.4706,4412091.9069	1	1	3	Χρονολόγηση του ΕΕ	Λίθος	Ακέραιο	Παρατηρήσεις για το ΕΕ	Κώστας Παπαδόπουλος	Ανασκαφική Ομάδα 1				Επεξεργασία Διαγραφή
EE_3	2/6/2022 12:00:00	Σφονδύλι	16,05	614687.0314,4412091.7527	1,3	1,2	2,5	Χρονολόγηση του ΕΕ	Λίθος	Ήμιου	Παρατηρήσεις για το ΕΕ	Κώστας Παπαδόπουλος	Ανασκαφική Ομάδα 2				Επεξεργασία Διαγραφή
EE_4	2/6/2022 12:00:00	Όνομα ΕΕ	16	614687.9927,4412091.7409	1	1	2	Χρονολόγηση του ΕΕ	Μέταλλο	Ένα τρίτο	Παρατηρήσεις για το ΕΕ	Κώστας Παπαδόπουλος	Ανασκαφική Ομάδα 2				Επεξεργασία Διαγραφή
EE_5	2/6/2022 12:00:00	Όνομα ΕΕ	15,85	614687.4646,4412092.5711	0,5	0,5	1	Χρονολόγηση του ΕΕ	Οργανικό	Δύο τρίτα	Παρατηρήσεις για το ΕΕ	Κώστας Παπαδόπουλος	Ανασκαφική Ομάδα 2				Επεξεργασία Διαγραφή
EE_11	15/6/2022 12:00:00	Όνομα ΕΕ	15,8	614686.6596,4412095.8881	1,5	1,5	1	Χρονολόγηση του ΕΕ	Οργανικό	Δύο τρίτα	Παρατηρήσεις για το ΕΕ	Κώστας Παπαδόπουλος	Ανασκαφική Ομάδα 3				Επεξεργασία Διαγραφή
EE_12	20/6/2022 12:00:00	Όνομα ΕΕ	15,78	614687.9561,4412096.2819	1,3	1	2,8	Χρονολόγηση του ΕΕ	Μέταλλο	Ένα τρίτο	Παρατηρήσεις για το ΕΕ	Κώστας Παπαδόπουλος	Ανασκαφική Ομάδα 1				Επεξεργασία Διαγραφή
EE_58	2/7/2024 10:47:00	ΠΕΡΟΝΗ	10,52	614278.189,4415346.525	1,2	1	0,5	Η Χρονολόγηση του ειδικού ευρήματος	Μέταλλο	Ακέραιο	Οι παρατηρήσεις του ειδικού ευρήματος	Γεωργία Ταμπακοπούλου	Ανασκαφική Ομάδα 3				Επεξεργασία Διαγραφή

Εικόνα 4.42: Απεικόνιση εγγραφής νέου ευρήματος στον πίνακα με τις εγγραφές των ευρημάτων της βάσης δεδομένων της εφαρμογής.

Με την ενεργοποίηση της εντολής 'Επεξεργασία' μίας συγκεκριμένης εγγραφής, εμφανίζεται η συμπληρωμένη φόρμα της εγγραφής, την οποία ο χρήστης μπορεί να επεξεργαστεί και να αποθηκεύσει ξανά.

ΕΥΡΗΜΑΤΑ

Id

8

ΕΙΔΙΚΟ ΕΥΡΗΜΑ (ΕΕ)

EE_58

ΗΜΕΡΟΜΗΝΙΑ

02/07/2024 10:47

ΕΙΔΟΣ

ΠΕΡΟΝΗ

ΥΨΟΜΕΤΡΟ

10,52

ΣΥΝΤΕΤΑΓΜΕΝΕΣ Χ(μ),Υ(μ) στο ΕΓΣΑ'87:

614278.189,4415346.525

ΜΗΚΟΣ σε (cm)

1,2

ΠΛΑΤΟΣ σε (cm)

1,5

ΥΨΟΣ σε (cm)

0,5

ΧΡΟΝΟΛΟΓΗΣΗ

Η Χρονολόγηση του ειδικού ευρήματος

ΥΛΙΚΟ

Μέταλλο

ΚΑΤΑΣΤΑΣΗ ΔΙΑΤΗΡΗΣΗΣ

Ακέραιο

ΠΑΡΑΤΗΡΗΣΕΙΣ

Οι παρατηρήσεις του ειδικού ευρήματος

ΣΥΝΤΑΚΤΗΣ

Γεωργία Ταμπακοπούλου

ΑΝΑΣΚΑΦΙΚΗ ΟΜΑΔΑ

Ανασκαφική Ομάδα 3

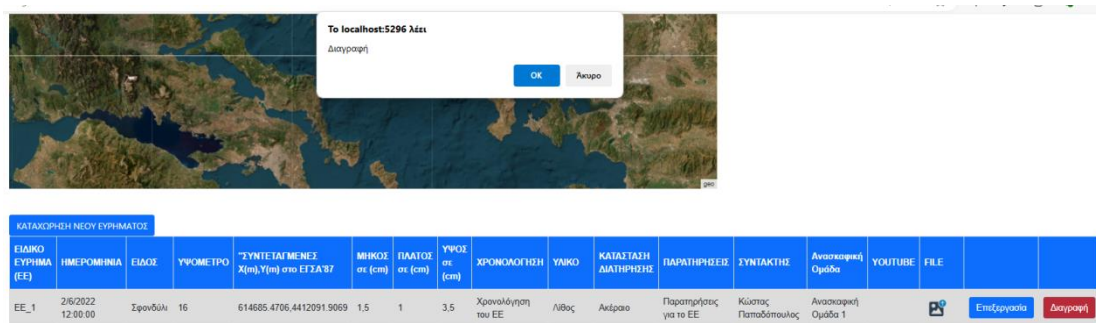
Youtube

https://www.youtube.com/watch?v=HBcdfItEqqw

[Κατοχώρηση](#)

Εικόνα 4.43: Απεικόνιση επεξεργασίας πεδίου ευρήματος και εκ νέου αποθήκευση στη βάση δεδομένων της εφαρμογής.

