



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΑΠΟΦΑΣΕΩΝ

Σύστημα για την παρακολούθηση και διαχείριση έργων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ESENDER OBEY

Επιβλέπων: Ιωάννης Ψαρράς
Καθηγητής ΕΜΠ

Αθήνα, Ιούλιος 2012



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΑΠΟΦΑΣΕΩΝ

Σύστημα για την παρακολούθηση και διαχείριση έργων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ESENDA OBEY

Επιβλέπον: Ιωάννης Ψαρράς
Καθηγητής ΕΜΠ

Εγκρίθηκε από τη τριμελή επιτροπή την

Ιωάννης Ψαρράς
Καθηγητής ΕΜΠ

Δημήτριος Ασκούνης
Αναπληρωτής Καθηγητής ΕΜΠ

Βασίλειος Ασημακόπουλος
Καθηγητής ΕΜΠ

Αθήνα, Ιούλιος 2012

ESENDA OBEY

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π

Copyright © Esenda Obey. 2012

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Πρόλογος

Υπεύθυνος καθηγητής στη εκπόνηση της διπλωματικής ήταν ο κ. Ι. Ψαρράς. Θα ήθελα να ευχαριστήσω ιδιαίτερα τον επιβλέποντα της διπλωματικής κ. Χ. Τσαβδάρη για την υποστήριξη και την καθοδήγηση που παρείχε, καθώς και για την διαρκή ενθάρρυνση και την εμπιστοσύνη που επέδειξε στην προσπάθειά μου.

Esenda Obey

ΠΕΡΙΛΗΨΗ

Στόχος της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η υλοποίηση μιας διαδικτυακής εφαρμογής διαχείρισης των έργων. Το σύστημα θα αποτελεί μια ιστοσελίδα η οποία θα περιέχει όλα τα έργα που έχει αναλάβει μια εταιρία, τις απαιτήσεις του κάθε έργου για ανθρώπινο δυναμικό και τα εβδομαδιαία καθήκοντα των εργαζομένων.

Οι σύγχρονες ανάγκες στον τομέα της διαχείρισης έργων απαιτούν μεγάλες ομάδες εργασιών οι οποίες οργανώνονται από ένα μεγάλο πλήθος υπεύθυνων έργων. Επειδή τα άτομα που απαρτίζουν την ομάδα είναι περιορισμένα σε πλήθος η ορθή κατανομή των πόρων της ομάδας (που στην προκείμενη περίπτωση αποτελούν τα ίδια τα μέλη της) πρέπει να γίνεται με προσοχή για την βέλτιστη αξιοποίηση των πόρων.

Τα εργαλεία που θα προσφέρει η παρούσα εφαρμογή απευθύνονται κυρίως στους συντονιστές των ομάδων εργασίας με σκοπό τον ορθό και γρήγορο προγραμματισμό των υποχρεώσεων τους σε εβδομαδιαίο επίπεδο

ABSTRACT

Main objective of the present diplomatic thesis is the planning and the concretization of a web based application for project management. The system will constitute a web page which will contain all projects that have been undertaken by a company, the requirements of each project for human resources and the weekly duties of employees.

Current needs in the management of projects require major working groups be organized by a large number of project managers. Because resources (which in this case are the group members) are limited, allocation should be done with proper care to make an optimum use of resources.

The application which will be developed in this thesis is mainly addressed to team coordinators aiming at the equitable and fast planning for their team members' weekly obligations.

Πίνακας περιεχομένων

Κεφάλαιο 1: Εισαγωγή

1.1 Διαχείριση έργων

1.2 Στελέχωση έργου

1.2.1 Διευθυντής έργου (ΥΣ)

1.2.2 Υπεύθυνος έργου – Project manager

1.2.3 Ομάδα έργου – Project team

1.3 Ορισμός του προβλήματος

Κεφάλαιο 2: Ρόλοι

2.1.1 Ρόλοι συστήματος

2.1.2 Ρόλοι έργων

Κεφάλαιο 3: Παρουσίαση

3.1 Η διαδικασία του Login

3.2 Η κεντρική σελίδα: η λίστα των έργων

3.3 Η οθόνη έργου

Καρτέλα E1

Καρτέλα E2

Καρτέλες: E3 Financial deliverables, E4 Technical deliverables

Καρτέλα E4: Project tasks and resources

Καρτέλα 3 Years

Καρτέλα E5

Η δημιουργία έργου

Η διαγραφή έργου

3.4 Η οθόνη χρήστη

Δημιουργία νέου χρήστη

Παράρτημα

Η βάση δεδομένων

Η τεχνολογία Silverlight

Η τεχνολογία WCF RIA Services

Δείγμα υλοποιημένου κώδικα

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

1.1 Διαχείριση Έργων

Η διαχείριση έργων είναι η εφαρμογή επιστημονικών τεχνικών στον σχεδιασμό και την ανάλυση των έργων, καθώς και στον έλεγχο της πορείας τους. Είναι η επίτευξη των σκοπών του έργου με τους διαθέσιμους πόρους: τον χρόνο, τον χώρο, τα χρήματα, τα υλικά, τον εξοπλισμό, και ειδικά τους ανθρώπους.

Έργο είναι οποιαδήποτε προσχεδιασμένη εργασία που περιλαμβάνει ένα αριθμό ανθρώπων και δραστηριοτήτων. Μπορεί να οριστεί σαν ένα σύνολο δραστηριοτήτων το οποίο:

- Πρέπει να ικανοποιήσει ένα συγκεκριμένο στόχο με το πέρας του
- Έχει προκαθορισμένες ημερομηνίες έναρξης και πέρατος
- Έχει πιθανόν περιορισμένη χρηματοδότηση
- Καταναλώνει πόρους (χρήματα, ανθρωποχρόνο, εξοπλισμός) κατά τη διάρκεια της εκτέλεσής του.

Έτσι η διαχείριση έργων αφορά τον σχεδιασμό, την οργάνωση, την παρακολούθηση της εκτέλεσης και τον έλεγχο των πόρων με σκοπό την επίτευξη ενός συγκεκριμένου και προκαθορισμένου στόχου. Όσον αφορά τον σχεδιασμό περιλαμβάνει τον καθορισμό των απαιτήσεων εργασίας, τον ποσοτικό προσδιορισμό της αναγκαίας εργασίας και τον καθορισμό των απαραίτητων πόρων. Όσο αφορά την παρακολούθηση και τον έλεγχο περιλαμβάνει την καταγραφή της εξέλιξης της εκτέλεσης του έργου, την σύγκριση προβλεπόμενης και πραγματικής εξέλιξης, την ανάλυση επιπτώσεων και τον καθορισμό των αναγκαίων προσαρμογών στον σχεδιασμό.

Η επιτυχή διαχείριση έργων προϋποθέτει την επίτευξη των στόχων του έργου στα προκαθορισμένα χρονικά πλαίσια, στα πλαίσια του προκαθορισμένου κόστους, στο επιθυμητό επίπεδο απόδοσης / τεχνολογίας και με την αποδοτική και αποτελεσματική χρήση των πόρων που ανατέθηκαν στην εκτέλεση του έργου.

Τα οφέλη από τη σωστή διαχείριση έργων είναι:

- Προσδιορισμός των υπευθυνοτήτων ανά επιμέρους δραστηριότητα
- Ελαχιστοποίηση της ανάγκης συνεχών αναφορών (reporting)
- Προσδιορισμός των χρονικών περιθωρίων για την εκτέλεση κάθε δραστηριότητας
- Ποσοτικοποίηση των πραγματοποιηθέντων δραστηριοτήτων σε σχέση με τα προγραμματισθέντα
- Έγκαιρη αναγνώριση πιθανών προβλημάτων, ώστε να είναι δυνατόν να ξεκινήσουν διορθωτικές ενέργειες
- Βελτίωση της δυνατότητας εκτιμήσεων για τον μελλοντικό σχεδιασμό των δραστηριοτήτων του ίδιου ή άλλων έργων
- Έγκαιρη γνώση των επιτεύξιμων στόχων, καθώς και της πιθανότητας να ξεπεραστούν οι περιορισμοί (κόστους, διαθέσιμου δυναμικού κλπ)

1.2 Στελέχωση Έργου

Η επιτυχής διοίκηση και διαχείριση ενός έργου εξαρτάται σε μεγάλο βαθμό εκτός από την οργανωτική δομή από τις ικανότητες του διευθυντή έργου, του υπευθύνου του έργου και των μελών της ομάδας έργου, που διαχειρίζονται και διοικούν τις βασικές λειτουργίες. Η διοίκηση ενός έργου δεν είναι μια διαδικασία ενός και μόνο ανθρώπου. Αντίθετα απαιτεί τον συντονισμό μιας ομάδας ανθρώπων που είναι αφοσιωμένοι στην επίτευξη των στόχων του έργου.

1.2.1 Διευθυντής Έργου (ΥΣ)

Ο υπεύθυνος σύμβασης έχει την ευθύνη διαχείρισης της σύμβασης για την οποία είναι υπεύθυνος σε 'στρατηγικό επίπεδο' και διασφάλισης της ποιότητας των παραδοτέων. Η ευθύνη αυτή αφορά σε όλο τον 'κύκλο ζωής' της σύμβασης και εξειδικεύεται ως εξής:

- Σχεδίαση συμβατικών χαρακτηριστικών του έργου δηλαδή συνεννόηση με όλους τους συνεργαζόμενους φορείς (εντολείς, υπεργολάβοι, εργολάβοι, παραλήπτες), συμφωνία επί των συμβατικών χαρακτηριστικών του έργου (ρόλοι, τεχνικό και οικονομικό αντικείμενο ανά συνεργαζόμενο φορέα κλπ).
- Σύνταξη Σύμβασης
- Υπογραφή Σύμβασης.
- Κατάρτιση / αναθεώρηση προϋπολογισμού έργου.
- Δέσμευση πόρων έργου. Η εξασφάλιση των απαραίτητων πόρων (ανθρώπινο δυναμικό, χώροι, hardware, software, κλπ υποδομή) βάσει του συμφωνηθέντος προγραμματισμού.
- Κατάρτιση συμβατικών παραδοτέων έργου. Κατάρτιση προδιαγραφών ποιότητας των συμβατικών τεχνικών αναφορών του έργου και εμπρόθεσμη υποβολή των αναφορών αυτών.
- Προώθηση αναπτυξιακών δυνατοτήτων έργου. Δημιουργία κλίματος καλής συνεργασίας με τα στελέχη του Εντολέα, διερεύνηση και προώθηση δυνατοτήτων περαιτέρω συνεργασίας

1.2.2 Υπεύθυνος Έργου – Project Manager

Ο υπεύθυνος έργου πρέπει να κατέχει ένα σύνολο από ιδιαίτερα χαρακτηριστικά για να μπορεί να διοικήσει την ομάδα έργου και να φέρει το έργο σε πέρας με επιτυχία.

Οι κύριες υπευθυνότητες του περιλαμβάνουν τις εξής:

- επίτευξη του τελικού στόχου του έργου με τους περιορισμούς των διαθέσιμων πόρων και μέσα στα προκαθορισμένα πλαίσια χρόνου, κόστους και απόδοσης
- λήψη των απαιτούμενων αποφάσεων για κάθε φάση και δραστηριότητα του έργου
- υπεύθυνος για τις σχέσεις με τους υπεύθυνους των λειτουργικών μονάδων

1.2.3 Ομάδα Έργου – Project Team

Η ομάδα έργου αποτελείται από στελέχη που συνήθως προέρχονται από διαφορετικές λειτουργικές μονάδες και καλύπτουν τους διάφορους χώρους τεχνικής και επιστημονικής εξειδίκευσης που χρειάζονται για την επιτυχή υλοποίηση του έργου.

Η στελέχωση της ομάδας έργου γίνεται με ευθύνη του διευθυντή και εξαρτάται από τις συγκεκριμένες απαιτήσεις του έργου, ώστε η ανάθεση συγκεκριμένων δραστηριοτήτων να γίνει στα κατάλληλα άτομα.

1.3 Ορισμός Προβλήματος

Στόχος της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η υλοποίηση μιας διαδικτυακής εφαρμογής διαχείρισης των έργων. Το σύστημα θα αποτελεί μια ιστοσελίδα η οποία θα περιέχει όλα τα έργα που έχει αναλάβει η εταιρία, τις απαιτήσεις του κάθε έργου για ανθρώπινο δυναμικό και τα εβδομαδιαία καθήκοντα των εργαζομένων.

Οι σύγχρονες ανάγκες στον τομέα της διαχείρισης έργων απαιτούν μεγάλες ομάδες εργασιών οι οποίες οργανώνονται από ένα μεγάλο πλήθος υπεύθυνων έργων. Επειδή τα άτομα που απαρτίζουν την ομάδα είναι περιορισμένα σε πλήθος η ορθή κατανομή των πόρων της ομάδας (που στην προκειμένη περίπτωση αποτελούν τα ίδια τα μέλη της) πρέπει να γίνεται με προσοχή για την βέλτιστη αξιοποίηση των πόρων.

Το πρόβλημα της ορθής διαχείρισης ενός έργου μπορεί να διασπαστεί σε δύο μεγάλες ενότητες: τον προγραμματισμό του έργου και την παρακολούθηση του.

Κατά την διάρκεια του προγραμματισμού, οι υπεύθυνοι καλούνται να κάνουν καταμερισμό εργασιών στα μέλη ομάδας εργασίας, με τέτοιο τρόπο ώστε να αποφύγουν φαινόμενα υποαπασχόλησης ή υπεραπασχόλησης. Σε αυτή την διαδικασία, η λέξη κλειδί είναι ο χρόνος που διαθέτουν τα μέλη ομάδας εργασίας. Αυτός ο χρόνος ανάλογα με το είδος του προγραμματισμού που γίνεται μπορεί να μετριέται σε ώρες, ημέρες ή ακόμα και μήνες. Είθισται να χρησιμοποιείται η ανθρωποημέρα ως βασική μονάδα μέτρησης αυτού του χρόνου, μιας και βολεύει τόσο σε μακροχρόνιο σχεδιασμό κάποιων μηνών όσο και σε βραχυχρόνιο σχεδιασμό μιας εβδομάδας.

Η διαδικασία της παρακολούθησης μπορεί να τυποποιηθεί με εβδομαδιαίο απολογισμό των ενεργειών της ομάδας εργασίας. Το κάθε άτομο ενημερώνει τους υπεύθυνους για το τι ποσοστό των ανατεθειμένων ολοκλήρωσε κατά την διάρκεια της προηγούμενης εβδομάδας. Όσο πιο σύντομο είναι το διάστημα που μεσολαβεί από απολογισμό σε απολογισμό τόσο πιο έγκαιρα μπορούν να προβλεφθούν κάποιες ασυνέπειες στην παράδοση έργων και να αντιμετωπισθούν τροποποιώντας τον προγραμματισμό.

ΚΕΦΑΛΑΙΟ 2: ΡΟΛΟΙ

Στο σύστημα υπάρχουν δύο επίπεδα ρόλων που μπορεί να έχουν οι χρήστες. Αυτοί είναι οι ρόλοι στο σύστημα (Επισκέπτης, Χρήστης, Διαχειριστής) και οι ρόλοι στα έργα (Υπεύθυνος Σύμβασης, Υπεύθυνος Έργου και Μέλος Ομάδας Έργου). Αυτά τα δύο επίπεδα ρόλων είναι παράλληλα και δίνουν διαφορετικές δυνατότητες στον χρήστη. Ο κάθε χρήστης μπορεί να έχει πολλούς ρόλους, τόσο στο σύστημα όσο και στα έργα. Στην συνέχεια γίνεται ανάλυση των ρόλων αυτών και των δυνατοτήτων που δίνουν στον χρήστη.

2.1.1 Ρόλοι Συστήματος

Οι ρόλοι του συστήματος μπορεί να θεωρηθούν το ανώτερο επίπεδο ρόλων που διαθέτει το σύστημα μιας και αυτοί οι ρόλοι, που μπορούν να έχουν οι χρήστες, αποτελούν τον σημαντικότερο παράγοντα στην μορφοποίηση των σελίδων που προσφέρει η δικτυακή εφαρμογή.

Επισκέπτης

Ο επισκέπτης είναι ο βασικός ρόλος που ανατίθεται σε όσους κάνουν μόνοι τους εγγραφή στο σύστημα. Αυτοί οι χρήστες έχουν δικαίωμα να δούνε μόνο την κεντρική σελίδα και την σελίδα όπου μπορούν να αλλάξουν τα στοιχεία τους.

Χρήστης

Ο χρήστης είναι ο ρόλος που πρέπει να έχουν όλα τα μέλη της ομάδας εργασίας. Έχοντας αυτόν τον ρόλο, ο χρήστης μπορεί να δει τα έργα στα οποία έχει κάποιον ρόλο (βλέπε ρόλοι έργων).

Διαχειριστής

Αυτός είναι ο πιο σημαντικός ρόλος του συστήματος. Ο διαχειριστής είναι υπεύθυνος για τους υπόλοιπους χρήστες του συστήματος. Μπορεί να τροποποιήσει τα στοιχεία τους, να αλλάξει τους κωδικούς τους και τους ρόλους τους στο σύστημα και μπορεί να διαγράψει υπάρχοντες χρήστες όσο εύκολα μπορεί να δημιουργήσει έναν νέο. Όπως όλοι οι διαχειριστές σε όλα τα συστήματα, έτσι και σε αυτό το σύστημα αυτόν τον ρόλο πρέπει να τον έχουν έμπιστα άτομα τα οποία γνωρίζουν τις δυνατότητες του συστήματος.

2.1.2 Ρόλοι Έργων

Για να αποκτήσει κάποιος χρήστης έναν ρόλο σε ένα έργο, πρέπει πρώτα ο συντονιστής της ομάδας να δημιουργήσει το έργο και στην συνέχεια να ενεργοποιήσει μια έκδοση του εντύπου E2

του έργου στο οποίο να ορίζει ότι ο συγκεκριμένος χρήστης έχει κάποιον ρόλο στο έργο. Ο συντονιστής αρκεί να αναθέσει τον υπεύθυνο σύμβασης σε μια ενεργή έκδοση του εντύπου E2 του έργου για να μπορέσει να ξεκινήσει το έργο.

Διευθυντής (Director)

Ο Διευθυντής (Υπεύθυνος Σύμβασης) έχει την ευθύνη διαχείρισης της σύμβασης για την οποία είναι υπεύθυνος σε ‘στρατηγικό επίπεδο’ και διασφάλισης της ποιότητας των παραδοτέων. Η ευθύνη αυτή αφορά σε όλο τον ‘κύκλο ζωής’ της σύμβασης του έργου. Αυτός ο ρόλος δίνει την αποκλειστική δυνατότητα στον χρήστη να τροποποιήσει τα έντυπα E1, E2 και E3 τροποποιώντας τα στοιχεία του και δημιουργώντας νέες εκδόσεις. Ο υπεύθυνος σύμβασης πέρα από αυτό έχει και όλες τις δυνατότητες που έχουν οι χρήστες με ρόλο Υπεύθυνος Έργου στο συγκεκριμένο έργο.

Υπεύθυνος Έργου (Project Manager)

Ο Υπεύθυνος Έργου έχει την ευθύνη για την καλή εκτέλεση του έργου σε ‘λειτουργικό’ και ‘τακτικό’ επίπεδο (καθημερινή λειτουργία – day-to-day operation). Η ευθύνη αυτή περιλαμβάνει την βέλτιστη αξιοποίηση των πόρων που έχουν διατεθεί στο έργο, την κατάρτιση και τήρηση των χρονοδιαγραμμάτων του έργου, την ανάθεση εργασιών στα μέλη ομάδας εργασίας και την παραγωγή ποιοτικών παραδοτέων. Οι χρήστες που έχουν αυτόν τον ρόλο σε ένα έργο έχουν την δυνατότητα συμπλήρωσης των εντύπων E4, E4.3 και E5.

Μέλος Ομάδας Έργου (Team members)

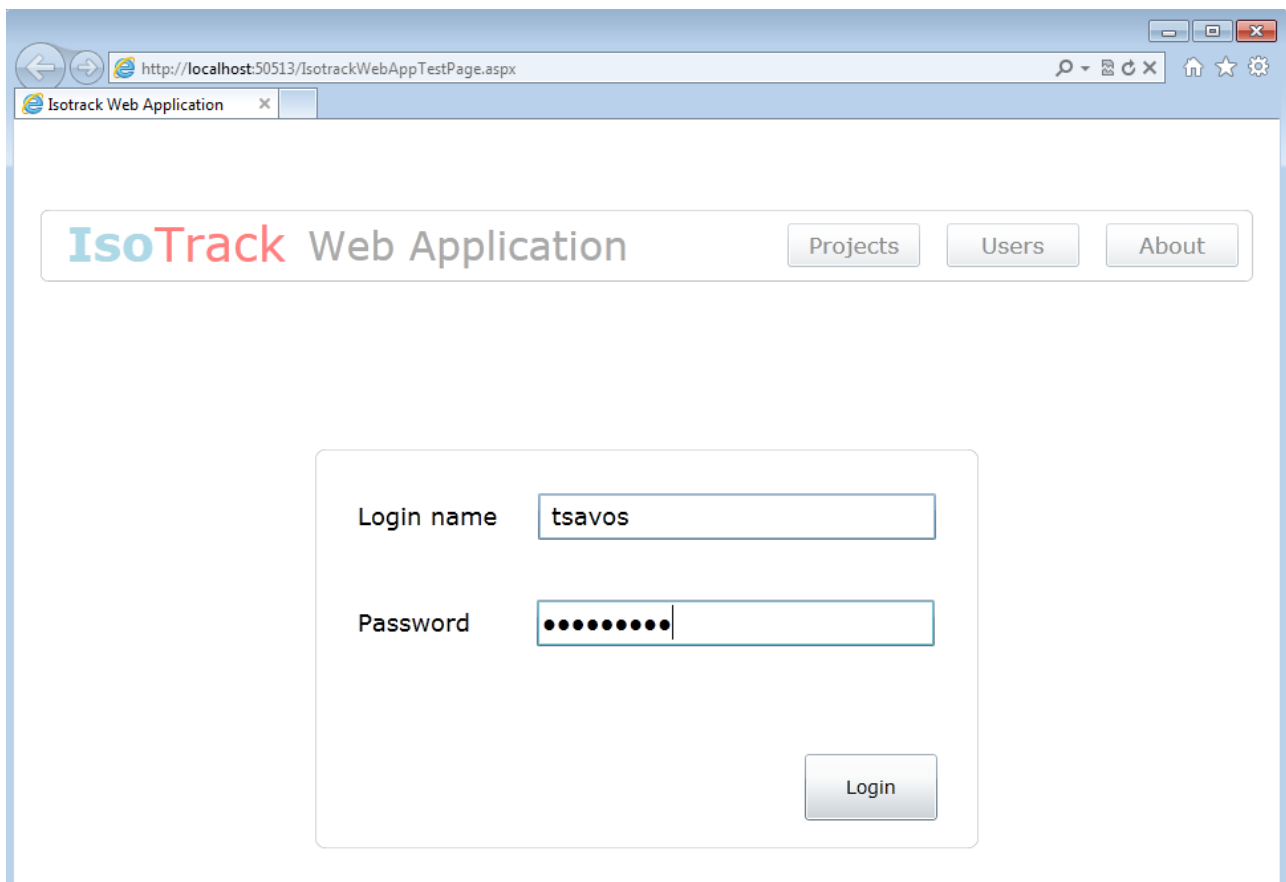
Τα μέλη της ομάδας έργου αποτελούν τους κύριους πόρους του έργου. Σκοπός τους είναι η έγκαιρη πραγματοποίηση των αντικειμένων εργασίας τα οποία τους έχουν ανατεθεί από τους υπεύθυνους έργου και σύμβασης. Όσοι έχουν αυτόν τον ρόλο στο έργο μπορούν απλά να δούνε τα έντυπα του έργου χωρίς να μπορούν να τα τροποποιήσουν ή να δημιουργήσουν νέες εκδόσεις.

ΚΕΦΑΛΑΙΟ 3: ΠΑΡΟΥΣΙΑΣΗ

Για να μπορέσει κάποιος χρήστης να λειτουργήσει την εφαρμογή που σχεδιάστηκε, πρέπει να ανοίξει το πρόγραμμα που χρησιμοποιεί για πλοήγηση στον παγκόσμιο ιστό (web browser) και να επισκεφτεί την ηλεκτρονική τοποθεσία της εφαρμογής.

3.1 Η διαδικασία του Login

Η πρώτη ενέργεια που καλείται ο χρήστης να εκτελέσει προκειμένου να χρησιμοποιήσει την εφαρμογή είναι να εισάγει το όνομα χρήστη (username) και το συνθηματικό (password) του.

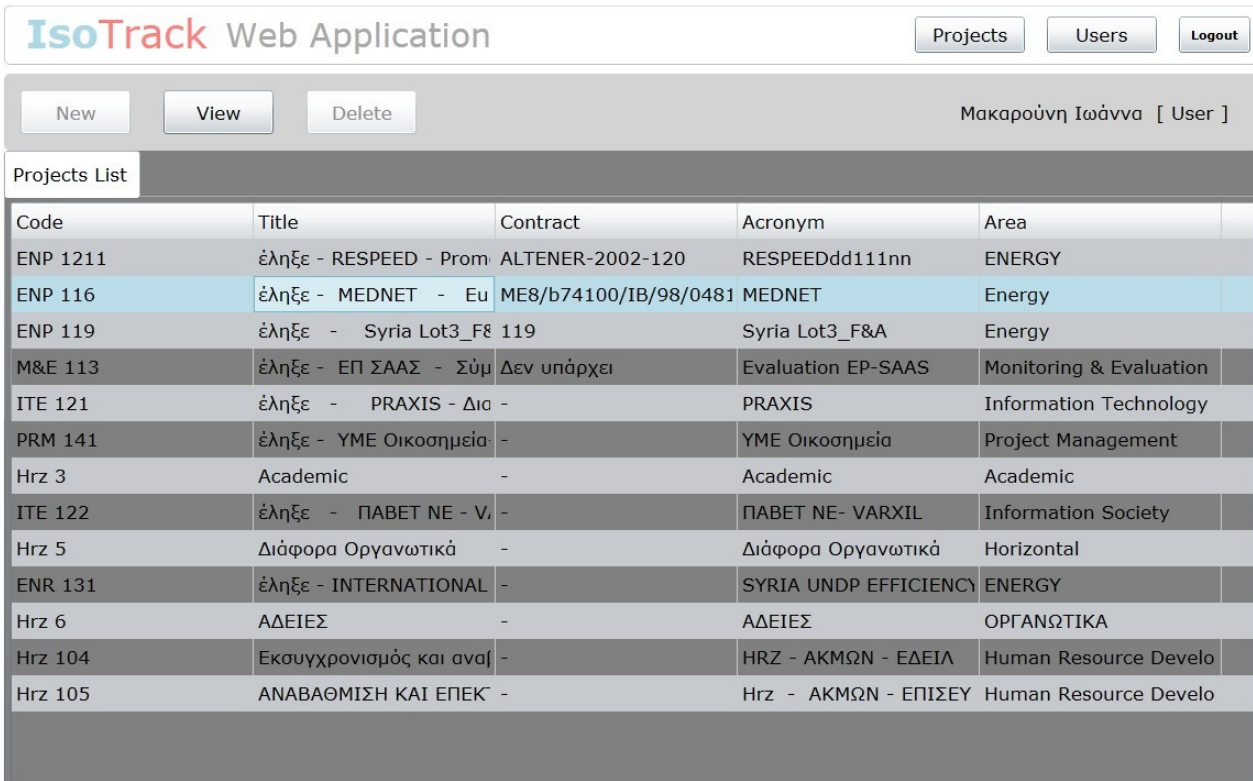


The screenshot shows a web browser window with the URL `http://localhost:50513/IsotrackWebAppTestPage.aspx`. The page title is "IsoTrack Web Application". At the top right, there are three buttons: "Projects", "Users", and "About". The main content area contains a login form with two input fields: "Login name" with the value "tsavos" and "Password" with masked characters. A "Login" button is positioned below the password field.

Με την ορθή εισαγωγή του ονόματος χρήστη και του συνθηματικού, ο χρήστης μεταφέρεται στην κεντρική σελίδα της εφαρμογής.

3.2 Η κεντρική σελίδα: η λίστα των έργων

Όλοι οι χρήστες έχουν σαν επιλογή την ‘Κεντρική Σελίδα’ η οποία είναι και η πρώτη σελίδα που επισκέπτονται οι χρήστες όταν κάνουν log in στην εφαρμογή. Σε αυτή την σελίδα, βλέπουμε την λίστα των έργων στα οποία ο τρέχον χρήστη έχει πρόσβαση. Αυτή η λίστα παρουσιάζει σύντομα κάποιες πληροφορίες σχετικά με τα έργα, πιο συγκεκριμένα: τον κωδικό (Code), τον τίτλο (Title) , το acronym και το πεδίο εφαρμογής του έργου.



The screenshot shows the IsoTrack Web Application interface. At the top, there is a header with the application name 'IsoTrack Web Application' and three buttons: 'Projects', 'Users', and 'Logout'. Below the header, there is a navigation bar with three buttons: 'New', 'View', and 'Delete', and a user profile indicator 'Μακαρούνη Ιωάννα [User]'. The main content area is titled 'Projects List' and contains a table with the following columns: Code, Title, Contract, Acronym, and Area. The table lists various projects with their respective details.

Code	Title	Contract	Acronym	Area
ENP 1211	έληξε - RESPEED - Prom	ALTENER-2002-120	RESPEEDdd111nn	ENERGY
ENP 116	έληξε - MEDNET - Eu	ME8/b74100/IB/98/0481	MEDNET	Energy
ENP 119	έληξε - Syria Lot3_F8	119	Syria Lot3_F&A	Energy
M&E 113	έληξε - ΕΠ ΣΑΑΣ - Σύμ	Δεν υπάρχει	Evaluation EP-SAAS	Monitoring & Evaluation
ITE 121	έληξε - PRAXIS - Δια		PRAXIS	Information Technology
PRM 141	έληξε - ΥΜΕ Οικοσημεία		ΥΜΕ Οικοσημεία	Project Management
Hrz 3	Academic		Academic	Academic
ITE 122	έληξε - ΠΑΒΕΤ ΝΕ - V		ΠΑΒΕΤ ΝΕ- VARXIL	Information Society
Hrz 5	Διάφορα Οργανωτικά		Διάφορα Οργανωτικά	Horizontal
ENR 131	έληξε - INTERNATIONAL		SYRIA UNDP EFFICIENCY	ENERGY
Hrz 6	ΑΔΕΙΕΣ		ΑΔΕΙΕΣ	ΟΡΓΑΝΩΤΙΚΑ
Hrz 104	Εκσυγχρονισμός και αναβ		HRZ - ΑΚΜΩΝ - ΕΔΕΙΑ	Human Resource Develo
Hrz 105	ΑΝΑΒΑΘΜΙΣΗ ΚΑΙ ΕΠΕΚ		Hrz - ΑΚΜΩΝ - ΕΠΙΣΕΥ	Human Resource Develo

Παρατηρούμε μια σειρά από ενέργειες στην διάθεση του χρήστη. Αυτές οι επιλογές είναι ενεργοποιημένα ή μη ανάλογα με τον ρόλο που έχει ο χρήστης μέσα στο σύστημα.

- **Projects:** Αυτή η επιλογή επαναφέρει την κεντρική σελίδα (η παραπάνω εικόνα) όπου ο χρήστης βλέπει την λίστα των έργων που είναι στο σύστημα και στα όποια έχει πρόσβαση. Εάν ο χρήστης είναι διαχειριστής (Administrator) τότε μπορεί να έχει πρόσβαση σε όλα τα έργα της εφαρμογής. Παρατηρούμε επίσης τις λειτουργικές επιλογές:
 - **New (Δημιουργία) :** Αυτή η ενέργεια επιτρέπει την δημιουργία νέου έργου. Άρα είναι ενεργοποιημένη μόνο εφόσον ο τρέχον χρήστης έχει ρόλο “Project Creator”.

- **View** (Προβολή): Αυτή η ενέργεια επιτρέπει στο τρέχον χρήστη να ανοίξει την “οθόνη” του έργου όπου μπορεί να βλέπει λεπτομερώς τα στοιχεία ενός έργου και να τα μεταβάλλει εφόσον διαθέτει τα δικαιώματα.
- **Delete** (Διαγραφή): Η ενέργεια αυτή επιτρέπει την οριστική διαγραφή ενός έργου από το σύστημα μαζί με οτιδήποτε πληροφορία που συνδέεται με αυτό το έργο. Είναι διαθέσιμη εφόσον ο τρέχον χρήστης έχει ρόλο Project Director.
- **Users** (Χρήστες): Αυτή η επιλογή φέρνει την λίστα των χρηστών του συστήματος. Ανάλογα με τον ρόλο του χρήστη μπορεί να του δοθεί δυνατότητα δημιουργίας, μεταβολής ή διαγραφής ενός χρήστη (περισσότερα πιο κάτω)
- **Logout**: Από αυτό το κουμπί ο τρέχον χρήστης αποσυνδέεται από την εφαρμογή και γυρίζει στην αρχική οθόνη **login**

Παρατηρείστε ότι κάτω από αυτό το μενού βλέπουμε το όνομα του χρήστη και το ρόλο του στο σύστημα. Αξίζει να σημειωθεί εδώ ότι ανάλογα με το ρόλο συστήματος του, ο χρήστης μπορεί να βλέπει όλα τα διαθέσιμα έργα στο σύστημα (εάν ο χρήστης είναι Administrator ή Project Director) ή, στην άλλη περίπτωση μόνο τα έργα στα οποία συμμετέχει.

Με την ενέργεια “View”, ο χρήστης μπορεί να έχει πρόσβαση στην οθόνη του τρέχοντος επιλεγμένου έργου.

3.3 Η οθόνη έργου

Η οθόνη έργου παρουσιάζει μια σειρά από καρτέλες που ενεργοποιούνται ανάλογα με τον ρόλο του χρήστη μέσα στο έργο. Παρατηρείστε ότι πάνω από την οθόνη έργου βρίσκεται ο κωδικός του συγκεκριμένου έργου μαζί με ένα κουμπί δίπλα με το σύμβολο “X” που επιτρέπει το κλείσιμο της οθόνης. Για να επιστρέψει κανείς στην λίστα των έργων, μπορεί να πατήσει το κουμπί “Projects List”.

Καρτέλα E1:

The screenshot displays the IsoTrack Web Application interface. At the top, there is a navigation bar with 'Projects', 'Users', and 'Logout' buttons. Below this, a user profile bar shows 'Μακαρούνη Ιωάννα [User]'. The main content area is titled 'Projects List' and shows a search bar with 'ENP 116' and a close button. Below the search bar, there are tabs for 'E1', 'E2', 'E3', 'E4', and 'E5', with 'E1' selected. The 'Project Manager' section is active, showing tabs for 'Main Information', 'Project Durations', 'Project Recipients', and 'Participants'. The 'Main Information' tab is selected, displaying the following fields:

- Project code:** ENP 116
- Project acronym:** MEDNET
- Project title:** έληξε - MEDNET - Euro-Mediterranean Energy Policy Training Network
- Project description:** Το Σχέδιο Δράσης του Ευρω-Μεσογειακού Φόρουμ Ενέργειας που υιοθετήθηκε σε Υπουργικό επίπεδο στις 11 Μαΐου 1998, παρουσιάζει την ανάγκη στενότερης συνεργασίας μεταξύ των Μεσογειακών χωρών στον τομέα της ενεργειακής πολιτικής. Το έργο αυτό έχει ως σκοπό την δημιουργία ενός μηχανισμού με την ισοψηφία ενός Ευρω-Μεσογειακού Λικτύου Εκπαίδευσης για την Ενθεσμιακή Πολιτική, που θα υποστηρίζει
- Project assignor:** Ευρωπαϊκή Επιτροπή, MEDA PROGRAMME
- Project area:** Energy

At the bottom right of the form, there are 'Post' and 'Cancel' buttons.

Ο διευθυντής έργου (ή υπεύθυνος σύμβασης) έχει πρόσβαση και μπορεί να μεταβάλει την καρτέλα E1. Εισάγει τα κύρια στοιχεία ενός έργου: ο κωδικός, το ακρωνύμιο, η επιστημονική περιοχή, ο αριθμός συμβολαίου και ο τίτλος, και στην υπό-καρτέλα “Project Durations”, πληροφορίες όπως η διάρκεια του έργου, η πραγματική και η αναθεωρημένη, οι ημερομηνίες έναρξης και λήξης). Παρατηρείστε στη πάνω γωνία δεξιά της καρτέλας, βλέπουμε το ρόλο του τρέχοντος χρήστη στο συγκεκριμένο έργο. Στην συνέχεια ο διευθυντής έργου μπορεί να προσθέσει συμμετέχοντες (και να προσδιορίσει τους ρόλους τους)και αποδέκτες του συγκεκριμένου έργου.