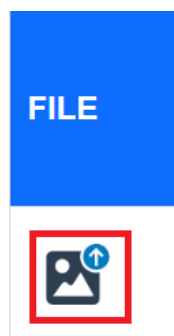
Με την ενεργοποίηση της εντολής 'Διαγραφή' μίας συγκεκριμένης εγγραφής ζητείται επιβεβαίωση από τον χρήστη πριν την οριστική διαγραφή από τη βάση δεδομένων.



Εικόνα 4.44: Απεικόνιση διαγραφής ευρήματος από τη βάση δεδομένων της εφαρμογής.

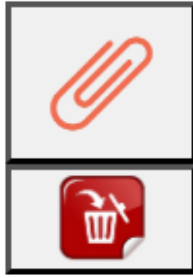
Με τη συμπλήρωση του πεδίου Youtube εκτελείται το υποσύστημα Αποθηκευσης πολυμέσων και συγκεκριμένα η λειτουργία που αφορά το Youtube. Στον πίνακα των εγγραφών των ευρημάτων, εμφανίζεται εικόνα με το λογότυπο του Youtube μετά την επιτυχή καταχώρηση του νέου ευρήματος. Ο χρήστης έχει τη δυνατότητα να επιλέξει την εικόνα του λογότυπου του Youtube της εγγραφής και να μεταφερθεί στην εν λόγω ιστοσελίδα, η οποία αυτόματα θα του προβάλει το βίντεο.

Επιπροσθέτως, μετά την καταχώρηση του νέου ευρημάτων, ο χρήστης μπορεί να επιλέξει το πεδίο 'FILE' της συγκεκριμένης εγγραφής και να επισυνάψει κάποιο αρχείο. Στην περίπτωση που η εγγραφή δεν έχει κάποιο συνημμένο αρχείο εμφανίζεται η παρακάτω εικόνα.



Εικόνα 4.45: Απεικόνιση εικόνας επισύναψης αρχείου σε μία συγκεκριμένη εγγραφή της βάσης δεδομένων της εφαρμογής.

Στην περίπτωση που μία εγγραφή έχει συνημμένο αρχείο, τότε εμφανίζεται στη στήλη 'FILE' εικόνα συνημμένου αρχείου και εικόνα διαγραφής του συνημμένου αρχείου. Επιλέγοντας ο χρήστης μία από αυτές τις εικόνες είτε θα μπορεί να κάνει λήψη του αρχείου, είτε να το διαγράψει.

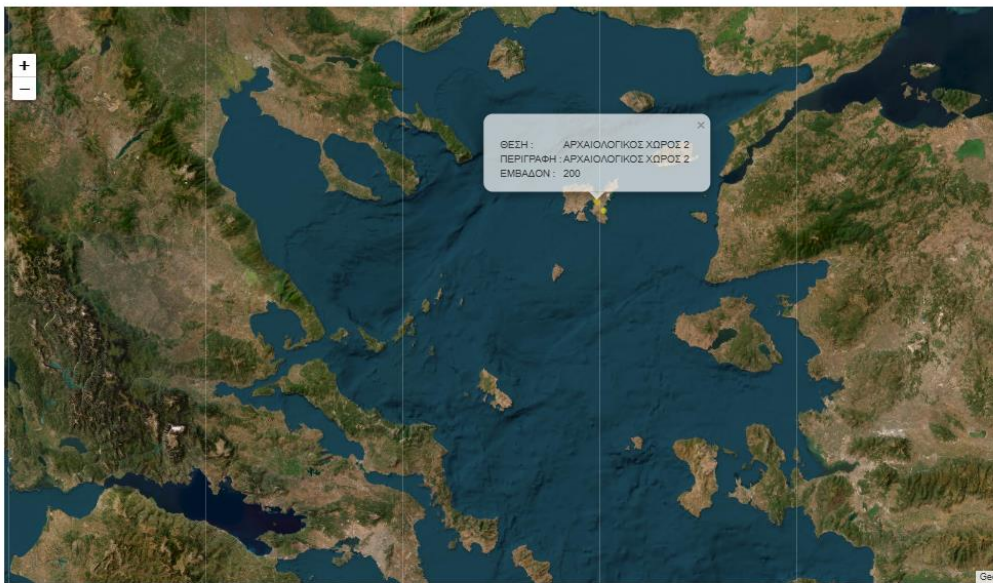


Εικόνα 4.46: Απεικόνιση εικόνας συνημμένου αρχείου και διαγραφής αρχείου σε μία συγκεκριμένη εγγραφή της βάσης δεδομένων της εφαρμογής.

6) Ετικέτα Αρχαιολογικοί χώροι – Ετικέτα Ανασκαφικά Τετράγωνα – Ετικέτα Δρόμοι – Ετικέτα Γεωδαιτικό Δίκτυο – Ετικέτα Χώροι – Ετικέτα Ανασκαφικές Ομάδες – Ετικέτα Στρώσεις – Ετικέτα Τοίχοι – Ετικέτα Κατασκευές: Με κάθε μία από αυτές τις ενέργειες εκτελείται το υποσύστημα των Λειτουργιών CRUD σε συνδυασμό με το υποσύστημα του Χαρτογραφικού Υποβάθρου.