Projects List ENP 116 [Project Manager]

E1 E2 E3 E4 E5

Main Information **Project Durations** **Project Recipients** Participants

Participant name	Participant role
ΕΠΙΣΕΥ - ΕΡΓΑΣΤΗΡΙΟ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣ	Is Leader
ICTAF	Is Partner
ALMEE	Is Partner
Bogazici University	Is Partner
CDER	Is Partner
MEMR	Is Partner
Ministry of Electricity (MOE)	Is Partner

Add Delete

Post Cancel

Projects List ENP 116 [Project Manager]

E1 E2 E3 E4 E5

Main Information **Project Durations** **Project Recipients** Participants

Recipient
EC / AIDCO
Mediterranean Partner Countries
ΥΠΟΥΡΓΕΙΟ ΜΕΤΑΦΟΡΩΝ - Δ.Α. ΣΑΑΣ
ΥΠΟΥΡΓΕΙΟ ΑΝΑΠΤΥΞΗΣ-ΤΟΜΕΑΣ ΕΝΕΡΓΕΙΑΣ
EC Delegations
ΥΠΟΥΡΓΕΙΟ ΕΡΓΑΣΙΑΣ

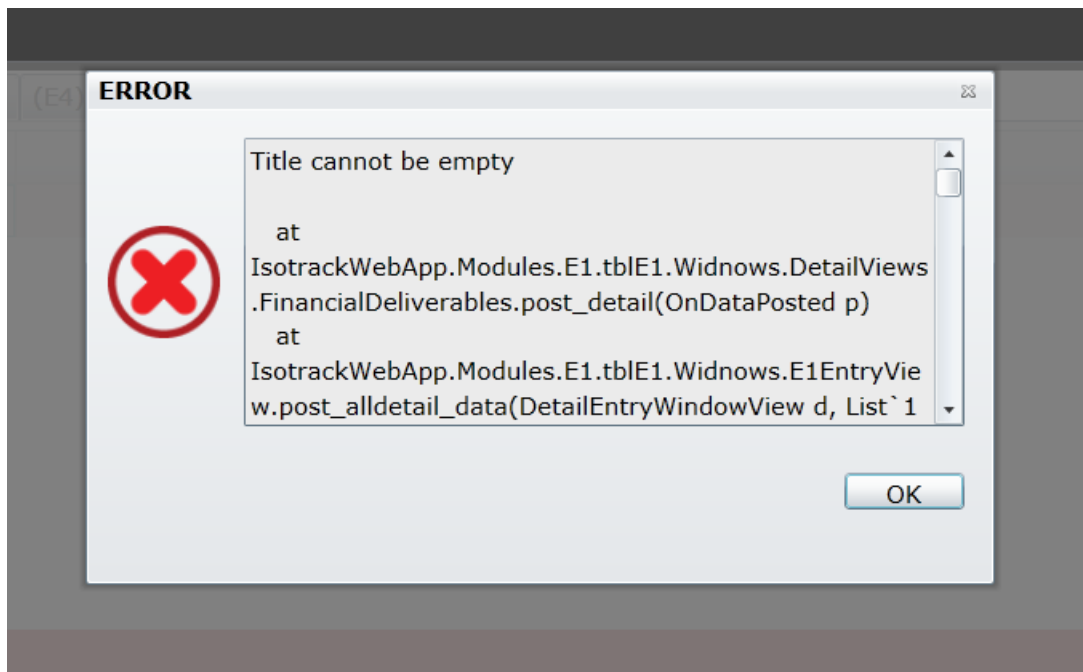
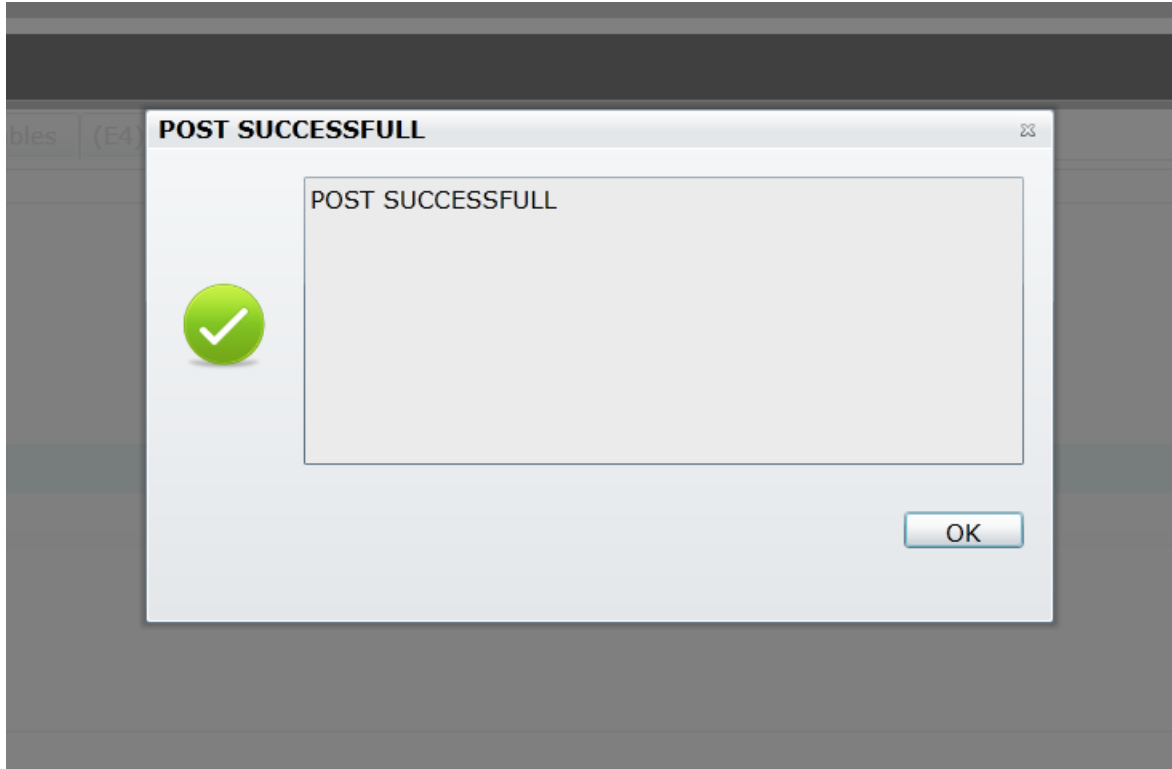
Add Delete

Post Cancel

Πατώντας το κουμπί “Σώσιμο” (Post) στο κάτω δεξί μέρος της οθόνης, μπορεί να αποθηκεύσει τις αλλαγές ή να τις ακυρώσει με το κουμπί “Ακύρωση” (Cancel).



Επιτυχής ή αποτυχημένη αποθήκευση αποδεικνύεται με αντίστοιχο μήνυμα



Στην συνέχεια ο υπεύθυνος σύμβασης καλείται να συμπληρώσει τις ημερομηνίες έναρξης και λήξης του έργου στην μορφή ‘μέρα/μήνας/χρόνος’. Για την ευκολία των χρηστών, έχει προστεθεί ένα εικονίδιο ενός ημερολογίου το οποίο με το που πατηθεί ανοίγει ένα νέο παράθυρο από το οποίο μπορεί να επιλέξει ο χρήστης την ζητούμενη ημερομηνία μέσω ενός πολύ εύχρηστου ηλεκτρονικού ημερολογίου.

Καρτέλα E2:

The screenshot displays the IsoTrack Web Application interface. At the top, the application name 'IsoTrack Web Application' is visible, along with navigation buttons for 'Projects', 'Users', and 'Logout'. Below this, there are buttons for 'New', 'View', and 'Delete', and the user name 'Μακαρούνη Ιωάννα [User]'. The main content area is titled 'Projects List' and shows a search bar with 'ENP 116'. Below the search bar, there are tabs for 'E1', 'E2', 'E3', 'E4', and 'E5', with 'E2' selected. The 'E2' tab is further divided into 'E2 Information' and 'E2 Resources', with 'E2 Resources' active. The 'E2 Resources' section contains several input fields: 'Project field' (MPs, EU), 'Equipment' (Laptops, presentation equipment, conference material), 'Infrastructure' (empty), 'Primary budget' (1.690.150,00), 'Secondary budget' (1.203.000,00), 'Primary team budget' (1.316.102,00), and 'Secondary team budget' (empty). At the bottom right, there are 'Post' and 'Cancel' buttons.

Δικαίωμα να τροποποιεί αυτό το έντυπο έχει μόνο ο διευθυντής έργου (υπεύθυνος σύμβασης). Από όλα τα πεδία τα οποία υπάρχουν σε αυτή την φόρμα σημαντικό είναι να αναφέρουμε ότι τα δύο πεδία του συνολικού προϋπολογισμού συμπληρώνονται αυτόματα από το άθροισμα των προϋπολογισμών της αντίστοιχης στήλης και αυτό γίνεται μόνο όταν αποθηκευτούν οι αλλαγές (πατώντας το κουμπί Υποβολή). Στην υπό-καρτέλα “E2 Resources”, ο υπεύθυνος έργου μπορεί να διαχειρίζεται την λίστα των πόρων (χρήστες) του έργου.

Projects List ENP 116

[Project Director]

E1 E2 E3 E4 E5

E2 Information E2 Resources

User name	Team role
Ψαρράς Γιάννης	Project Director
Μακαρούνη Ιωάννα	Project Manager
Σταματούκος Σταύρος	Team member
Ασκούνης Δημήτρης	Team member
Σαμουηλίδης Ι.-Ε.	Team member
Τσαλμάνης Άρης	Team member
Γαλιγάλης Σάκης	Team member
Papastamatiu Panos	Team member
Derakis Kostas	Team member

Add Delete

User name Ψαρράς Γιάννης

Team role Project Director

Project type Internal

User man days 144,00

Post Cancel

Από αυτή την οθόνη, μπορεί να προστεθεί ένας χρήστης και να προσδιοριστεί ο ρόλος του στο έργο.

Καρτέλες: E3 Financial deliverables, E3 Technical deliverables

Projects List ENP 116 ENP 1211 ENP 125 ✕

E1 E2 E3 E4 E5 [Project Director]

Financial Deliverables Technical Deliverables

Title	Scheduled date	Reported date
Final financial deliverable	7/10/2005	
Interim financial deliverable	9/17/2004	

Add Delete

Post ✓ Cancel ✕

Projects List ENP 116 ENP 1211 ENP 125 ✕

E1 E2 E3 E4 E5 [Project Director]

Financial Deliverables Technical Deliverables

Title	Scheduled date	Reported date
1st Meeting	10/20/2003	10/20/2003
1st Workshop Lebanon	12/3/2004	
2nd Meeting	2/16/2004	2/17/2004
2nd Workshop Turkey	2/17/2005	
3rd Workshop Morocco	4/7/2005	
3rd Workshop Syria		
Final Report	7/10/2005	
Interim Report	9/17/2004	

Add Delete

Post ✓ Cancel ✕

Αυτά τα έντυπα μπορεί να τα συμπληρώσει μόνο ο υπεύθυνος σύμβασης. Εδώ καθορίζονται τα τεχνικά και τα οικονομικά παραδοτέα του έργου καθώς και οι ημερομηνίες παράδοσης τους. Για να προστεθούν παραδοτέα, ο χρήστης πατάει στο κουμπί πρόσθεση (Add) και συμπληρώνει το όνομα του παραδοτέου. Για την συμπλήρωση των ημερομηνιών, με το που πατάει ο συντάκτης στο πεδίο όπου

πρέπει να την συμπληρώσει, αμέσως εμφανίζεται σε νέο παράθυρο το ηλεκτρονικό ημερολόγιο που προαναφέρθηκε. Για να διαγράψει ένα παραδοτέο, ο χρήστης πρέπει να πατήσει το κουμπί Διαγραφή (Delete).

Καρτέλα (E4) Project Tasks and Resources:

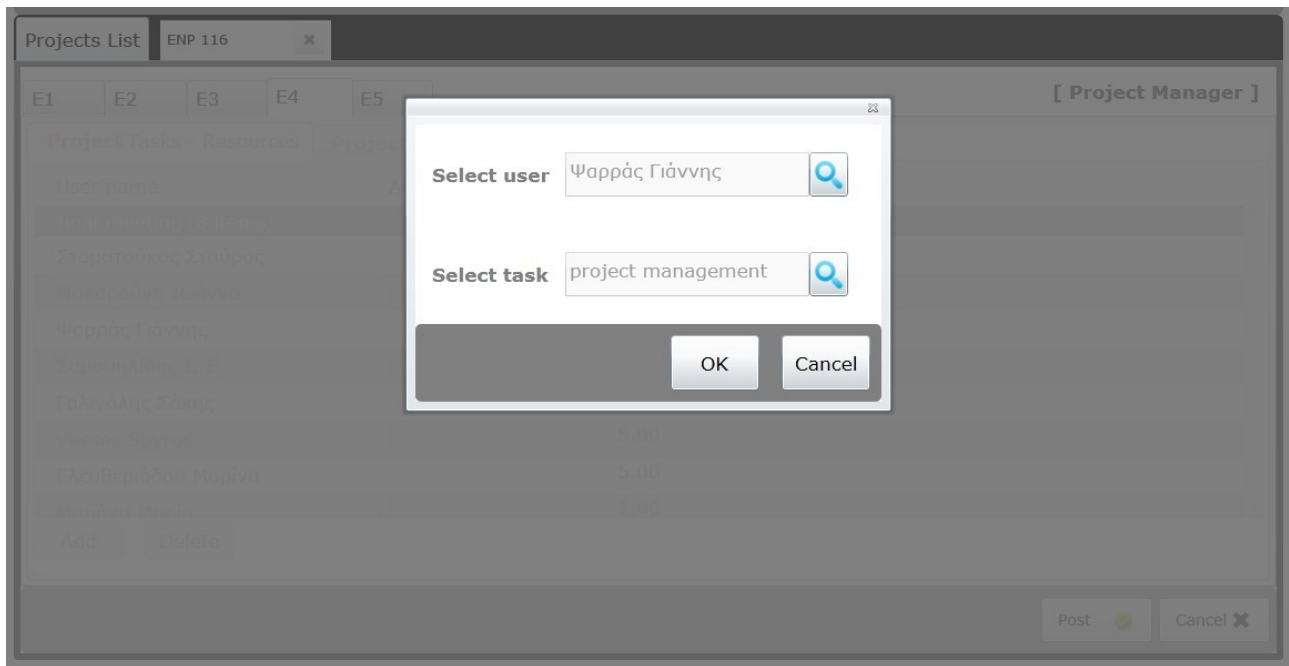
The screenshot displays a web application interface for managing project tasks and resources. At the top, there are tabs for 'Projects List', 'ENP 116', 'ENP 1211', and 'ENP 125'. Below this, there are tabs for 'E1', 'E2', 'E3', 'E4', and 'E5', with 'E4' selected. The main content area is titled 'Project Tasks - Resources' and contains a table with the following data:

User name	Activity	
Handbook Methodology Guide (5 items)		
1st Meeting Athens - Oct 2003 (5 items)		
Ψαρράς Γιάννης		3.00
Φλάμος Αλέξανδρος		5.00
Δούκας Χάρης		0.00
Αναγνωστόπουλος Κώστας		5.00
Παλιτζιάνας Κώστας		9.00
2nd Meeting Athens -February 2004- (5 items)		
Ψαρράς Γιάννης		4.00

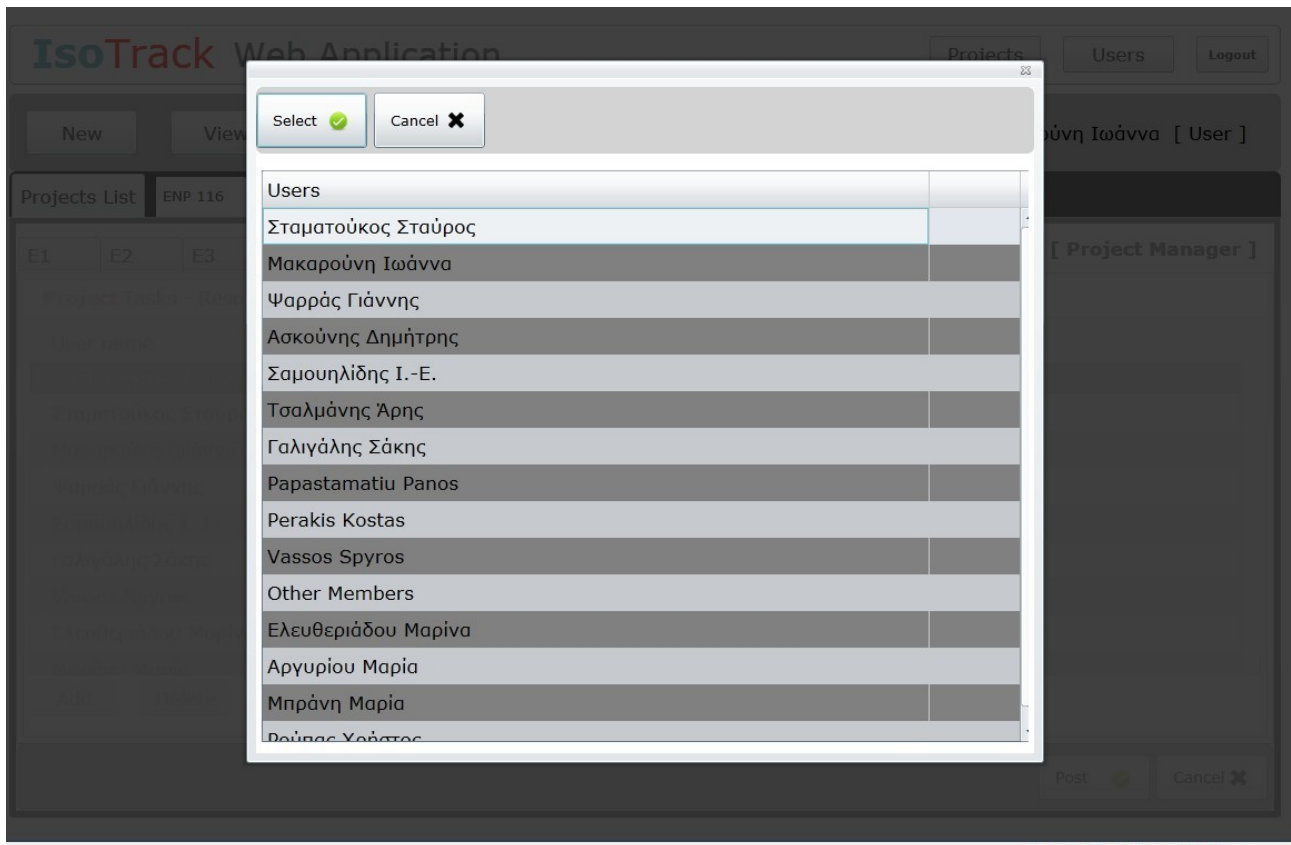
Below the table, there are 'Add' and 'Delete' buttons. At the bottom right of the interface, there are 'Post' and 'Cancel' buttons.