ΧΑΡΤΗΣ ΧΡΗΣΤΕΣ **ΑΡΧΑΙΟΛΟΓΙΚΟΙ ΧΩΡΟΙ** ΤΕΤΡΑΓΩΝΟ ΔΡΟΜΟΙ ΓΕΩΔΑΙΤΙΚΟ ΔΙΚΤΥΟ ΧΩΡΟΙ ΑΝΑΣΚΑΦΙΚΗ ΟΜΑΔΑ ΣΤΡΩΣΕΙΣ ΕΥΡΗΜΑΤΑ ΤΟΙΧΟΙ ΚΑΤΑΣΚΕΥΕΣ ΧΩΡΙΚΑ ΕΡΩΤΗΜΑΤΑ Hello admin@admin.gr! Logout

ΑΡΧΑΙΟΛΟΓΙΚΟΣ ΧΩΡΟΣ

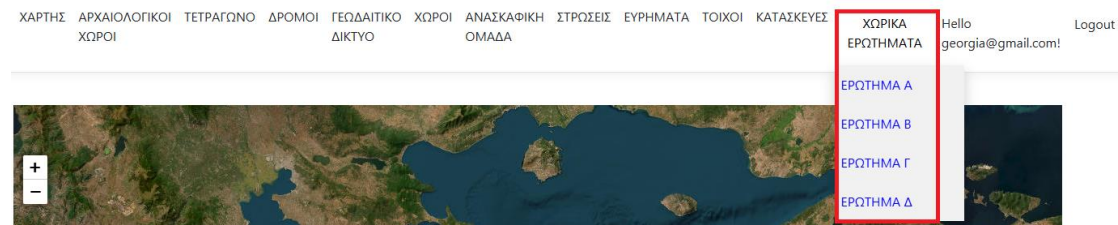


ΚΑΤΑΧΩΡΗΣΗ ΝΕΟΥ ΑΡΧΑΙΟΛΟΓΙΚΟΥ ΧΩΡΟΥ					
ΘΕΣΗ	ΠΕΡΙΓΡΑΦΗ	ΣΥΝΤΕΤΑΓΜΕΝΕΣ Χ(m),Υ(m) στο ΕΓΣΑ'87 (πολύγωνο)	ΕΜΒΑΔΟΝ (m ²)		
Αρχαιολογικός χώρος 1	Σύντομη Περιγραφή	614674.7425,4412089.2722,614670.5408,4412121.7798,614688.4320,4412126.9946	1762.328	Επεξεργασία	Διαγραφή
ΑΡΧΑΙΟΛΟΓΙΚΟΣ ΧΩΡΟΣ 2	ΑΡΧΑΙΟΛΟΓΙΚΟΣ ΧΩΡΟΣ 2	610190.76,4417678.70,610824.44,4417428.67,610565.15,4417251.40	200	Επεξεργασία	Διαγραφή

Εικόνα 4.47: Απεικόνιση διεπαφής χρήστη της οντότητας 'Αρχαιολογικός Χώρος' της βάσης δεδομένων της εφαρμογής.

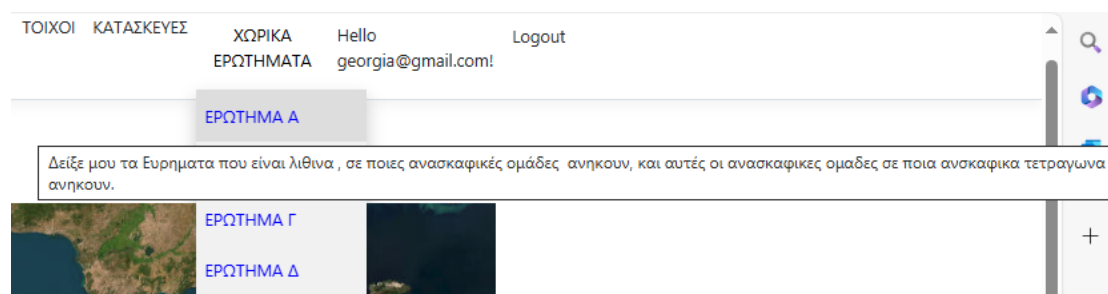
7) Ετικέτα Χωρικά Ερωτήματα: Με αυτή την ενέργεια καλείται το υποσύστημα των Λειτουργιών CRUD σε συνδυασμό με το υποσύστημα του Χαρτογραφικού Υποβάθρου και το υποσύστημα των Χωρικών Ερωτημάτων.

Ο χρήστης από το κεντρικό μενού της εφαρμογής επιλέγει την ετικέτα «Χωρικά Ερωτήματα» και αυτόματα εμφανίζεται ένα dropdown μενού με τα διαθέσιμα χωρικά ερωτήματα της εφαρμογής.

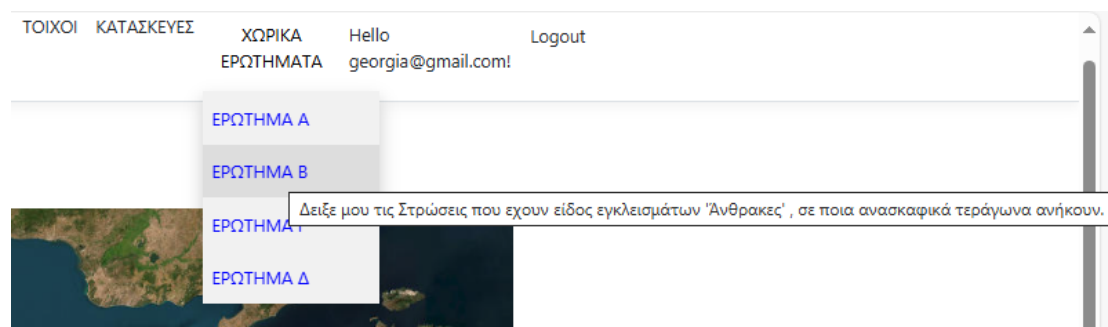


Εικόνα 4.48: Απεικόνιση του dropdown μενού των χωρικών ερωτημάτων της εφαρμογής.

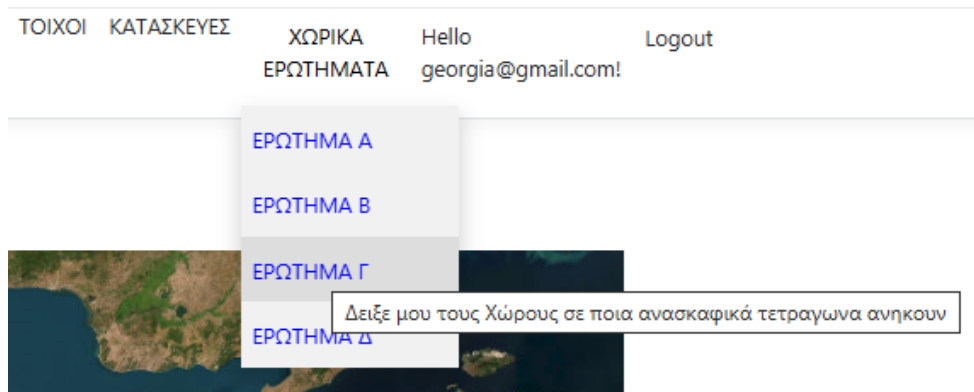
Τα χωρικά ερωτήματα στο μενού εμφανίζονται ως υπερσύνδεσμοι, όπου με το πάσασμα του 'ποντικιού' του υπολογιστή πάνω από κάθε υπερσύνδεσμο εμφανίζεται ένα pop up παράθυρο με το θέμα του χωρικού ερωτήματος.



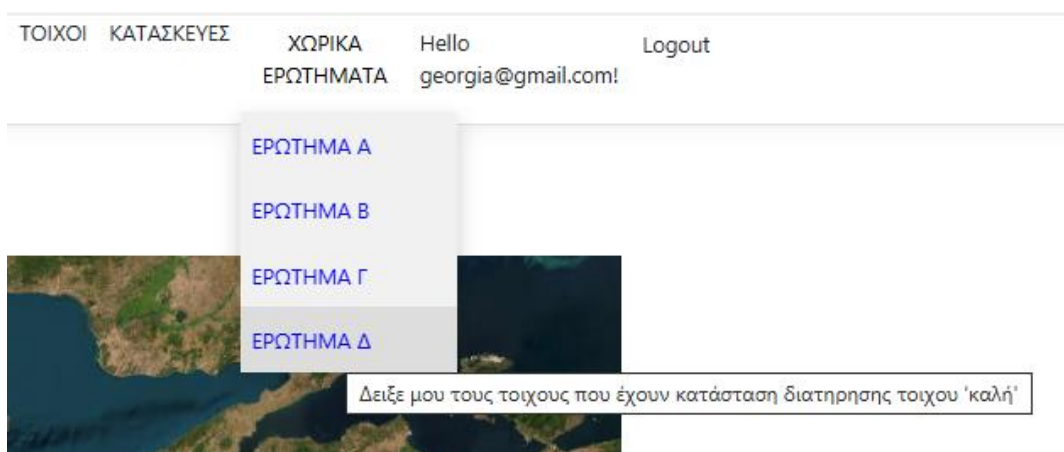
Εικόνα 4.49: Απεικόνιση της pop up λειτουργίας εμφάνισης του θέματος του ερωτήματος Α στο μενού των χωρικών ερωτημάτων της εφαρμογής.



Εικόνα 4.50: Απεικόνιση της pop up λειτουργίας εμφάνισης του θέματος του ερωτήματος Β στο μενού των χωρικών ερωτημάτων της εφαρμογής.



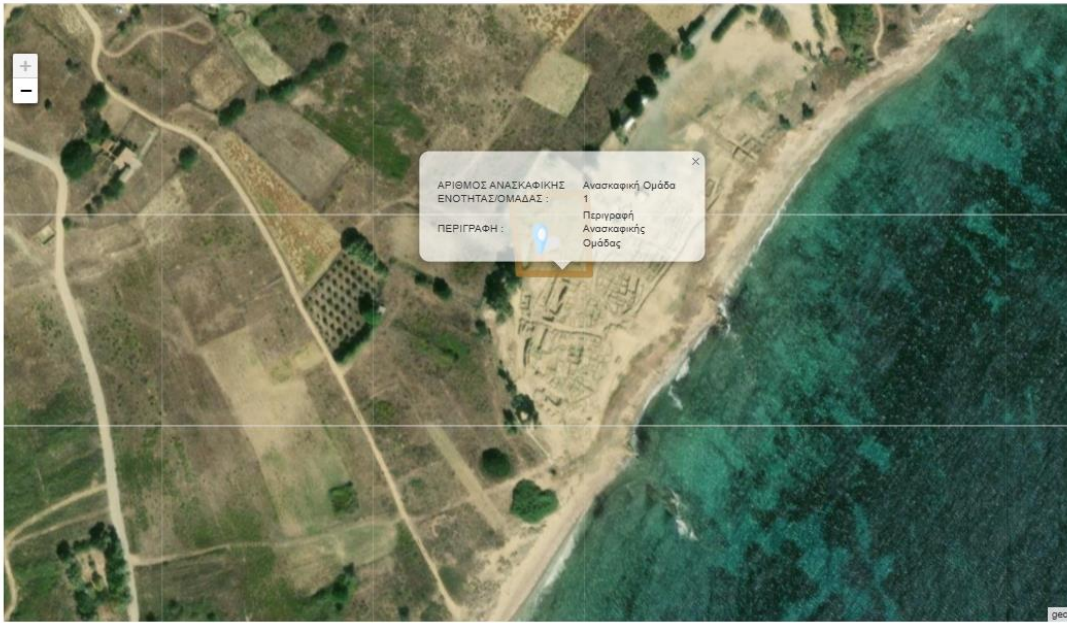
Εικόνα 4.51: Απεικόνιση της pop up λειτουργίας εμφάνισης του θέματος του ερωτήματος Γ στο μενού των χωρικών ερωτημάτων της εφαρμογής.