Δικαίωμα συμπλήρωσης και μεταβολής αυτής της καρτέλας έχει ο υπεύθυνος έργου (Project manager). Από αυτή την οθόνη μπορεί να επιλέξει χρήστες που θα αποτελέσουν στην συνέχεια τους ανθρώπινους πόρους του έργου. Παρατηρείστε ότι οι χρήστες ομαδοποιούνται ανά υπό-εργασία.

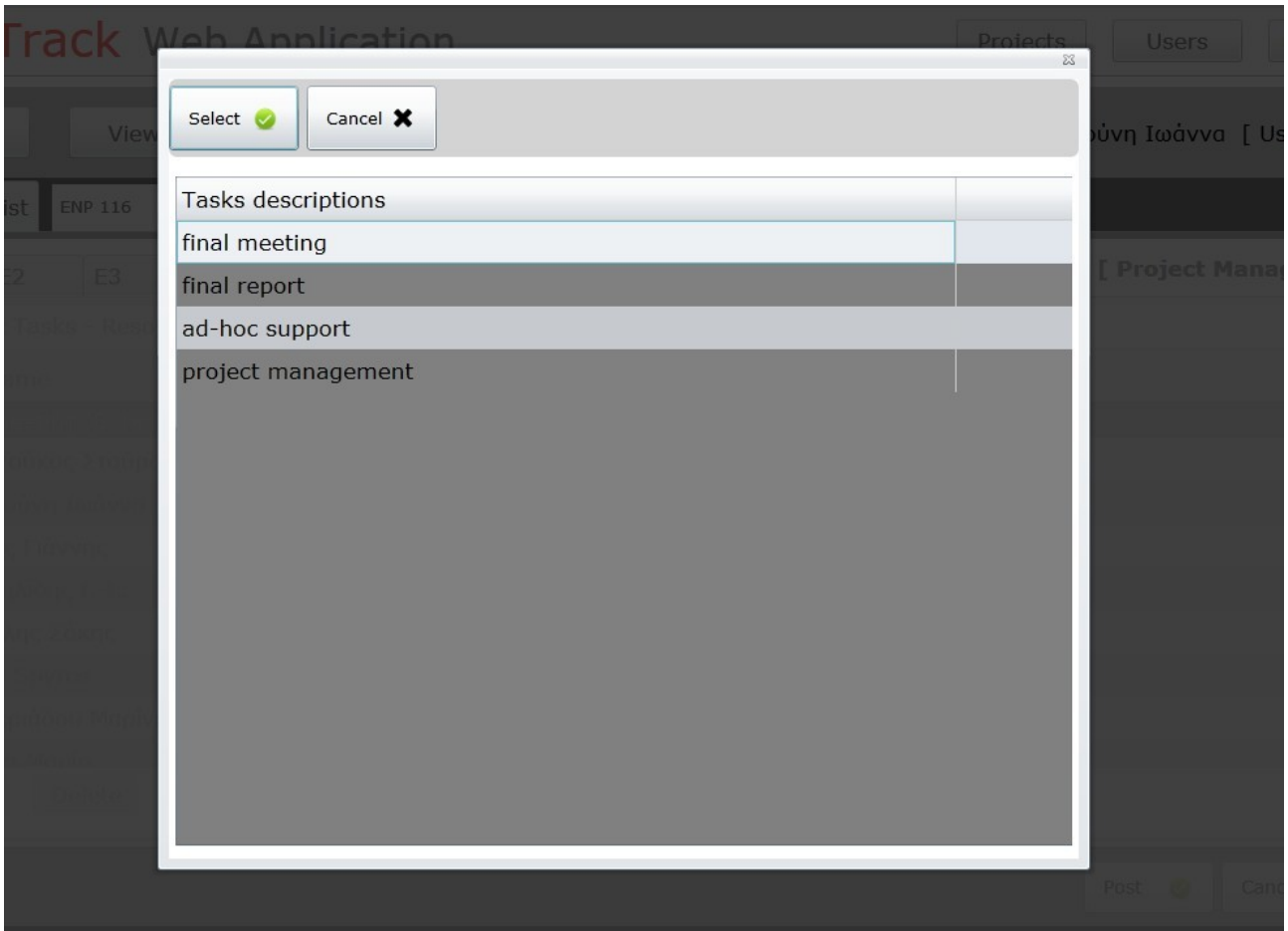
Με το πάτημα “Add” (προσθήκη), εμφανίζεται διάλογος όπου πρώτα επιλέγει τον χρήστη που θα αποτελέσει στην συνέχεια πόρο στο έργο. Επιτρεπτοί χρήστες είναι μόνο αυτοί που έχουν οριστεί προηγουμένως στην καρτέλα E2 Resources. Στην συνέχεια επιλέγει σε πια υπό-εργασία θα εργάζεται.



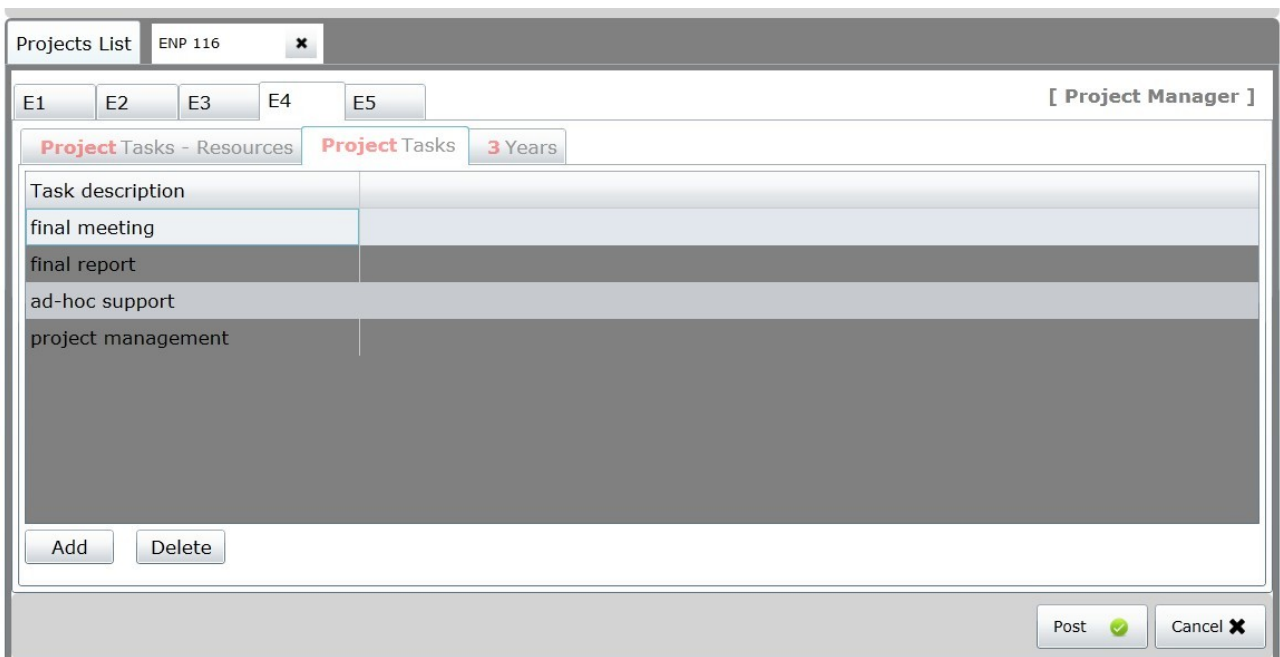
Επιλογή επιτρεπτού χρήστη



Επιλογή επιτρεπτής υπό-εργασίας



Επιτρεπτές εργασίες ορίζονται στην επόμενη υπό-καρτέλα “Project Tasks”.



Αυτό το έντυπο μπορεί να το συμπληρώσει ο υπεύθυνος έργου (Project manager). Εδώ ο υπεύθυνος έργου κατασκευάζει την λίστα των εργασιών που αποτελούν το έργο. Με το πάτημα του κουμπιού Add (πρόσθεση), εμφανίζεται ένας διάλογος όπου ο υπεύθυνος έργου μπορεί να προσδιορίζει το όνομα της εργασίας που στην συνέχεια, με την αποδοχή, θα προστεθεί στο σύνολο των εργασιών του έργου.

Αφού επιλέξει ένα χρήστη και μια υπό-εργασία, ο υπεύθυνος έργου πρέπει να πατήσει την υποδοχή (“Select”) για να επιστρέψει στην προηγούμενη καρτέλα.

Καρτέλα 3 Years:

Projects List ENP 116 ✖

E1 E2 E3 E4 E5 [Project Manager]

Project Tasks - Resources Project Tasks 3 Years

User name	Month Man Days 1	Month Man Days 2	Month Man Days 3	Month Man Days 4	Month Ma
Αργυρίου Μαρία	0.00	0.00	0.00	0.00	

Add Delete

Post ✓ Cancel ✖

Δικαίωμα συμπλήρωσης έχει ο υπεύθυνος έργου. Ο χρήστης καλείται να δημιουργήσει μια νέα γραμμή η οποία αντιστοιχεί σε κάποια χρονιά την οποία και συμπληρώνει. Ο χρήστης μπορεί να επιλέξει το μέλος ομάδας εργασίας από την λίστα όλων των επιτρεπτών χρηστών του έργου. Οι ανθρωποημέρες ανά μήνα μπορούν να είναι πραγματικοί αριθμοί με τα δεκαδικά ψηφία να χωρίζονται από τα ακέραια με το κόμμα. Για να διαγραφεί κάποια γραμμή (είτε χρόνο ή μέλος ομάδας εργασίας), ο χρήστης πρέπει να την επιλέξει και να πατήσει το κουμπί delete.

Καρτέλα E5:



Projects List ENP 116

E1 E2 E3 E4 E5 [Project Manager]

E5			
Week No	Start Date	End Date	
1	11/14/2011	11/20/2011	
2	11/21/2011	11/27/2011	
3	11/28/2011	12/4/2011	

E5 Deliverables		
User name	Review man days	Review week percents (%)
ad-hoc support (1 item)		
Πτωχός Δημήτρης	0.00	0.00
project management (1 item)		
Σαμουηλίδης Ι.-Ε.	0.00	

Add Delete Add Delete

Post  Cancel 

Σε αυτή την οθόνη, πρώτα, ο υπεύθυνος έργου προσθέτει μια εγγραφή που αντιστοιχεί σε μια εβδομάδα (η επόμενη). Στην συνέχεια ένας χρήστης μπορεί να προσθέσει ή να μεταβάλει ένα παραδοτέο (E5 Deliverable) που αναφέρεται στην επιλεγμένη εβδομάδα. Εδώ εφόσον ο χρήστης έχει ρόλο μέλος ομάδας (Team member) , μπορεί να βλέπει και να τροποποιεί μόνο εγγραφές που τον αφορούν.

Η δενδρική μορφή με την οποία παρουσιάζονται τα δεδομένα είναι αντίστοιχη με αυτή που έχει το αρχικό έντυπο μιας και υπάρχουν οι γραμμές που συμπληρώνονται τα μέλη της ομάδας εργασίας και από κάτω μπαίνουν τα αντικείμενα εργασίας με τα οποία ασχολήθηκαν την προηγούμενη εβδομάδα (που συμπληρώνεται στις απολογιστικές στήλες) και αυτά με τα οποία θα ασχοληθούν την επόμενη εβδομάδα (όπως συμπληρώνεται στις στήλες πρόβλεψης).

Η δημιουργία Έργου

Χρήστης με ρόλο συστήματος Project Creator μπορεί να δημιουργήσει και να

αποθηκεύσει νέο έργο στο σύστημα. Με το πάτημα του κουμπιού “New”, ο χρήστης μεταφέρεται σε διάλογο όπου καλείται να συμπληρώσει τα βασικά χαρακτηριστικά του έργου. (τον κωδικό, το ακρωνύμιο, την επιστημονική περιοχή, τον αριθμό συμβολαίου και τον τίτλο του) Αν ο χρήστης προσπαθήσει να δημιουργήσει το έργο χωρίς να έχει συμπληρώσει τουλάχιστον τον κωδικό ή τον ακρώνυμο, τότε του εμφανίζεται μήνυμα που τον παροτρύνει να τα συμπληρώσει. Σημειώνεται ότι για να μπορέσουν να δουν το έργο μετά την δημιουργία του, το σύστημα προσθέτει αυτόματα μια έγγραφη στο E2 Resources όπου ο δημιουργός του έργου είναι και ο υπεύθυνος σύμβασης. Αργότερα μπορεί να μεταβάλει αυτή την πληροφορία και να προσθέσει άλλα μέλη στο έργο.

The screenshot displays the 'IsoTrack Web Application' interface. A modal window titled 'New Project' is open, showing a form for creating a new project. The form has two tabs: 'Main Information' (selected) and 'Project Durations'. The 'Main Information' tab contains the following fields:

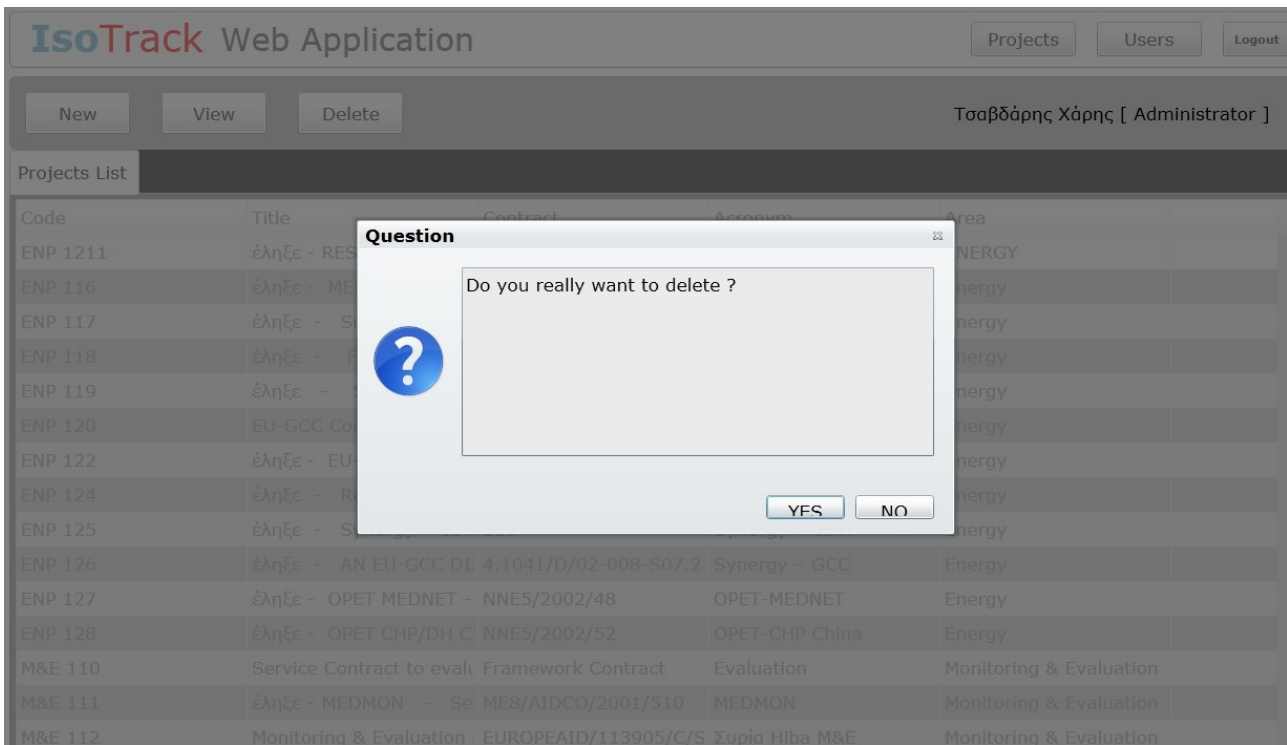
- Project code:** NEW_PROJECT
- Project acronym:** N_PRJ
- Project title:** (empty text box)
- Project description:** (empty text area)
- Project assignor:** (empty text box)
- Project area:** (empty text box)
- Project contract:** (empty text box)

At the bottom of the modal, there are two buttons: 'Post' (with a green checkmark icon) and 'Cancel' (with a red X icon). The background shows a list of projects with codes like ENP 1211, ENP 116, ENP 117, ENP 118, ENP 119, ENP 120, ENP 122, ENP 124, ENP 125, ENP 126, ENP 127, ENP 128, M&E 110, and M&E 111.

Με το πάτημα “Post” γίνεται η αποδοχή και το νέο έργο αποθηκεύεται στο σύστημα. Αντίθετα με το πάτημα “Cancel” γίνεται ακύρωση της ενέργειας και επιστρέφει ο χρήστης στην προηγούμενη οθόνη.

Η διαγραφή έργου:

Δικαίωμα διαγραφής έργου έχει χρήστης με ρόλο συστήματος “Project Creator”. Με το πάτημα “Delete”, εμφανίζεται σχετικό διάλογο που επιτρέπει στο χρήστη να επιβεβαιώσει την ενέργεια αυτή που έχει οριστικές συνέπειες. Πράγματι με την αποδοχή, θα διαγραφούν από το σύστημα, θα διαγραφεί ότι πληροφορία συνδέεται με το επιλεγμένο έργο.



The screenshot displays the IsoTrack Web Application interface. At the top, there is a navigation bar with 'Projects', 'Users', and 'Logout' buttons. Below this, a secondary bar contains 'New', 'View', and 'Delete' buttons, along with the user information 'Τσαβδάρης Χάρης [Administrator]'. The main content area is titled 'Projects List' and contains a table with columns for Code, Title, Contract, Acronym, and Area. A modal dialog box titled 'Question' is overlaid on the table, asking 'Do you really want to delete ?' with 'YES' and 'NO' buttons.

Code	Title	Contract	Acronym	Area
ENP 1211	έληξε - RES			ENERGY
ENP 116	έληξε - ME			Energy
ENP 117	έληξε - S			Energy
ENP 118	έληξε - T			Energy
ENP 119	έληξε -			Energy
ENP 120	EU-GCC Co			Energy
ENP 122	έληξε - EU			Energy
ENP 124	έληξε - R			Energy
ENP 125	έληξε - S			Energy
ENP 126	έληξε - AN EU-GCC DI	4.1041/D/02-008-S07.2	Synergy - GCC	Energy
ENP 127	έληξε - OPET MEDNET -	NNE5/2002/48	OPET-MEDNET	Energy
ENP 128	έληξε - OPET CHP/DH C	NNE5/2002/52	OPET-CHP China	Energy
M&E 110	Service Contract to evalu	Framework Contract	Evaluation	Monitoring & Evaluation
M&E 111	έληξε - MEDMON - Se	ME8/AIDCO/2001/510	MEDMON	Monitoring & Evaluation
M&E 112	Monitoring & Evaluation	EUROPEAID/113905/C/S	Συρία Hiba M&E	Monitoring & Evaluation

3.4 Η οθόνη χρήστη

Από το κεντρικό μενού (Projects, Users, Logout), με το πάτημα του κουμπιού “Users”, μπορεί κανείς να έχει πρόσβαση στην λίστα όλων των χρηστών στο σύστημα.