Εικόνα 4.52: Απεικόνιση της pop up λειτουργίας εμφάνισης του θέματος του ερωτήματος Δ στο μενού των χωρικών ερωτημάτων της εφαρμογής.

Με την επιλογή κάθε υπερσυνδέσμου, εμφανίζεται μία σελίδα - προβολή, η οποία στο επάνω μέρος έχει το χάρτη στον οποίο αποτυπώνονται τα αποτελέσματα του χωρικού ερωτήματος και στο κάτω μέρος εμφανίζεται ένας πίνακας, του οποίου οι στήλες περιέχουν τα στοιχεία του χωρικού ερωτήματος και οι γραμμές την περιγραφική πληροφορία των αποτελεσμάτων του χωρικού ερωτήματος.

ΕΡΩΤΗΜΑ Α



Δείξε μου τα Ευρήματα που είναι λιθίνα , σε ποιες ανασκαφικές ομάδες ανήκουν, και αυτές οι ανασκαφικές ομάδες σε ποια ανασκαφικά τετράγωνα ανήκουν.

ΕΥΡΗΜΑ	ΥΛΙΚΟ	ΑΝΑΣΚΑΦΙΚΗ ΟΜΑΔΑ	ΤΕΤΡΑΓΩΝΟ
ΕΕ_2	Λίθος	Ανασκαφική Ομάδα 1	ΤΕΤΡΑΓΩΝΟ 1
ΕΕ_3	Λίθος	Ανασκαφική Ομάδα 2	ΤΕΤΡΑΓΩΝΟ 1

Εικόνα 4.53: Απεικόνιση της σελίδας εμφάνισης του αποτελέσματος του ερωτήματος Α των χωρικών ερωτημάτων της εφαρμογής.

5. Συμπερασματικά Σχόλια

Τα συμπεράσματα, μετά την ολοκλήρωση της εφαρμογής, επικεντρώνονται κυρίως σε ελέγχους που έγιναν σε θέματα λειτουργικότητας του πληροφοριακού συστήματος.

Ο πρώτος έλεγχος που έγινε ως προς την λειτουργικότητα της εφαρμογής, αφορούσε την βασικότερη επιδίωξη της που ήταν η καταχώρηση και η αποθήκευση της πληροφορίας από μη εξοικειωμένους χρήστες στις βάσεις δεδομένων. Για να επιτευχθεί αυτό τεχνικά, θα πρέπει να υπάρχει αποτελεσματική επικοινωνία μεταξύ της εφαρμογής και του συστήματος διαχείρισης της βάσης δεδομένων. Πιο συγκεκριμένα, ελέγχθηκε η πληροφορία της βάσης που υπάρχει στην εφαρμογή, εάν είναι ακριβώς η ίδια με αυτή που είναι αποθηκευμένη στο σύστημα διαχείρισης της βάσης δεδομένων. Διαπιστώθηκε ότι υπάρχει απόλυτη επικοινωνία μεταξύ τους, εκτελώντας από την εφαρμογή όλες τις λειτουργίες CRUD (Δημιουργία, Ανάγνωση, Ενημέρωση, Διαγραφή) σε όλες τις οντότητες – πίνακες της βάσης δεδομένων.

Στη συνέχεια, παρουσιάζεται ένα στιγμιότυπο εικόνας του πίνακα των Ευρημάτων της εφαρμογής και ένα στιγμιότυπο εικόνας του ίδιου πίνακα στο σύστημα διαχείρισης της βάσης δεδομένων.

ΕΙΔΟΣ ΣΥΦΥΜΑ (ΕΕ)	ΗΜΕΡΟΜΗΝΙΑ	ΕΙΔΟΣ	ΥΨΟΜΕΤΡΟ	*ΣΥΝΤΕΤΑΓΜΕΝΕΣ Χ(m),Υ(m) στο ΕΓΓΛΒ*	ΜΗΚΟΣ σε (cm)	ΠΛΑΤΟΣ σε (cm)	ΥΨΟΣ σε (cm)	ΧΡΟΝΟΛΟΓΙΣΗ	ΥΛΙΚΟ	ΚΑΤΑΣΤΑΣΗ ΔΙΑΤΗΡΗΣΗΣ	ΠΑΡΑΤΗΡΗΣΕΙΣ	ΣΥΝΤΑΚΤΗΣ	Ανασκαφική Ομάδα	YOUTUBE	FILE	Επιμέτρηση	Διαγραφή
EE_2	11/6/2022 12:00:00	Σφονδύλι	15,95	614685.4706,4412091.9069	1	1	3	Χρονολόγηση του ΕΕ	Λίθος	Αέρισο	Παρατηρήσεις για το ΕΕ	Κώστας Παπαδόπουλος	Ανασκαφική Ομάδα 1			Επιμέτρηση	Διαγραφή
EE_3	2/6/2022 12:00:00	Σφονδύλι	16,05	614687.0314,4412091.7527	1,3	1,2	2,5	Χρονολόγηση του ΕΕ	Λίθος	Ήμισο	Παρατηρήσεις για το ΕΕ	Κώστας Παπαδόπουλος	Ανασκαφική Ομάδα 2			Επιμέτρηση	Διαγραφή
EE_4	2/6/2022 12:00:00	Όνομα ΕΕ	16	614687.9927,4412091.7409	1	1	2	Χρονολόγηση του ΕΕ	Μέταλλο	Ένα τρίτο	Παρατηρήσεις για το ΕΕ	Κώστας Παπαδόπουλος	Ανασκαφική Ομάδα 2			Επιμέτρηση	Διαγραφή
EE_5	2/6/2022 12:00:00	Όνομα ΕΕ	15,85	614687.4646,4412092.5711	0,5	0,5	1	Χρονολόγηση του ΕΕ	Οργανικό	Δύο τρίτα	Παρατηρήσεις για το ΕΕ	Κώστας Παπαδόπουλος	Ανασκαφική Ομάδα 2			Επιμέτρηση	Διαγραφή
EE_11	15/6/2022 12:00:00	Όνομα ΕΕ	15,8	614686.6596,4412095.8881	1,5	1,5	1	Χρονολόγηση του ΕΕ	Οργανικό	Δύο τρίτα	Παρατηρήσεις για το ΕΕ	Κώστας Παπαδόπουλος	Ανασκαφική Ομάδα 3			Επιμέτρηση	Διαγραφή
EE_12	20/6/2022 12:00:00	Όνομα ΕΕ	15,78	614687.9561,4412096.2819	1,3	1	2,8	Χρονολόγηση του ΕΕ	Μέταλλο	Ένα τρίτο	Παρατηρήσεις για το ΕΕ	Κώστας Παπαδόπουλος	Ανασκαφική Ομάδα 1			Επιμέτρηση	Διαγραφή
EE_58	2/7/2024 10:47:00	ΠΕΡΟΝΗ	10,52	614278.189,4415346.525	1,2	1,5	0,5	Η Χρονολόγηση του ειδικού ευρήματος	Μέταλλο	Αέρισο	Οι παρατηρήσεις του ειδικού ευρήματος	Γεωργία Ταμπακοπούλου	Ανασκαφική Ομάδα 3			Επιμέτρηση	Διαγραφή