The screenshot shows the 'IsoTrack Web Application' interface. At the top right, there are buttons for 'Projects', 'Users', and 'Logout'. Below these, there are buttons for 'New', 'View', and 'Delete'. The user 'Tsabdaris Charis' is identified as the Administrator. The main content is a table titled 'Users List' with the following data:

Name	Mail	Address	Is active	Telephone
Πτωχός Δημήτρης	dptoh@epu.ntua.gr	-	<input type="checkbox"/>	- 01010101010101010
Καραμολέγκος Γιώργος	gkaramo@epu.ntua.gr	-	<input type="checkbox"/>	-
Σταματούκος Σταύρος	sstam@epu.ntua.gr	sstam@epu.ntua.gr	<input type="checkbox"/>	
Μακαρούνη Ιωάννα	imakar@epu.ntua.gr	imakar@epu.ntua.gr	<input type="checkbox"/>	-
Ψαρράς Γιάννης	john@epu.ntua.gr		<input type="checkbox"/>	
Ασκούνης Δημήτρης	askous@epu.ntua.gr		<input type="checkbox"/>	444444444444
Φλάμος Αλέξανδρος	aflamos@epu.ntua.gr		<input type="checkbox"/>	
Καγιαννάς Αργύρης	akagian@epu.ntua.gr	akagian@epu.ntua.gr	<input type="checkbox"/>	
Σαμουηλίδης Ι.-Ε.	jes@epu.ntua.gr	jes@epu.ntua.gr	<input type="checkbox"/>	
Λεβέντης Πάνος	panos@epu.ntua.gr	panos@epu.ntua.gr	<input type="checkbox"/>	
Ζευώλης Δημήτρης	jimzev@epu.ntua.gr	jimzev@epu.ntua.gr	<input type="checkbox"/>	
Παντελιάς Σαράντης	sarpant@otenet.gr		<input type="checkbox"/>	
Παπακωνσταντίνου Θανά	tpapas@epu.ntua.gr		<input type="checkbox"/>	
Κωστάρας Γιώργος	gkost@epu.ntua.gr		<input type="checkbox"/>	
Ματσόπουλος Γιώργος	gmatso@epu.ntua.gr		<input type="checkbox"/>	

Σε αντιστοιχία με την λίστα των έργων, βλέπουμε ότι κι εδώ ο χρήστης έχει στην διάθεση του 3 βασικές ενέργειες, ενεργοποιημένες ανάλογα με τον ρόλο του χρήστη στο σύστημα: New, View, Delete. Προφανώς δικαίωμα δημιουργίας ή διαγραφής χρήστη έχει χρήστης με ρόλο Administrator. Το κουμπί “View” επιτρέπει να βλέπει κανείς τα βασικά στοιχεία του επιλεγμένου χρήστη (Το όνομα χρήστη, το κωδικό σύνδεσης, την ηλεκτρονική διεύθυνση, το τηλέφωνο...κτλ). Αξίζει να σημειώνεται ότι εάν ανοίξει κανείς καρτέλα άλλου χρήστη, τότε δεν μπορεί να μεταβάλει κανένα στοιχείο(όπως στην παρακάτω εικόνα όπου ο τρέχον χρήστης είναι ο Τσαβδάρης ενώ η καρτέλα χρήστη ανήκει στον Λεβέντη Πάνο. Παρατηρούμε ότι τα πεδία στην οθόνη είναι όλα απενεργοποιημένα).

New View Delete Τσαβδάρης Χάρης [Administrator]

Users List Πτωχός Δημήτρης Σταματούκος Σταύρος Λεβέντης Πάνος ✕

General Information **User Roles**

User name Λεβέντης Πάνος

Login panos

Email panos@epu.ntua.gr

Telephone

Fax

Address panos@epu.ntua.gr

Postal code 0

Active

Post Cancel ✕

Εκτός από τις βασικές πληροφορίες, υπάρχει η υπό-καρτέλα “User roles” όπου παρουσιάζεται την λίστα των ρόλων του επιλεγμένου χρήστη. Δικαίωμα μεταβολής αυτής της πληροφορίας έχει μόνο ο Administrator.

Users List Πτωχός Δημήτρης Σταματούκος Σταύρος Λεβέντης Πάνος Τσαβδάρης Χάρης ✕

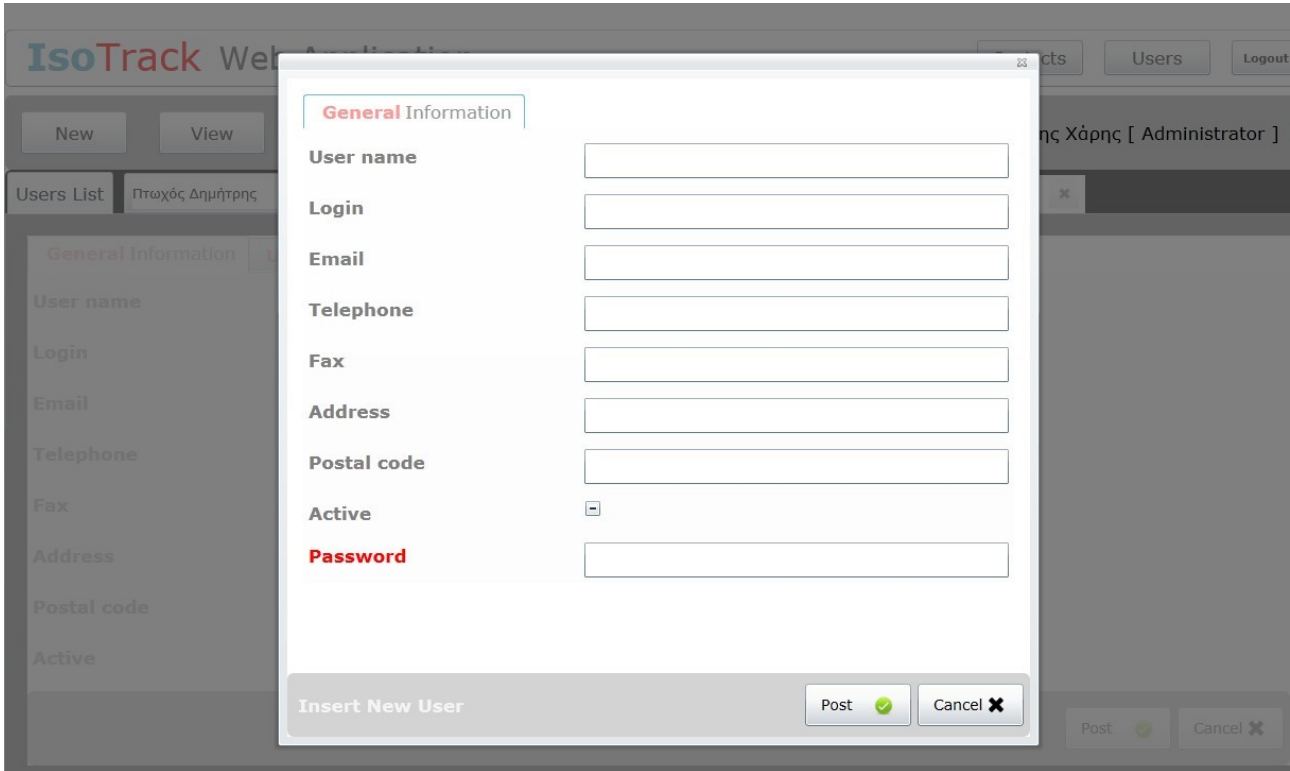
General Information **User Roles**

Application role	
Administrator	
User	

Post Cancel ✕

Δημιουργία νέου χρήστη

Εφόσον ο τρέχον χρήστης έχει ρόλο Administrator στο σύστημα, μπορεί να προσθέσει νέο χρήστη στο σύστημα. Με το πάτημα του κουμπιού “New” μεταφέρεται σε διάλογο όπου καλείται να συμπληρώσει βασικά στοιχεία του χρήστη.



The screenshot displays the 'IsoTrack Web Application' interface. A modal dialog box titled 'General Information' is open, allowing the creation of a new user. The dialog contains the following fields:

- User name
- Login
- Email
- Telephone
- Fax
- Address
- Postal code
- Active (checkbox)
- Password (highlighted in red)

At the bottom of the dialog, there are two buttons: 'Post' (with a green checkmark icon) and 'Cancel' (with a red X icon). The background shows the application's main menu with options like 'New', 'View', 'Users List', and 'Users', along with a user profile indicator for 'Χάρης [Administrator]'.

Αν ο χρήστης προσπαθήσει να δημιουργήσει νέο χρήστη χωρίς να έχει συμπληρώσει τουλάχιστον τον όνομα, τον κωδικό σύνδεσης και το password, τότε του εμφανίζεται μήνυμα που τον παροτρύνει να τα συμπληρώσει.

Με το πάτημα “Post” γίνεται η αποδοχή και το νέο έργο αποθηκεύεται στο σύστημα. Αντίθετα με το πάτημα “Cancel” γίνεται ακύρωση της ενέργειας και επιστρέφει ο χρήστης στην προηγούμενη οθόνη.

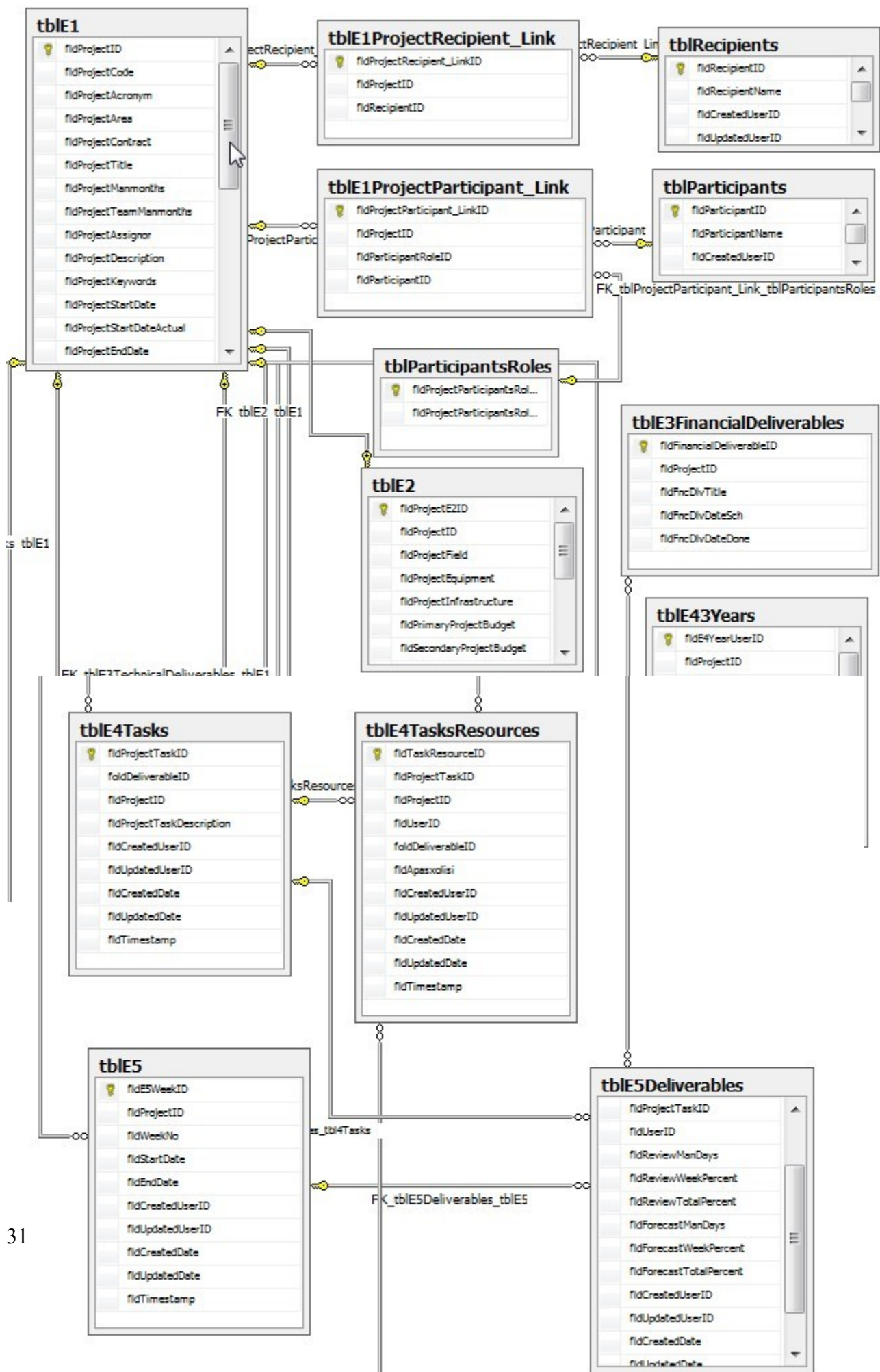
Η διαγραφή χρήστη γίνεται αντίστοιχα με την διαγραφή έργου. Εμφανίζεται σχετικός διάλογος που επιτρέπει στο χρήστη να επιβεβαιώσει την ενέργεια αυτή με οριστικές συνέπειες.

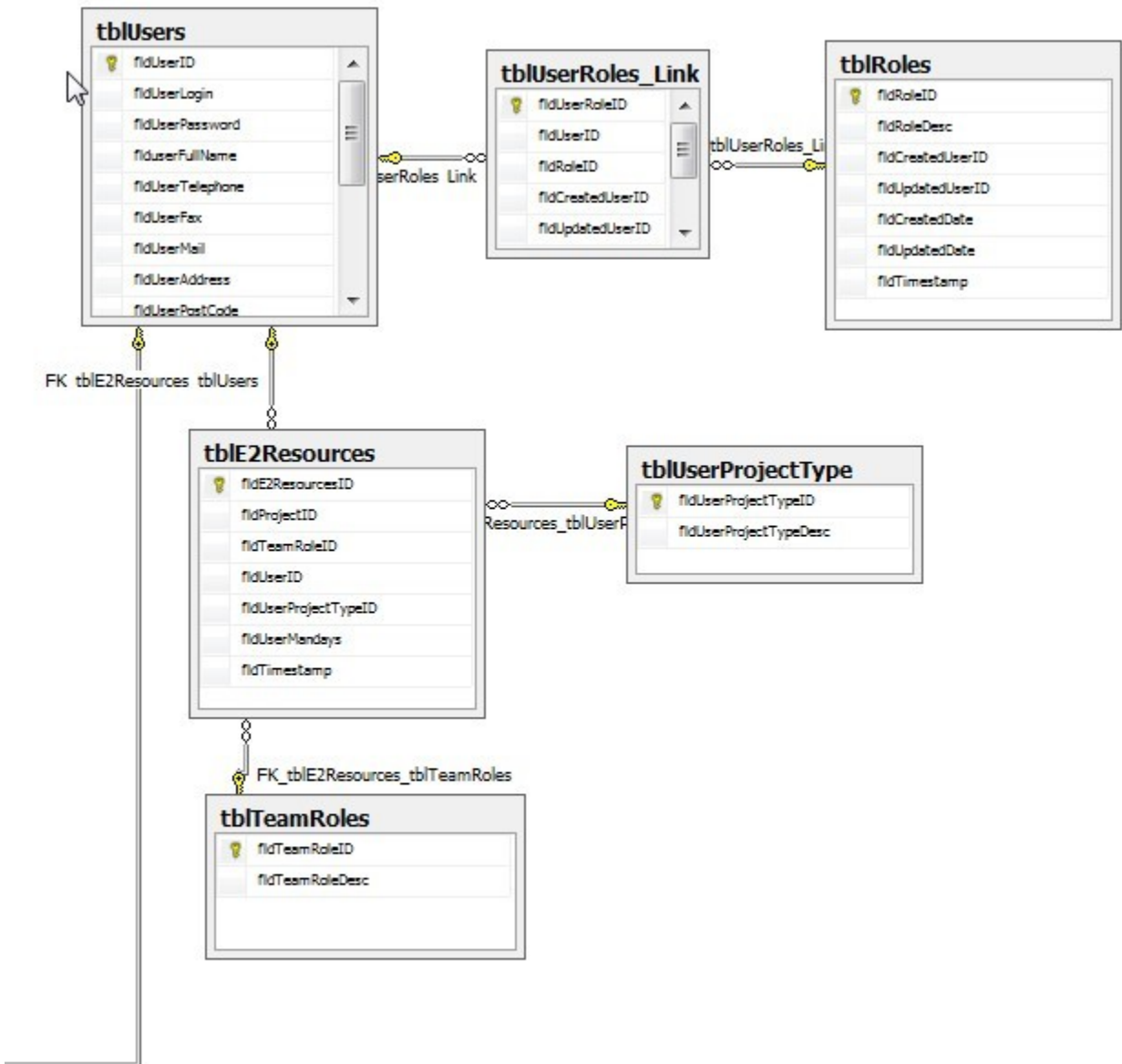
ΠΑΡΑΡΤΗΜΑ

Η βάση δεδομένων

Κύριο συστατικό της εφαρμογής είναι η βάση δεδομένων από την οποία παίρνει και αποθηκεύει τις πληροφορίες του το πρόγραμμα. Η βάση αυτή είναι εγκατεστημένη σε έναν Microsoft Sql Server ο οποίος μπορεί να είναι εγκατεστημένος σε οποιοδήποτε μηχάνημα του τοπικού δικτύου στο οποίο εγκαθίσταται η εφαρμογή. Οι πληροφορίες σύνδεσης με αυτόν τον server (connection string) πρέπει να συμπληρωθούν στο αρχείο web.conf στην μεταβλητή ConnectionString. Η βάση μας αποτελείται από 22 πίνακες.

Στην συνέχεια παρατίθενται όλοι οι πίνακες και οι συσχετίσεις τους.





Η τεχνολογία Silverlight

Το Silverlight είναι μια πλατφόρμα ανάπτυξης και εκτέλεσης εφαρμογών κυρίως διαδικτύου (αλλά όχι μόνο) από την Microsoft. Βασίζεται στην ευρεία τεχνολογία .Net. Το Silverlight χρησιμοποιείται για την δημιουργία πλούσιων εφαρμογών Internet (RIA – Rich Internet Applications). Είναι τόσο μια τεχνολογία διεπαφής χρήσης όσο και μια ισχυρή πλατφόρμα για την δημιουργία διαδικτυακών εφαρμογών. Αποτελείται από 4 βασικά μέρη:

1. Το Silverlight Plug-In (απαραίτητο πρόσθετο στην εφαρμογή περιήγησης)
2. Η ιστοσελίδα που φιλοξενεί την εφαρμογή Silverlight.
3. Το αρχείο .XAP που είναι η ίδια Silverlight εφαρμογή.
4. Η γλώσσα προγραμματισμού διεπαφής χρήστη XAML (eXtensible Application Markup Language).



Το Silverlight Plug-In είναι ο «κινητήρας» που εκτελεί την εφαρμογή Silverlight μέσα στον περιηγητή (browser). Η εφαρμογή μπορεί να φιλοξενηθεί είτε μέσα σε ιστοσελίδα HTML με ειδικό tag <object>, είτε σε ιστοσελίδα Aspx σε μορφή Silverlight control ή χρησιμοποιώντας MediaPlayer control. Στην παρούσα διπλωματική, η εφαρμογή φιλοξενηθεί σε ιστοσελίδα aspx σε μορφή silverlight control.

αρχείο New_IsoTrackWebAppTestPage.aspx

```
(...)  
<body>  
  <form id="form1" runat="server" style="height:100%">  
    <div id="silverlightControlHost">  
      <object data="data:application/x-silverlight-2," type="application/x-  
silverlight-2" width="100%" height="100%">  
        <param name="source"  
value="ClientBin/New_IsoTrackWebApp.xap"/>  
        <param name="onError" value="onSilverlightError" />  
        <param name="background" value="white" />  
        <param name="minRuntimeVersion" value="4.0.50826.0" />  
        <param name="autoUpgrade" value="true" />  
        <a href="http://go.microsoft.com/fwlink/?  
LinkID=149156&v=4.0.50826.0" style="text-decoration:none">  
            
        </a>  
      </object><iframe id="_sl_historyFrame"  
style="visibility:hidden;height:0px;width:0px;border:0px"></iframe></div>  
    </form>  
</body>
```

Σημειώσαμε με έντονους χαρακτήρες τα σημεία του κώδικα όπου γίνεται η αναφορά στην εφαρμογή Silverlight (ειδικότερα το αρχείο .xap) μέσα στην ιστοσελίδα. Το αρχείο .xap είναι η συμπιεσμένη μορφή της εφαρμογής όπως το παράγει το Visual Studio. Περιέχει ότι χρειάζεται για την εκτέλεση: αρχεία .xaml, .net assemblies και άλλους πόρους (resources files).

Το Silverlight Xaml είναι μια γλώσσα προγραμματισμού σχεδίασης διεπαφής χρήστη, πιο προχωρημένη από παραδοσιακό HTML επειδή επιτρέπει την επέκταση της γλώσσα μέσω προσαρμοσμένων σημάτων (custom tags). Ένα τυπικό «αντικείμενο» xaml, έστω “myobject” αποτελείται από 2 αρχεία: myobject.xaml και myobject.cs (code behind). Το πρώτο αρχείο περιέχει τον ορισμό της φυσικής διεπαφής χρήστη. Και το δεύτερο αρχείο περιέχει την εσωτερική λογική λειτουργία και τον ορισμό της ανταπόκρισης του αντικειμένου στην ενέργειες του χρήστη.

Η τεχνολογία WCF RIA Services

Όπως το αναφέραμε προηγουμένως, κύριο συστατικό της εφαρμογής είναι η βάση δεδομένων από την οποία παίρνει και αποθηκεύει τις πληροφορίες του το πρόγραμμα. Παραδοσιακά, η βάση δεδομένων σε μια καταναυημένη εφαρμογή (όπως εφαρμογές διαδικτύου) βρίσκεται σε απομακρυσμένη τοποθεσία. Η επικοινωνία μεταξύ διεπαφή χρήστη (client-side ui) και την βάση δεδομένων εξασφαλίζεται από μια εφαρμογή-εξυπηρετήσης (application server). Τίθεται λοιπόν ανάγκη κατάλληλου μηχανισμού πρόσβασης στην βάση δεδομένων και αποθήκευσης αλλαγών. Επιλέχτηκε η τεχνολογία WCF RIA Services (Windows Communication Foundation for Riche Internet Applications). Παρέχει ένα σύνολο από βιβλιοθήκες που μας επιτρέπουν να κατασκευάσουμε αντικείμενα με πλούσιες δυνατότητες όπως ανάκτηση δεδομένων από την βάση, ενημέρωση βάσης, εφαρμογή αυστηρών ελέγχων ορθότητας δεδομένων, ασύγχρονη επικοινωνία με την απομακρυσμένη διεπαφή χρήστη...κτλ Τα RIA services χρησιμοποιούν μια σειρά από συστατικά όσο στο client τόσο και στο server για να επιτυγχάνουν την ομαλή σύνδεση απομακρυσμένων στοιχείων της εφαρμογής.

1. **Libraries project links:** Η μεριά της εφαρμογής που βρίσκεται στο server περιέχει τους ορισμούς και αναφορές σε διάφορα άλλα αντικείμενα και βιβλιοθήκες (Domain services, metadata, shared code). Με βάση αυτή την πληροφορία, το Visual Studio παράγει αυτόματα κατάλληλο κώδικα στην μεριά του client κατά την μεταγλώττιση.
2. **Domain Services:** Αυτά τα αντικείμενα είναι ο πυρήνας της τεχνολογίας RIA. Εδώ ορίζονται (από μεριά server) οι διαδικασίες δημιουργία-διάβασμα-μεταβολή-διαγραφή οντοτήτων στην βάση αλλά και οποιεσδήποτε άλλες διαδικασίες που υποστηρίζουν την λογική και τους κανόνες ορθότητας δεδομένων της εφαρμογής. Με την βοήθεια του Visual Studio, κατάλληλος κώδικας παράγεται αυτόματα στην μεριά του client που θα επιτρέπει εύκολη κλίση αυτών των διαδικασιών.
3. **Entities:** Αυτές οι οντότητες είναι αντικείμενα που αντιπροσωπεύουν τους πίνακες της βάσης δεδομένων μαζί με τα πεδία τους και περιέχουν τους κανόνες ορθότητας δεδομένων εξασφαλίζοντας έτσι την ακεραιότητα της εφαρμογής. Όπως και προηγουμένως, κι εδώ το Visual Studio θα παράγει κατάλληλο κώδικα στην μεριά του client για απομακρυσμένη διαχείριση αυτών των αντικειμένων.
4. **Domain Context:** Αυτό το αντικείμενο είναι το βασικό κομμάτι στην μεριά του client. Παράγεται αυτόματα από το Visual Studio και δίνει δυνατότητα πρόσβασης λειτουργιών στο απομακρυσμένο server. Εσωτερικά αποτελείται από WCF proxy που κάνει όλες τις κλήσεις στο server και εντοπίζει τυχόν αλλαγές που έχουν γίνει στα αντικείμενα.
5. **DomainDataSource:** Είναι το αντικείμενο που σε συνεργασία με το Domain Context

εκτελεί τα ερωτήματα βάσης (queries), αποθηκεύει προσωρινά δεδομένα στο client για περαιτέρω επεξεργασία και στέλνει τις αλλαγές πίσω για ενημέρωση της βάσης.

Δείγμα υλοποιημένου κώδικα

Παρατίθεται μέρος του κώδικα από σελίδες οι οποίες παρουσιάζουν προγραμματιστικό ενδιαφέρον.

Login.xaml

```
<UserControl x:Class="New_IsoTrackWebApp.login.login"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d"
  d:DesignHeight="300" d:DesignWidth="600" >

  <Grid x:Name="LayoutRoot" Background="White">

    <Border Grid.Row="0" BorderBrush="Gray" BorderThickness="0.3"
      CornerRadius="8" VerticalAlignment="Center" Height="300" Width="500" >
      <Grid>
        <Grid.RowDefinitions >
          <RowDefinition Height="80" />
          <RowDefinition Height="80"/>
          <RowDefinition Height="*/>
        </Grid.RowDefinitions>
        <StackPanel Orientation="Horizontal" Grid.Row="0" HorizontalAlignment="Center"
          Margin="0,20,0,0">
          <TextBlock Text="Login name" VerticalAlignment="Center" />
          <TextBox Name="txt_loginname" VerticalAlignment="Center" Margin="30,0,0,0"
            Width="300"/>
        </StackPanel>
        <StackPanel Orientation="Horizontal" Grid.Row="1" HorizontalAlignment="Center"
          Margin="0,20,0,0">
          <TextBlock Text="Password" VerticalAlignment="Center" />
          <PasswordBox Name="txt_password"
            VerticalAlignment="Center" Margin="50,0,0,0" Width="300"/>
        </StackPanel>
        <StackPanel Orientation="Horizontal" HorizontalAlignment="Right" Height="50"
          Grid.Row="2" VerticalAlignment="Bottom" Margin="0,0,20,20">
          <Button Content="Login" Width="100" Margin="0,0,10,0" Name="btnLogin"
            FontSize="14" />
        </StackPanel>
      </Grid>
    </Border>

  </Grid>
</UserControl>
```

Login.cs

```
namespace New_IsoTrackWebApp.login
{
    public partial class login : UserControl, ILogin
    {
        OnLoginHandler OnLogin;
        public login(OnLoginHandler OnLogin)
        {
            InitializeComponent();
            this.OnLogin = OnLogin;
            btnLogin.Click += new RoutedEventHandler(btnLogin_Click);
        }

        void btnLogin_Click(object sender, RoutedEventArgs e)
        {
            btnLogin.IsEnabled = false;
            OnLogin(txt_loginname.Text,txt_password.Password, this);
        }

        string EncrypteString(string str)
        {
            byte[] s = System.Text.Encoding.Unicode.GetBytes(str);
            return Convert.ToBase64String(s);
        }
        public delegate void OnLoginHandler(string user_name, string password,
            Ilogin ilogin);

        public void Login_Completed(bool failed)
        {
            btnLogin.IsEnabled = failed;
        }
    }

    public interface ILogin
    {
        void Login_Completed(bool failed);
    }
}
```

[DataForm.cs](#)

Η φόρμα DataForm.cs είναι η βασική φόρμα από την οποία κληρονομούν όλες οι άλλες φόρμες της εφαρμογής προχυμένου να έχουν δυνατότητες ενημέρωσης απομακρυσμένης βάσης. Αυτή η φόρμα δηλαδή περιέχει τις βασικές λειτουργίες **PostData** και το αντικείμενο **DataSource** Data_Source

```
using System;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Ink;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using New_IsoTrackWebApp.Data;
using IsotrackWebApp.Library.Messages;
using System.Reflection;
using System.Collections.Generic;
using System.Linq;
using Telerik.Windows.Controls;

namespace New_IsoTrackWebApp.Common
{
    public class DataForm: UserControl
    {
        DataSource _data_source;
        protected DataSource Data_Source
        {
            get
            {
                if (_data_source == null)
                {
                    _data_source = Get_DataSource();
                }
                return _data_source;
            }
        }

        protected virtual DataSource Get_DataSource()
        {
            return null;
        }

        public virtual void PostData(Data.OnDataPostedHandler p)
        {
            Data_Source.SubmitChanges(s =>
            {
```

```

        if (s != null)
        {
            switch (s.HasError)
            {
                case true:
                    ErrorWindow.ShowError(s.Error.Message);
                    break;
                case false:
                    SuccessWindow.ShowMessage();
                    break;
            }
        }
        p(s);
    });
}

IEnumerable<DependencyObject> GetChildsRecursive(DependencyObject root)
{
    List<DependencyObject> elts = new List<DependencyObject>();
    if (root is TabControl)
    {
        TabControl tbc = root as TabControl;
        tbc.Items.Cast<TabItem>().ToList().ForEach(t =>
        {
            elts.AddRange(GetChildsRecursive(t));
        });
    }
    else
    {
        elts.Add(root);
        for (int i = 0; i < VisualTreeHelper.ChildrenCount(root); i++)
            elts.AddRange(GetChildsRecursive(VisualTreeHelper.GetChild(root, i)));
    }
    return elts;
}

protected void enable_controls(DependencyObject root, bool Enabled)
{
    IEnumerable<DependencyObject> list = GetChildsRecursive(root);

    foreach (DependencyObject obj in list)
    {
        if (obj is TextBox)
        {
            (obj as TextBox).IsReadOnly = !Enabled;
        }
        if (obj is RadNumericUpDown)
        {
            (obj as RadNumericUpDown).IsEnabled = Enabled;
        }
    }
}

protected void Lock_and_wait()
{
    this.Cursor = Cursors.Wait;
    this.IsEnabled = false;
}

protected void Release()
{
    this.Cursor = null;
    this.IsEnabled = true;
}

public virtual void CancelChanges()
{
    Data_Source.CancelChanges();
}

```

```

    }

    public bool Has_Changes
    {
        get
        {
            return Internal_Has_Changes();
        }
    }

    protected virtual bool Internal_Has_Changes()
    {
        return Data_Source.HasChanges;
    }
}
}

```

DataSource.cs

Εδώ ορίζεται το χρήσιμο αντικείμενο DataSource που σε συνδυασμό με το DomainContext είναι ο πυρήνας της εφαρμογής από την μεριά του client. Περιέχει τις μεθόδους **LoadData** (που εκτελεί συγκεκριμένη διαδικασία σε απομακρυσμένο server (συνήθως ερώτημα βάσης - query)), **SubmitChanges** (ενημέρωση βάσης με τις αλλαγές), **CancelChanges** (ακύρωση αλλαγών). Το αντικείμενο DataSource βρίσκεται σε κάθε φόρμα τύπου DataForm σε μορφή εσωτερικό μέλος (Property) Data_Source.

```

using System;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Ink;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using System.ServiceModel.DomainServices.Client;
using System.Collections.Generic;
using IsoTrackWebApp.Library.Messages;

namespace New_IsoTrackWebApp.Data
{
    public class DataSource
    {
        DomainDataSource ds;
        DomainContext _service_context;

        public DataSource(DomainContext service_context)
        {
            this._service_context = service_context;
            ds = new DomainDataSource();
            ds.DomainContext = service_context;

            ds.LoadedData += new EventHandler<LoadedDataEventArgs>(ds_LoadedData);
            ds.SubmittedChanges += new
                EventHandler<SubmittedChangesEventArgs>(ds_SubmittedChanges);
        }

        public DomainDataSourceView Items

```

```

    {
        get
        {
            return ds.DataView;
        }
    }

    public DomainContext Service_Context
    {
        get
        {
            return _service_context;
        }
    }

    OnDataLoadedHandler Internal_OnDataLoaded;
    OnDataPostedHandler Internal_OnDataPosted;

    public void LoadData(string operation, List<Parameter> Params, OnDataLoadedHandler
    DataLoaded)
    {
        ds.QueryName = operation;
        ds.QueryParameters.Clear();
        if ((Params != null) && (Params.Count > 0))
        {
            Params.ForEach(p =>
            {
                ds.QueryParameters.Add(p);
            });
        }
        this.Internal_OnDataLoaded = DataLoaded;
        ds.Load();
    }

    public void SubmitChanges( OnDataPostedHandler p)
    {
        if (!HasChanges)
        {
            if (p != null)
            {
                p(null);
            }
        }
        else
        {
            try
            {
                Validate_data();
            }
            catch(Exception ex )
            {
                ErrorWindow.ShowError(ex.Message);
                return;
            }

            this.Internal_OnDataPosted = p;
            ds.SubmitChanges();
        }
    }

    public virtual void DeleteData(long KeyID, OnDataDeletedHandler d)
    {
    }

    protected virtual void Validate_data()
    {
    }

```

```

public void CancelChanges()
{
    ds.RejectChanges();
}

public bool HasChanges
{
    get
    {
        return ds.HasChanges;
    }
}

void ds_SubmittedChanges(object sender, SubmittedChangesEventArgs e)
{
    if (Internal_OnDataPosted != null)
    {
        Internal_OnDataPosted(e);
    }
}

void ds_LoadedData(object sender, LoadedDataEventArgs e)
{
    if (Internal_OnDataLoaded != null)
    {
        Internal_OnDataLoaded(e);
    }
}

public delegate void OnDataLoadedHandler(LoadedDataEventArgs e);
public delegate void OnDataPostedHandler(SubmittedChangesEventArgs e);
public delegate void OnDataDeletedHandler(InvokeOperation<bool> e);
}

```

GridForm.cs

Αυτή η φόρμα είναι η βασική φόρμα που κληρονομεί κάθε άλλη φόρμα της εφαρμογή προχυμένου να αποκτήσει λειτουργίες «λίστας» δηλαδή λίστα οντοτήτων (όπως Projects List και Users List). Οι λειτουργίες λίστες είναι 3 ενέργειες New,View,Delete (ανάλογα σε πια οντότητα βρισκόμαστε Projects List ή Users list). Μια φόρμα τύπου GridForm περιέχει εσωτερικό πεδίο Data_Source που την επιτρέπει να «κατεβάσει» δεδομένα από την βάση δεδομένων προχυμένου να τα παρουσιάσει σε μορφή λίστα στο χρήστη.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using System.ServiceModel.DomainServices.Client;
using New_IsoTrackWebApp.tblE1.tblE1_UI;
using New_IsoTrackWebApp.Data;
using New_IsoTrackWebApp.Common;
using IsoTrackWebApp.Library.Messages;
using System.Reflection;

```



```

using New_IsoTrackWebApp.tblUsers.tblUser;

namespace New_IsoTrackWebApp.Common
{
    public class GridForm: UserControl, ITabControl
    {
        string LoadOperation;
        string KeyField;
        string KeyName;

        public GridForm()
        {
            Internal_InitializeComponents();
        }

        public GridForm(string LoadOperation, string KeyField, string KeyName)
        {
            Internal_InitializeComponents();

            this.LoadOperation = LoadOperation;
            this.KeyField = KeyField;
            this.KeyName = KeyName;

            Get_btnAdd().Click += new RoutedEventHandler(btnAdd_Click);
            Get_btnDelete().Click += new RoutedEventHandler(btnDelete_Click);
            Get_btnEdit().Click += new RoutedEventHandler(btnEdit_Click);
            Tab_Control.SelectionChanged += new
                SelectionChangedEventHandler(Tab_Control_SelectionChanged);
            Load_Data(null);
        }

        #region DataSource

        DataSource _Data_Source = null;
        protected DataSource Data_Source
        {
            get
            {
                if (_Data_Source == null)
                {
                    _Data_Source = Get_DataSource();
                }
                return _Data_Source;
            }
        }

        #endregion

        #endregion

        #region Add operation

        protected virtual Button Get_btnAdd()
        {
            return null;
        }

        // Ενέργεια New
        protected virtual void AddNewRecord()
        {
        }
    }
}

```

```

void btnAdd_Click(object sender, RoutedEventArgs e)
{
    AddNewRecord();
}

#endregion

#region Delete operation

protected virtual Button Get_btnDelete()
{
    return null;
}

// Επέγεια Delete
void btnDelete_Click(object sender, RoutedEventArgs e)
{
    long KeyID = 0;

    if (Tab_Control.SelectedIndex == 0)
    {
        if (Current_Item == null)
            return;