Εικόνα 4.54: Απεικόνιση του πίνακα των εγγραφών των Ευρημάτων της εφαρμογής.

	Name	DateFinding	TypeFinding	Altitude	Coordinates	LengthFinding
1	EE_2	2022-06-11 12:00:00.000	Σφονδύλι	15,95	614685.4706,4412091.9069	1
2	EE_3	2022-06-02 12:00:00.000	Σφονδύλι	16,05	614687.0314,4412091.7527	1,3
3	EE_4	2022-06-02 12:00:00.000	Όνομα ΕΕ	16	614687.9927,4412091.7409	1
4	EE_5	2022-06-02 12:00:00.000	Όνομα ΕΕ	15,85	614687.4646,4412092.5711	0,5
5	EE_11	2022-06-15 12:00:00.000	Όνομα ΕΕ	15,8	614686.6596,4412095.8881	1,5
6	EE_12	2022-06-20 12:00:00.000	Όνομα ΕΕ	15,78	614687.9561,4412096.2819	1,3
7	EE_58	2024-07-02 10:47:00.000	ΠΕΡΟΝΗ	10,52	614278.189,4415346.525	1,2

Εικόνα 4.55: Απεικόνιση του πίνακα των εγγραφών των Ευρημάτων στο σύστημα διαχείρισης της βάσης δεδομένων (Sql Server).

Ο δεύτερος έλεγχος αφορούσε το κατά πόσο είναι ορθή η αποτύπωση των δεδομένων στο χάρτη. Βασικό ζητούμενο ήταν το πληροφοριακό σύστημα να λαμβάνει συντεταγμένες στο ΕΓΣΑ' 87, έτσι ώστε η πληροφορία της βάσης να είναι άμεσα συσχετίσιμη με οποιαδήποτε χωρική πληροφορία αφορά τον Ελλαδικό χώρο. Για το σκοπό αυτό, όπως ήδη έχει αναφερθεί, δημιουργήθηκε συνάρτηση μετατροπής συντεταγμένων.

Τεχνικά ο έλεγχος πραγματοποιήθηκε, δημιουργώντας πολύγωνα στο ΕΓΣΑ'87 σε χαρακτηριστικά σημεία στην υπηρεσία 'Θέασης' της Κτηματολόγιο ΑΕ και έχοντας τη χωρική πληροφορία των πολυγώνων καταχωρούνταν ως εγγραφές στη βάση δεδομένων μέσω της εφαρμογής. Μετά από αυτή τη διαδικασία δεν παρατηρήθηκαν εμφανείς διαφορές, οπότε θεωρείται ορθή η αποτύπωση της χωρικής πληροφορίας. Σε περίπτωση που ήταν επιθυμητή η διαπίστωση μεγαλύτερης ακρίβειας στην αποτύπωση της χωρικής πληροφορίας, θα ήταν αναγκαία η προσθήκη της λειτουργίας μέτρησης απόστασης στο χάρτη, έτσι ώστε να είναι δυνατή η σύγκριση της αποτυπωθείσας πληροφορίας με αυτή που προκύπτει από τον υπολογισμό της απόστασης βάσει των συντεταγμένων των εγγραφών της βάσης.

Στη συγκεκριμένη περίπτωση δεν υπάρχει απαίτηση για μεγαλύτερη ακρίβεια αποτύπωσης, γι' αυτό ο έλεγχος επικεντρώνεται κυρίως στην παρατήρηση χονδροειδών σφαλμάτων.

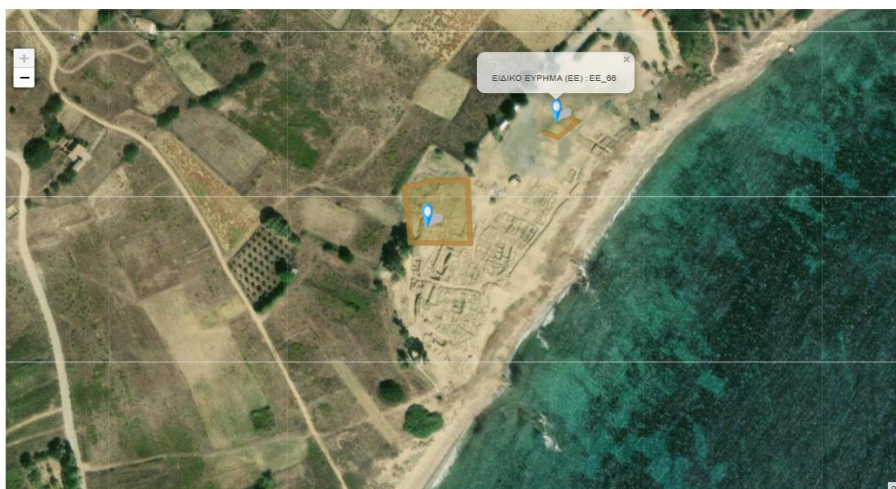


Εικόνα 4.56: Απεικόνιση πολυγώνου στο χαρτογραφικό υπόβαθρο της εφαρμογής (αριστερά) και στην υπηρεσία 'Θέασης' της Κτηματολόγιο ΑΕ (δεξιά).