        KeyID = Convert.ToInt64(Get_Value(Current_Item, KeyField));
    }
    else if (Tab_Control.SelectedIndex > 0)
    {
        TabItem tb = Tab_Control.SelectedItem as TabItem;

        KeyValuePair<int, TabItem> KeyVal = Opened_Records.FirstOrDefault(v =>
        {
            return v.Value == tb;
        });

        KeyID = Convert.ToInt64(KeyVal.Key);
    }

    QuestionWindow.AskQuestion("Do you really want to delete ?", p =>
    {
        if (p)
        {
            Data_Source.DeleteData(KeyID, d =>
            {
                Ignore_changes = true;
                try
                {
                    Close(Convert.ToInt32(KeyID));
                }
                finally
                {
                    Ignore_changes = false;
                }
            });
        }
    });
}

#endregion

#region Edit operation

protected virtual Button Get_btnEdit()
{
    return null;
}

```

```

void btnEdit_Click(object sender, RoutedEventArgs e)
{
    Edit_Record();
}

public int KeyValue
{
    get
    {
        return Convert.ToInt32(Get_Value(Current_Item, KeyField));
    }
}

// Ενέργια View
void Edit_Record()
{
    if (Current_Item == null)
        return;

    if (Is_Opened(KeyValue))
    {
        Tab_Control.SelectedItem = Opened_Records[KeyValue];
    }
    else
    {
        TabItem tb = get_new_tab();
        Opened_Records[KeyValue] = tb;
        tb.Header = new CustomTabItem(this, string.Format("{0} ",
            Convert.ToString(Get_Value(Current_Item, KeyName))), KeyValue);
        tb.Content = Get_EntryForm(KeyValue);
        this.Tab_Control.SelectedItem = tb;
        this.Tab_Control.Items.Cast<TabItem>().ToList().ForEach(t =>
        {
            if (t != tb)
            {
                if (t.Header is CustomTabItem)
                {
                    (t.Header as CustomTabItem).DeActivateHeader();
                }
            }
        });
    }
}

Dictionary<int, TabItem> Opened_Records = new Dictionary<int, TabItem>();

bool Is_Opened(int ID)
{
    return Opened_Records.ContainsKey(ID);
}

object Get_Value( Entity Entity, string Property )
{
    PropertyInfo pInfo = Entity.GetType().GetProperty(Property);
    return pInfo.GetValue(Current_Item, new object[] { });
}

TabItem get_new_tab()
{
    TabItem tb = new TabItem();
}

```

```

        tb.VerticalAlignment = System.Windows.VerticalAlignment.Center;
        tb.HorizontalAlignment = System.Windows.HorizontalAlignment.Left;
        tb.Margin = new Thickness { Top = 0, Left = 10, Bottom = 0, Right = 0 };
        Tab_Control.Items.Add(tb);
        return tb;
    }

#endregion

#region Load_Data

protected virtual void Load_Data( OnDataLoadedHandler DataLoaded)
{
    Data_Source.LoadData(LoadOperation, Get_Parameters(), p =>
    {
        Data_Grid.ItemsSource = Data_Source.Items;

        if (DataLoaded != null)
        {
            DataLoaded(p);
        }
    });
}

#endregion

bool Ignore_changes = false;

public void Close(int ID)
{
    if (!Opened_Records.ContainsKey(ID))
    {
        return;
    }
    TabItem tb = Opened_Records[ID];

    if (Ignore_changes)
    {
        Tab_Control.Items.Remove(tb);
    }
    else
    {
        bool CanClose = true;
        if (tb.Content is Main_UI)
        {
            Main_UI main = tb.Content as Main_UI;
            CanClose = main.Can_Close();
        }
        else
        {
            User_UI usr_ui = (User_UI)tb.Content;
            CanClose = !usr_ui.Has_Changes;
        }

        if (!CanClose)
        {
            QuestionWindow.AskQuestion("All pending changes will be lost.
            Continue?", p =>
            {
                switch (p)

```

```

        {
            case true:
                Tab_Control.Items.Remove(tb);
                break;
            case false:
                break;
        }
    });
}
else
{
    Tab_Control.Items.Remove(tb);
    KeyValuePair<int, TabItem> kv = Opened_Records.FirstOrDefault(k =>
        k.Value == tb);
    Opened_Records.Remove(kv.Key);
}
}
}
}
}
}
}
}
}
}

```

E1_UI.xaml

```

<common:DataForm
  x:Class="New_IsoTrackWebApp.tblE1.tblE1_UI.E1_UI"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:common="clr-namespace:New_IsoTrackWebApp.Common"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:sdk="http://schemas.microsoft.com/winfx/2006/xaml/presentation/sdk"
  xmlns:telerik="http://schemas.telerik.com/2008/xaml/presentation"
  mc:Ignorable="d"
  d:DesignHeight="800" d:DesignWidth="1000" FontSize="17">

  <Grid x:Name="LayoutRoot" Background="White" >

    <Grid.RowDefinitions>

      <RowDefinition Height="*" />
      <RowDefinition Height="40" />

    </Grid.RowDefinitions>

    <sdk:TabControl x:Name="tab_control" Grid.Row="0" BorderThickness="0" >

      <!-- Main Information-->
      <sdk:TabItem BorderThickness="0" x:Name="tb_main" >

        <sdk:TabItem.Header >
          <Border Grid.Row="0" CornerRadius="5" >

            <StackPanel Orientation="Horizontal">
              <TextBlock Text="Main" FontSize="17" Margin="10,0,0,0"
                VerticalAlignment="Center" FontWeight="SemiBold"
                Foreground="Red" Opacity="0.4" />
              <TextBlock Text="Information" FontSize="17" Margin="3,0,0,0"
                VerticalAlignment="Center"
                Foreground="Gray" Opacity="0.8" />
            </StackPanel>
          </Border>
        </sdk:TabItem.Header>
      </sdk:TabItem>
    </TabControl>
  </Grid>

```

```

    </Border>
</sdk:TabItem.Header>

<StackPanel Name="stack_panell1" Grid.Row="0">
  <!-- fldProjectCode , fldProjectAcronym-->
  <Grid Margin="0,15,0,0">
    <Grid.ColumnDefinitions >
      <ColumnDefinition Width="200"/>
      <ColumnDefinition Width="300" />
      <ColumnDefinition Width="200"/>
      <ColumnDefinition Width="300" />
    </Grid.ColumnDefinitions>
    <TextBlock Grid.Column="0" Text="Project code" VerticalAlignment="Center"
      FontWeight="SemiBold" Foreground="Gray" />
    <TextBox Grid.Column="1" Text="{Binding fldProjectCode, Mode=TwoWay}" />

    <TextBlock Grid.Column="2" Text="Project acronyme" VerticalAlignment="Center"
      Margin="10,0,0,0" FontWeight="SemiBold" Foreground="Gray"/>
    <TextBox Grid.Column="3" Text="{Binding fldProjectAcronym, Mode=TwoWay}" />
  </Grid>

  <!-- projecttitle -->
  <Grid Margin="0,15,0,0">
    <Grid.ColumnDefinitions >
      <ColumnDefinition Width="200" />
      <ColumnDefinition />
    </Grid.ColumnDefinitions>
    <TextBlock Grid.Column="0" Text="Project title" VerticalAlignment="Center"
      FontWeight="SemiBold" Foreground="Gray"/>
    <TextBox Grid.Column="1" Text="{Binding fldProjectTitle, Mode=TwoWay}" />
  </Grid>

  <!-- projectdescription -->
  <Grid Margin="0,15,0,0">
    <Grid.ColumnDefinitions >
      <ColumnDefinition Width="200" />
      <ColumnDefinition />
    </Grid.ColumnDefinitions>
    <TextBlock Grid.Column="0" Text="Project description" VerticalAlignment="Center"
      FontWeight="SemiBold" Foreground="Gray"/>
    <TextBox Grid.Column="1" Text="{Binding fldProjectDescription, Mode=TwoWay}"
      Height="100" TextWrapping="Wrap" AcceptsReturn="true" />
  </Grid>

  <!-- fldProjectAssignor -->
  <Grid Margin="0,15,0,0">
    <Grid.ColumnDefinitions >
      <ColumnDefinition Width="200" />
      <ColumnDefinition />
    </Grid.ColumnDefinitions>
    <TextBlock Grid.Column="0" Text="Project assignor" VerticalAlignment="Center"
      FontWeight="SemiBold" Foreground="Gray"/>
    <TextBox Grid.Column="1" Text="{Binding fldProjectAssignor, Mode=TwoWay}" />
  </Grid>

  <!-- fldProjectArea -->
  <Grid Margin="0,15,0,0">
    <Grid.ColumnDefinitions >
      <ColumnDefinition Width="200" />
      <ColumnDefinition />
    </Grid.ColumnDefinitions>
    <TextBlock Grid.Column="0" Text="Project area" VerticalAlignment="Center"
      FontWeight="SemiBold" Foreground="Gray"/>
    <TextBox Grid.Column="1" Text="{Binding fldProjectArea, Mode=TwoWay}" />
  </Grid>

  <!-- fldProjectContract -->
  <Grid Margin="0,15,0,0">
    <Grid.ColumnDefinitions >
      <ColumnDefinition Width="200" />

```

```

        <ColumnDefinition />
    </Grid.ColumnDefinitions>
    <TextBlock Grid.Column="0" Text="Project contract" VerticalAlignment="Center"
        FontWeight="SemiBold" Foreground="Gray"/>
    <TextBox Grid.Column="1" Text="{Binding fldProjectContract, Mode=TwoWay}" />
</Grid>

</StackPanel>

</sdk:TabItem>

<!-- Project durations-->
<sdk:TabItem BorderThickness="0" x:Name="tb_duration">

    <sdk:TabItem.Header >
        <Border Grid.Row="0" CornerRadius="5" >
            <StackPanel Orientation="Horizontal">
                <TextBlock Text="Project" FontSize="17" Margin="10,0,0,0"
                    VerticalAlignment="Center" FontWeight="SemiBold"
                    Foreground="Red" Opacity="0.4" />
                <TextBlock Text="Durations" FontSize="17" Margin="3,0,0,0"
                    VerticalAlignment="Center"
                    Foreground="Gray" Opacity="0.8" />
            </StackPanel>
        </Border>
    </sdk:TabItem.Header>

    <StackPanel Name="stack_panel12" Grid.Row="0">

        <!-- fldProjectDuration , fldProjectDurationActual-->
        <Grid Margin="0,15,0,0">
            <Grid.ColumnDefinitions >
                <ColumnDefinition Width="200"/>
                <ColumnDefinition Width="210" />
                <ColumnDefinition Width="240"/>
                <ColumnDefinition Width="300" />
            </Grid.ColumnDefinitions>

            <Grid.RowDefinitions >
                <RowDefinition />
                <RowDefinition />
            </Grid.RowDefinitions>

            <TextBlock Grid.Row="0" Grid.Column="0" Text="Duration" VerticalAlignment="Center"
                FontWeight="SemiBold" Foreground="Gray" />
            <telerik:RadNumericUpDown Name="ctrl_duration" Grid.Row="0" Grid.Column="1" Value
               ="{Binding fldProjectDuration, Mode=TwoWay}" Width="200"
                HorizontalAlignment="Left"/>

            <TextBlock Grid.Row="0" Grid.Column="2" Text="Actual duration"
                VerticalAlignment="Center" Margin="0,0,0,0" FontWeight="SemiBold"
                Foreground="Gray"/>
            <telerik:RadNumericUpDown Name="ctrl_actual_duration" Grid.Row="0" Grid.Column="3"
                Value="{Binding fldProjectDurationActual, Mode=TwoWay}" Width="200"
                HorizontalAlignment="Left"/>

            <TextBlock Margin="0,10,0,0" Grid.Row="1" Grid.Column="2" Text="Revised duration"
                VerticalAlignment="Center" FontWeight="SemiBold" Foreground="Gray" />
            <telerik:RadNumericUpDown Name="ctrl_revised_duration" Margin="0,10,0,0" Grid.Row="1"
                Grid.Column="3" Value="{Binding fldProjectDurationRevision, Mode=TwoWay}"
                Width="200" HorizontalAlignment="Left"/>

        </Grid>

        <!-- fldProjectStartDate-->
        <Grid Margin="0,20,0,0">
            <Grid.ColumnDefinitions >
                <ColumnDefinition Width="200"/>
                <ColumnDefinition Width="210" />
            </Grid.ColumnDefinitions>

```

```

        <ColumnDefinition Width="240"/>
        <ColumnDefinition Width="300" />
    </Grid.ColumnDefinitions>

    <Grid.RowDefinitions >
        <RowDefinition />
        <RowDefinition />
    </Grid.RowDefinitions>

    <TextBlock Grid.Row="0" Grid.Column="0" Text="Start date" VerticalAlignment="Center"
        FontWeight="SemiBold" Foreground="Gray" />
    <telerik:RadDatePicker Name="ctrl_startdate" Grid.Row="0" Grid.Column="1"
        SelectedDate="{Binding fldProjectStartDate, Mode=TwoWay}" Width="200"
        HorizontalAlignment="Left"/>

    <TextBlock Grid.Row="0" Grid.Column="2" Text="End date" VerticalAlignment="Center"
        Margin="0,0,0,0" FontWeight="SemiBold" Foreground="Gray"/>
    <telerik:RadDatePicker Name="ctrl_enddate" Grid.Row="0" Grid.Column="3"
        SelectedDate="{Binding fldProjectEndDate, Mode=TwoWay}" Width="200"
        HorizontalAlignment="Left"/>

    <TextBlock Margin="0,10,0,0" Grid.Row="1" Grid.Column="2" Text="Revised end date"
        VerticalAlignment="Center" FontWeight="SemiBold" Foreground="Gray" />
    <telerik:RadDatePicker Name="ctrl_revised_enddate" Margin="0,10,0,0" Grid.Row="1"
        Grid.Column="3" SelectedDate="{Binding fldProjectEndDateRevision,
        Mode=TwoWay}" Width="200" HorizontalAlignment="Left"/>

</Grid>

<!-- fldProjectWorkpackagesNo-->
<Grid Margin="0,20,0,0">
    <Grid.ColumnDefinitions >
        <ColumnDefinition Width="200"/>
        <ColumnDefinition Width="210" />
        <ColumnDefinition Width="240"/>
        <ColumnDefinition Width="300" />
    </Grid.ColumnDefinitions>

    <Grid.RowDefinitions >
        <RowDefinition />
        <RowDefinition />
    </Grid.RowDefinitions>

    <TextBlock Grid.Row="0" Grid.Column="0" Text="Total work package"
        VerticalAlignment="Center" FontWeight="SemiBold" Foreground="Gray" />
    <telerik:RadNumericUpDown Name="ctrl_workpackage" Grid.Row="0" Grid.Column="1" Value
        ="{Binding fldProjectWorkpackagesNo, Mode=TwoWay}" Width="200"
        HorizontalAlignment="Left"/>

    <TextBlock Grid.Row="0" Grid.Column="2" Text="Total activities"
        VerticalAlignment="Center" Margin="0,0,0,0" FontWeight="SemiBold"
        Foreground="Gray"/>
    <telerik:RadNumericUpDown Name="ctrl_activitiesno" Grid.Row="0" Grid.Column="3"
        Value ="{Binding fldProjectActivitiesNo, Mode=TwoWay}" Width="200"
        HorizontalAlignment="Left"/>

    <TextBlock Margin="0,10,0,0" Grid.Row="1" Grid.Column="2" Text="Total deliverables"
        VerticalAlignment="Center" FontWeight="SemiBold" Foreground="Gray" />
    <telerik:RadNumericUpDown Name="ctrl_deliverables" Margin="0,10,0,0" Grid.Row="1"
        Grid.Column="4" Value ="{Binding fldProjectDeliverablesNo, Mode=TwoWay}"
        Width="200" HorizontalAlignment="Left"/>

</Grid>

</StackPanel>

</sdk:TabItem>

<!-- tb_recipients-->
<sdk:TabItem BorderThickness="0" x:Name="tb_recipients" >

```



```

        <sdk:TabItem.Header >
            <Border Grid.Row="0" CornerRadius="5" >

                <StackPanel Orientation="Horizontal">
                    <TextBlock Text="Project" FontSize="17" Margin="10,0,0,0"
                        VerticalAlignment="Center" FontWeight="SemiBold"
                        Foreground="Red" Opacity="0.4" />
                    <TextBlock Text="Recipients" FontSize="17" Margin="3,0,0,0"
                        VerticalAlignment="Center"
                        Foreground="Gray" Opacity="0.8" />
                </StackPanel>
            </Border>
        </sdk:TabItem.Header>
    </sdk:TabItem>

    <!-- tb_participants-->
    <sdk:TabItem BorderThickness="0" x:Name="tb_participants" >
        <sdk:TabItem.Header >
            <Border Grid.Row="0" CornerRadius="5" >
                <StackPanel Orientation="Horizontal">
                    <TextBlock Text="Participants" FontSize="17" Margin="3,0,0,0"
                        VerticalAlignment="Center"
                        Foreground="Gray" Opacity="0.8" />
                </StackPanel>
            </Border>
        </sdk:TabItem.Header>
    </sdk:TabItem>

    </sdk:TabControl>

</Grid>

</common:DataForm >

```

[E1_ui.cs](#)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using New_IsoTrackWebApp.Data;
using New_IsoTrackWebApp.Common;
using IsoTrackWebApp.Library.Messages;
using Telerik.Windows.Controls;

namespace New_IsoTrackWebApp.tblE1.tblE1_UI
{
    public partial class E1_UI : DataForm
    {
        // Δημιουργεία

        long ProjectID;
        public E1_UI(long ProjectID)
        {
            InitializeComponent();
            this.ProjectID = ProjectID;
        }
    }
}

```

```

load_project();
tab_control.SelectionChanged += new System.Windows.Controls.SelectionChangedEventHandler(
    tab_control_SelectionChanged);
}

// Φορτώση συγκεκριμένου έργου ProjectID
// Παρατηρήστε το αντικείμενο Data_Source (τύπου DomainDataSource) που θα
// εκτελέσει την κλήση Get_TblE1_By_ID

void load_project()
{
    Lock_and_wait();

    Data_Source.LoadData( "Get_TblE1_By_ID",
        new List<Parameter> { new Parameter { ParameterName =
            "ProjectID", Value = this.ProjectID } }, p =>
    {
        this.DataContext = Data_Source.Items.Cast<New_IsoTrackWebApp.Web.tblE1>()
            .FirstOrDefault();

        ApplicationContext.CurrentProject = (Web.tblE1)this.DataContext;

        bool enabled = ApplicationContext.Team_Role ==
            Data.TeamRole.ProjectDirector;

        enable_controls(stack_panel1, enabled);

        Controls.ForEach(c =>
        {
            if (c is RadNumericUpDown)
            {
                (c as RadNumericUpDown).IsEnabled = enabled;
            }
            if (c is RadDatePicker)
            {
                (c as RadDatePicker).IsReadOnly = !enabled;
            }
            if (c is TextBox)
            {
                (c as TextBox).IsReadOnly = !enabled;
            }
        });

        tb_recipients.Content = new E1_Rec_UI(ProjectID);
        tb_participants.Content = new E1_Part_UI(ProjectID);

        Release();
    });
}

public override void PostData(OnDataPostedHandler p)
{
    if ((tab_control.SelectedItem == tb_recipients)
        || (tab_control.SelectedItem == tb_participants))
    {
        ((DataForm)((TabItem)tab_control.SelectedItem).Content).PostData(p);
    }
    else