Ο τρίτος έλεγχος αφορούσε την ορθότητα των αποτελεσμάτων των χωρικών ερωτημάτων. Λόγω της πολύ περιορισμένης πληροφορίας που υπάρχει στη βάση δεδομένων, η ορθότητά τους ελέγχθηκε ανατρέχοντας στους αντίστοιχους πίνακες – οντότητες και ελέγχοντας με αυτό τον τρόπο τα αποτελέσματα. Ενδεικτικά, θα παρουσιαστεί ο έλεγχος του Α Ερωτήματος.

Αρχικά, ελέγχεται το αποτέλεσμα του Ερωτήματος Α στη διεπαφή χρήστη τόσο στο χάρτη όσο και στον πίνακα.

ΕΡΩΤΗΜΑ Α



Δείξε μου τα Ευρήματα που είναι λίθινα, σε ποιες ανασκαφικές ομάδες ανήκουν, και αυτές οι ανασκαφικές ομάδες σε ποια ανασκαφικά τετράγωνα ανήκουν.

ΕΥΡΗΜΑ	ΥΛΙΚΟ	ΑΝΑΣΚΑΦΙΚΗ ΟΜΑΔΑ	ΤΕΤΡΑΓΩΝΟ
ΕΕ_2	Λίθος	Ανασκαφική Ομάδα 1	ΤΕΤΡΑΓΩΝΟ 1
ΕΕ_66	Λίθος	Ανασκαφική Ομάδα 5	23.88

Εικόνα 4.57: Απεικόνιση διεπαφής χρήστη του Ερωτήματος Α στην εφαρμογή.

Μετά τον έλεγχο στη συγκεκριμένη διεπαφή διαπιστώνεται ότι ο χάρτης αποτυπώνει τις εγγραφές του πίνακα.

Στη συνέχεια, ελέγχεται ο πίνακας των Ευρημάτων, για το αν τα ευρήματα ΕΕ_2 και ΕΕ_66 είναι λίθινα και αν το εύρημα ΕΕ_2 ανήκει στην Ανασκαφική Ομάδα 1 και το εύρημα ΕΕ_66 ανήκει στην Ανασκαφική Ομάδα 5. Διαπιστώνεται ότι τα παραπάνω ισχύουν, όπως απεικονίζεται στο στιγμιότυπο εικόνας του πίνακα των Ευρημάτων που ακολουθεί.

ΚΑΤΑΧΩΡΗΣΗ ΝΕΟΥ ΕΥΡΗΜΑΤΟΣ													
ΕΙΔΙΚΟ ΕΥΡΗΜΑ (ΕΕ)	ΗΜΕΡΟΜΗΝΙΑ	ΕΙΔΟΣ	ΥΨΟΜΕΤΡΟ	ΣΥΝΤΕΤΑΓΜΕΝΕΣ Χ(μ),Υ(μ) στο ΕΓΣΑ'87	ΜΗΚΟΣ σε (cm)	ΠΛΑΤΟΣ σε (cm)	ΥΨΟΣ σε (cm)	ΧΡΟΝΟΛΟΓΗΣΗ	ΥΛΙΚΟ	ΚΑΤΑΣΤΑΣΗ ΔΙΑΤΗΡΗΣΗΣ	ΠΑΡΑΤΗΡΗΣΕΙΣ	ΣΥΝΤΑΚΤΗΣ	Ανασκαφική Ομάδα
ΕΕ_2	11/6/2022 12:00:00	Σφρονδύλι	15,95	614685.4706,4412091.9069	1	1	3	Χρονολόγηση του ΕΕ	Λίθος	Ακέραιο	Παρατηρήσεις για το ΕΕ	Κώστας Παπαδόπουλος	Ανασκαφική Ομάδα 1
ΕΕ_66	1/7/2024 01:13:00	ΠΕΡΟΝΗ	0	614769.389,4412168.08	2	1	1,1	Η Χρονολόγηση του ειδικού ευρήματος	Λίθος	Ακέραιο	Οι παρατηρήσεις του ειδικού ευρήματος	Γεωργία Ταμπακοπούλου	Ανασκαφική Ομάδα 5

Εικόνα 4.58: Απεικόνιση ενός μέρους του πίνακα των Ευρημάτων στην εφαρμογή.

Έπειτα, ακολούθησε ο έλεγχος στο πίνακα 'Ανασκαφική Ομάδα', όπου διαπιστώθηκε ότι η Ανασκαφική Ομάδα 1 ανήκει στο Τετράγωνο 1 και η Ανασκαφική Ομάδα 5 ανήκει στο Τετράγωνο 23.88.

Τέλος, θα πρέπει να αναφερθεί ότι πιο ασφαλή συμπεράσματα θα εξαγονταν σε σχέση με τη λειτουργικότητα της εφαρμογής, εάν δινόταν προς χρήση μία δοκιμαστική της έκδοση, στην οποία θα καταχωρούνταν πραγματικά ανασκαφικά δεδομένα και θα ελέγχονταν η λειτουργικότητά της σε πραγματικές συνθήκες.

5.1 Μελλοντικές Βελτιώσεις

Σε αυτή την ενότητα προτείνονται κάποιες μελλοντικές βελτιώσεις που κυρίως αφορούν την προσθήκη διάφορων αυτοματισμών, ενισχύοντας με αυτόν τον τρόπο την ορθότητα της πληροφορίας των δεδομένων, την αποτελεσματικότερη θέασή τους και τη στατιστική ανάλυση της πληροφορίας της βάσης με τον υπολογισμό κάποιων μεγεθών που θεωρείται κρίσιμο να μελετηθούν.

Μία μελλοντική βελτίωση που προτείνεται είναι η καταχώρηση των συντεταγμένων των εγγραφών να γίνεται με φόρτωση αρχείου και όχι με την πληκτρολόγηση από τον χρήστη. Οι συντεταγμένες είναι αριθμοί πολλών ψηφίων, με αποτέλεσμα η πληκτρολόγησή τους να είναι μία διαδικασία επιρρεπής σε λάθη.

Επίσης, στις οντότητες – πίνακες που υπάρχει το πεδίο 'Εμβαδόν' προτείνεται, να έπεται αυτού των 'Συντεταγμένων' στη φόρμα εισαγωγής και να υπολογίζεται αυτόματα από την εφαρμογή βάσει της πληροφορίας που ήδη θα έχει καταχωρηθεί από τις Συντεταγμένες.

Μία ακόμη μελλοντική βελτίωση που προτείνεται είναι η καταχώρηση και η αποθήκευση της τρισδιάστατης πληροφορίας της βάσης δεδομένων και η αξιοποίησή της μέσω της δημιουργίας σχεδίων και συγκεκριμένα τομών των ανασκαφικών τετραγώνων με αυτοματοποιημένο τρόπο. Στα σχέδια αυτά, θα αποτυπώνεται σε τομή όλη η υψομετρική πληροφορία των ευρημάτων της ανασκαφής για το συγκεκριμένο ανασκαφικό τετράγωνο.

Σημαντική βελτίωση θα ήταν η δημιουργία μίας επιπλέον σελίδας στο κεντρικό μενού της εφαρμογής, η οποία θα οπτικοποιεί κάποιους στατιστικούς δείκτες που θα αφορούν συγκεκριμένα δεδομένα της χωρικής βάσης. Υπάρχει συχνά ενδιαφέρον από τους αρχαιολόγους, υπολογισμού της συχνότητας εμφάνισης κάποιων ειδικών ευρημάτων, του υπολογισμού του μέσου όρου της κατάστασης διατήρησης κάποιων αντικειμένων, του υπολογισμού του ποσοστού κάποιας κατηγορίας ευρήματος επί του συνολου των ευρημάτων κ.α. Επομένως, θα επιτυγχάνεται με αυτόν τον τρόπο μία στατιστική ανάλυση των δεδομένων σε πρώτο βαθμό της ανασκαφικής διαδικασίας, αυτόματα μετά την ολοκλήρωση της καταχώρησή τους στη βάση δεδομένων.

Τέλος, προτείνεται η βελτίωση του γραφικού περιβάλλοντος της διεπαφής χρήστη και η μορφοποίηση της σε αποκριτική (responsive) σε όλες τις υποσελίδες της, έτσι ώστε να διευκολύνεται η θέασή της από όλες τις κινητές συσκευές.

6. Βιβλιογραφία

Γεωργόπουλος Α., Ταπεινάκη Σ., Αγραφιώτης Π., Δεληγιάννης Ι., Μητροπούλου Αικ, **Οι σύγχρονες τεχνολογίες για την υποστήριξη του ανασκαφικού έργου ανάδειξης του αρχαιολογικού χώρου της Πολιόχνης στην Λήμνο**, Πρακτικά Ημερίδας Ελληνικού ICOMOS (υπό δημοσίευση)

Λιανός Μ., 2023, **Ανάπτυξη διαδικτυακής βάσης δεδομένων για την υποστήριξη ανασκαφικού έργου**, Διπλωματική Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα, Μάρτιος 2023

Hatzithoma – Panagiotou S., Georgopoulos A., Tapinaki S., 2014, **DEVELOPMENT OF A CULTURAL HERITAGE MANAGEMENT SYSTEM-IMPLEMENTATION TO THE ANCIENT ATHENIAN AGORA**, 1ST CAA GR CONFERENCE, Rethymno, Crete, Greece 2014

Tapinaki S., Georgopoulos A., Ioannidis C., Frentzos E., Stampolidis N., Maragoudakis N., 2016, **DEVELOPMENT OF A GEOREFERENCED ARCHAEOLOGICAL INFORMATION DATA BASE FOR ELEUTHERNA IN CRETE**, Proceedings of the 8th International Congress on Archaeology, Computer Graphics, Cultural Heritage and Innovation 'ARQUEOLÓGICA 2.0' in Valencia (Spain), Sept. 5 – 7, 2016

ΙΣΤΟΣΤΕΛΙΑΔΕΣ

Μαργαρίτης Κ., Τσιομπίκας Γ. (2004), **Εισαγωγή στη C++**, (http://www.ebooks4greeks.gr/2011.Download_free-ebooks/Pliroforikis/glossa_programmatismoy_C++_eBooks4Greeks.gr.pdf, τελευταία επίσκεψη: 10/05/2024).

GeeksforGeeks (<https://www.geeksforgeeks.org/introduction-to-c-sharp/>, last visit 4/6/2024)

GeeksforGeeks (<https://www.geeksforgeeks.org/mvc-design-pattern/>, last visit 10/6/2024)

OsGeo (https://live.osgeo.org/archive/10.5/el/overview/leaflet_overview.html, last visit 15/6/2024)

TechTarget (<https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app>, last visit 29/5/2024)

Ramotion (<https://www.ramotion.com/blog/responsive-web-application-development/>, last visit 31/5/2024)