```

```

        {
            if (IsNew)
            {
                New_IsoTrackWebApp.Web.E1.tblE1_Context srv = new Web.E1.tblE1_Context();
                srv.Insert_New_TblE1(Data_Source.Items[0] as Web.tblE1,
                    ApplicationContext.CurrentUser.fldUserID, q =>
                {
                    if (!q.HasError)
                    {
                        SuccessWindow.ShowMessage();
                        p(null);
                    }
                    else
                    {
                        ErrorWindow.ShowError("Error: post failed");
                        p( null );
                    }
                }, null);
            }
            else
            {
                base.PostData(p);
            }
        }
    }
    public override void CancelChanges()
    {
        if ((tab_control.SelectedItem == tb_recipients)
            || (tab_control.SelectedItem == tb_participants))
        {
            ((DataForm)((TabItem)tab_control.SelectedItem).Content).CancelChanges();
        }
        else
        {
            base.CancelChanges();
        }
    }
    protected override DataSource Get_DataSource()
    {
        return new DataSource( new New_IsoTrackWebApp.Web.E1.tblE1_Context() );
    }
}

```

[tblE1_Grid.cs](#)

Αρχείο λίστας Έργων (Project list)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using System.ServiceModel.DomainServices.Client;
using New_IsoTrackWebApp.tblE1.tblE1_UI;
using New_IsoTrackWebApp.Data;
using New_IsoTrackWebApp.Common;
using IsoTrackWebApp.Library.Messages;

```

```

namespace New_IsoTrackWebApp.tblE1.Grid
{
    public partial class tblE1_Grid : GridForm
    {

        public tblE1_Grid()
            : base("GetProjectsOfCurrentUser", "fldProjectID", "fldProjectCode")
        {
        }

        protected override List<Parameter> Get_Parameters()
        {
            List<Parameter> list = new List<Parameter>();
            list.Add(new Parameter { ParameterName = "UserID", Value =
                ApplicationContext.CurrentUser.fldUserID });
            return list;
        }

        protected override void Internal_InitializeComponents()
        {
            InitializeComponent();
            datagrid.AutoGeneratingColumn += new
                EventHandler<DataGridAutoGeneratingColumnEventArgs>
                    (datagrid_AutoGeneratingColumn);
            display_roles();
        }

        void display_roles()
        {
            txt_user.Text = ApplicationContext.CurrentUser.flduserFullName;
            string Roles = string.Empty;

            bool Is_Administrator =
                ApplicationContext.CurrentUser.tblUserRoles_Link.ToList().FirstOrDefault(
                    r => r.fldRoleID == 1) != null;
            bool Is_Project_Creator =
                ApplicationContext.CurrentUser.tblUserRoles_Link.ToList().FirstOrDefault(
                    => r.fldRoleID == 2) != null;
            bool Is_User =
                ApplicationContext.CurrentUser.tblUserRoles_Link.ToList().FirstOrDefault(
                    => r.fldRoleID == 3) != null;
            bool Is_Visitor =
                ApplicationContext.CurrentUser.tblUserRoles_Link.ToList().FirstOrDefault(
                    => r.fldRoleID == 4) != null;

            if (Is_Administrator)
            {
                Roles = "Administrator";

                if (Is_Project_Creator)
                {
                    Roles += ", Project Creator";
                }
            }
            else if (Is_Project_Creator)
            {
                Roles = "Project Creator";
            }
            else if (Is_User)
            {
                Roles = "User";
            }
            else
            {
                Roles = "Visitor";
            }
        }
    }
}

```

```

    }

    txt_user.Text = string.Format("{0} [ {1} ]", txt_user.Text, Roles);
    btnNew.IsEnabled = btnDelete.IsEnabled = Is_Administrator || Is_Project_Creator;
}

void datagrid_AutoGeneratingColumn(object sender,
    DataGridAutoGeneratingColumnEventArgs e)
{
    e.Column.IsReadOnly = true;
    e.Column.Width = new DataGridLength(220);
    if (e.PropertyName == "fldProjectCode")
    {
        e.Column.Header = "Code ";
        e.Column.DisplayIndex = 0;
        e.Column.SortMemberPath = "fldProjectCode";
    }
    else if (e.PropertyName == "fldProjectAcronym")
    {
        e.Column.Header = "Acronym";
        e.Column.DisplayIndex = 1;
        e.Column.SortMemberPath = "fldProjectAcronym";
    }
    else if (e.PropertyName == "fldProjectArea")
    {
        e.Column.Header = "Area";
        e.Column.DisplayIndex = 1;
        e.Column.SortMemberPath = "fldProjectArea";
    }
    //fldProjectContract
    else if (e.PropertyName == "fldProjectContract")
    {
        e.Column.Header = "Contract";
        e.Column.DisplayIndex = 1;
        e.Column.SortMemberPath = "fldProjectContract";
    }
    //fldProjectTitle
    else if (e.PropertyName == "fldProjectTitle")
    {
        e.Column.Header = "Title";
        e.Column.DisplayIndex = 1;
        e.Column.SortMemberPath = "fldProjectTitle";
    }
    else
        e.Cancel = true;
}

protected override Button Get_btnAdd()
{
    return btnNew;
}

protected override void Load_Data(OnDataLoadedHandler DataLoaded)
{
    base.Load_Data(p =>
    {
    });
}

protected override void AddNewRecord()
{
    ChildWindow host = new ChildWindow();
    host.Content = new New_E1(host, p =>
    {
        Load_Data(null);
    });
    host.Show();
}

protected override Button Get_btnDelete()
{

```

```

        return btnDelete;
    }
    protected override Button Get_btnEdit()
    {
        return btnEdit;
    }
    protected override DataGrid Get_DataGrid()
    {
        return datagrid;
    }
    protected override DataSource Get_DataSource()
    {
        return new E1_DataSource();
    }

    protected override UserControl Get_EntryForm(int KeyValue)
    {
        return new Main_UI(Convert.ToInt64(KeyValue));
    }
    protected override TabControl Get_TabControl()
    {
        return this.tab_control;
    }

    class E1_DataSource : DataSource
    {
        public E1_DataSource()
            :base(new Web.E1.tblE1_Context())
        {
        }
        public override void DeleteData(long KeyID, OnDataDeletedHandler d)
        {
            Web.E1.tblE1_Context srv = new Web.E1.tblE1_Context();
            srv.DeleteByID(KeyID, e =>
            {
                if (e.HasError)
                {
                    ErrorWindow.ShowError(e.Error.Message);
                }
                else
                {
                    Web.tblE1 rec = this.Items.Cast<Web.tblE1>().FirstOrDefault(t => t.fldProjectID ==
                                                                                               KeyID);

                    if (rec != null)
                    {
                        Items.Remove(rec);
                    }
                    SuccessWindow.ShowMessage("Successfull record deletion");
                    d(e);
                }
            }, null);
        }
    }
}
}
}

```

[E2_UI.xaml](#)


```

<common:DataForm
  x:Class="New_IsoTrackWebApp.tblE1.tblE1_UI.E2_UI"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:common="clr-namespace:New_IsoTrackWebApp.Common"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:sdk="http://schemas.microsoft.com/winfx/2006/xaml/presentation/sdk"
  xmlns:telerik="http://schemas.telerik.com/2008/xaml/presentation"
  mc:Ignorable="d"
  d:DesignHeight="400" d:DesignWidth="800" FontSize="17">

  <Grid x:Name="LayoutRoot" Background="White" >

    <sdk:TabControl x:Name="tabcontrol" BorderThickness="0" BorderBrush="White" >

      <sdk:TabItem x:Name="tab_main" >

        <sdk:TabItem.Header>
          <Border Grid.Row="0" CornerRadius="5" >
            <StackPanel Orientation="Horizontal">
              <TextBlock Text="E2" FontSize="17" Margin="10,0,0,0"
                VerticalAlignment="Center" FontWeight="SemiBold"
                Foreground="Red" Opacity="0.4" />
              <TextBlock Text="Information" FontSize="17" Margin="3,0,0,0"
                VerticalAlignment="Center"
                Foreground="Gray" Opacity="0.8" />
            </StackPanel>
          </Border>
        </sdk:TabItem.Header>

        <Grid>

          <Grid.RowDefinitions>

            <RowDefinition Height="10" />
            <RowDefinition />

          </Grid.RowDefinitions>

          <ScrollViewer Grid.Row="1" >

            <StackPanel Name="stack_panel1" >
              <!-- fldProjectCode , fldProjectAcronym-->
              <Grid Margin="0,15,0,0">
                <Grid.ColumnDefinitions >
                  <ColumnDefinition Width="200"/>
                  <ColumnDefinition />
                </Grid.ColumnDefinitions>
                <TextBlock Grid.Column="0" Text="Project field" VerticalAlignment="Center"
                  FontWeight="SemiBold" Foreground="Gray" />
                <TextBox Grid.Column="1" Text="{Binding fldProjectField, Mode=TwoWay}"
                  AcceptsReturn="true" TextWrapping="Wrap"/>
              </Grid>

              <!-- projecttitle -->
              <Grid Margin="0,15,0,0">
                <Grid.ColumnDefinitions >
                  <ColumnDefinition Width="200" />
                  <ColumnDefinition />
                </Grid.ColumnDefinitions>
                <TextBlock Grid.Column="0" Text="Equipment" VerticalAlignment="Center"
                  FontWeight="SemiBold" Foreground="Gray"/>
                <TextBox Grid.Column="1" Text="{Binding fldProjectEquipment, Mode=TwoWay}"
                  AcceptsReturn="true" TextWrapping="Wrap"/>
              </Grid>

              <!-- fldProjectInfrastructure -->
              <Grid Margin="0,15,0,0">
                <Grid.ColumnDefinitions >

```


E2_UI.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using New_IsoTrackWebApp.Common;
using New_IsoTrackWebApp.Data;
using IsoTrackWebApp.Library.Messages;

namespace New_IsoTrackWebApp.tblE1.tblE1_UI
{
    public partial class E2_UI : DataForm
    {
        long ProjectID;
        public E2_UI(long ProjectID)
        {
            InitializeComponent();
            this.ProjectID = ProjectID;
            load_project();
            tabcontrol.SelectionChanged += new SelectionChangedEventHandler(tabControl_SelectionChanged);
        }

        bool skip_tab_changed = false;
        void tabControl_SelectionChanged(object sender, SelectionChangedEventArgs e)
        {
            if (skip_tab_changed)
                return;
            TabItem tb_prev = e.RemovedItems.Cast<TabItem>().FirstOrDefault();
            DataForm f = null; // (DataForm)tb_prev.Content;

            if (tb_prev == tab_main)
            {
                f = this;
            }
            else
            {
                f = res_ui;
            }

            if (f == null)
                return;

            if (f.Has_Changes)
            {
                QuestionWindow.AskQuestion("All pending changes will be lost. Continue?", p =>
                {
                    switch (p)
                    {
                        case true:
                            f.CancelChanges();
                            break;
                        case false:
                            skip_tab_changed = true;
                            tabcontrol.SelectedItem = tb_prev;
                            skip_tab_changed = false;

                            break;
                    }
                })
            }
        }
    }
}
```

```

    });
}

protected override DataSource Get_DataSource()
{
    return new DataSource(new New_IsoTrackWebApp.Web.E1.tblE1_Context());
}

void load_project()
{
    Data_Source.LoadData("Get_TblE2_By_ID",
        new List<Parameter> { new Parameter { ParameterName = "ProjectID", Value =
            this.ProjectID } }, p =>
        {
            this.DataContext =
                Data_Source.Items.Cast<New_IsoTrackWebApp.Web.tblE2>().FirstOrDefault();
            tab_res.Content = res_ui;
            enable_controls(stack_panel1, ApplicationContext.Team_Role ==
                Data.TeamRole.ProjectDirector);
        });
}

E2_Res_UI _res_ui;
E2_Res_UI res_ui
{
    get
    {
        if (_res_ui == null)
        {
            _res_ui = new E2_Res_UI(ProjectID);
        }
        return _res_ui;
    }
}

protected override bool Internal_Has_Changes()
{
    if (tabcontrol.SelectedItem == tab_main)
    {
        return base.Internal_Has_Changes();
    }
    else
    {
        return res_ui.Has_Changes;
    }
}

public override void PostData(OnDataPostedHandler p)
{
    if (tabcontrol.SelectedItem == tab_main)
    {
        base.PostData(p);
    }
    else
    {
        res_ui.PostData(p);
    }
}

public override void CancelChanges()
{
    if (tabcontrol.SelectedItem == tab_main)
    {
        base.CancelChanges();
    }
    else
    {

```

```

        res_ui.CancelChanges();
    }
}

DataForm get_active_form()
{
    if (tabcontrol.SelectedItem == tab_main)
    {
        return this;
    }
    else
    {
        return res_ui;
    }
}
}
}

```

[E3_UI.Xaml](#)

```

<common:DataForm xmlns:sdk="http://schemas.microsoft.com/winfx/2006/xaml/presentation/sdk"
x:Class="New_IsoTrackWebApp.tblE1.tblE1_UI.E3_UI"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:common="clr-namespace:New_IsoTrackWebApp.Common"
mc:Ignorable="d"
d:DesignHeight="300" d:DesignWidth="800">
<Grid x:Name="LayoutRoot" Background="White">

    <sdk:TabControl x:Name="tabcontrol" >

        <sdk:TabItem x:Name="tab_fin">

            <sdk:TabItem.Header>
                <Border Grid.Row="0" CornerRadius="5" >
                    <StackPanel Orientation="Horizontal">
                        <TextBlock Text="Financial" FontSize="17" Margin="10,0,0,0"
                            VerticalAlignment="Center" FontWeight="SemiBold"
                            Foreground="Red" Opacity="0.4" />
                        <TextBlock Text="Deliverables" FontSize="17" Margin="3,0,0,0"
                            VerticalAlignment="Center"
                            Foreground="Gray" Opacity="0.8" />
                    </StackPanel>
                </Border>
            </sdk:TabItem.Header>
        </sdk:TabItem>

        <sdk:TabItem x:Name="tab_tech">

            <sdk:TabItem.Header>
                <Border Grid.Row="0" CornerRadius="5" >
                    <StackPanel Orientation="Horizontal">
                        <TextBlock Text="Technical" FontSize="17" Margin="10,0,0,0"
                            VerticalAlignment="Center" FontWeight="SemiBold"
                            Foreground="Red" Opacity="0.4" />
                        <TextBlock Text="Deliverables" FontSize="17" Margin="3,0,0,0"
                            VerticalAlignment="Center"
                            Foreground="Gray" Opacity="0.8" />
                    </StackPanel>
                </Border>
            </sdk:TabItem.Header>
        </sdk:TabItem>
    </sdk:TabControl>
</Grid>

```

```

        </sdk:TabItem.Header>

        </sdk:TabItem>

    </sdk:TabControl>

</Grid>

</common:DataForm>

```

E3_UI-cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using New_IsoTrackWebApp.Common;
using IsotrackWebApp.Library.Messages;

namespace New_IsoTrackWebApp.tb1E1.tb1E1_UI
{
    public partial class E3_UI : DataForm
    {
        long ProjectID;
        public E3_UI(long ProjectID)
        {
            InitializeComponent();
            this.ProjectID = ProjectID;
            load_UI();
            tabcontrol.SelectionChanged += new
                SelectionChangedEventHandler(tabcontrol_SelectionChanged);
        }
        bool skip_tab_changed = false;
        void tabcontrol_SelectionChanged(object sender, SelectionChangedEventArgs e)
        {
            if (skip_tab_changed)
                return;
            TabItem tb_prev = (TabItem)(e.RemovedItems[0]);
            DataForm f = (DataForm)tb_prev.Content;
            if (f.Has_Changes)
            {
                QuestionWindow.AskQuestion("All pending changes will be lost. Continue?", p
                =>
                {
                    switch (p)
                    {
                        case true:
                            f.CancelChanges();
                            break;
                        case false:

```

```

        skip_tab_changed = true;
        tabcontrol.SelectedItem = tb_prev;
        skip_tab_changed = false;

        break;
    }
    });
}

public override void PostData(Data.OnDataPostedHandler p)
{
    ((DataForm)((TabItem)tabcontrol.SelectedItem).Content).PostData(p);
}

public override void CancelChanges()
{
    DataForm f = (tabcontrol.SelectedItem as TabItem).Content as DataForm;
    f.CancelChanges();
}

protected override bool Internal_Has_Changes()
{
    return ((DataForm)((TabItem)tabcontrol.SelectedItem).Content).Has_Changes;
}

void load_UI()
{
    tab_fin.Content = new E3_Fin_UI(ProjectID);
    tab_tech.Content = new E3_Tech_UI(ProjectID);
}
}
}

```

[E4_task_res_UI.xaml](#)

```

<common:DataForm
    x:Class="New_IsoTrackWebApp.tblE1.tblE1_UI.E4_task_res_UI"
    xmlns:sdk="http://schemas.microsoft.com/winfx/2006/xaml/presentation/sdk"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:common="clr-namespace:New_IsoTrackWebApp.Common"
    xmlns:telerik="http://schemas.telerik.com/2008/xaml/presentation"

    mc:Ignorable="d"
    d:DesignHeight="300" d:DesignWidth="800">

    <Grid x:Name="LayoutRoot" Background="White">

        <Grid.RowDefinitions>
            <RowDefinition />
            <RowDefinition Height="50" />
        </Grid.RowDefinitions>

```

```

<Border Grid.Row="0" Background="Gray" Margin="0,0,0,0">
    <sdk:DataGrid x:Name="datagrid_tasks"
        Background="Transparent" AutoGenerateColumns="False"
        VerticalScrollBarVisibility="Auto"
        HorizontalScrollBarVisibility="Hidden"
        FontWeight="Medium" FontSize="17" >
        <sdk:DataGrid.RowGroupHeaderStyles>
            <!-- Style for groups at top level -->
            <Style TargetType="sdk:DataGridRowGroupHeader">
                <Setter Property="PropertyNameVisibility" Value="Collapsed" />
                <Setter Property="Background" Value="#FF112255" />
                <Setter Property="Foreground" Value="#FFEEEEEE" />
                <Setter Property="SublevelIndent" Value="15" />
                <Setter Property="Height" Value="30" />
            </Style>
            <!-- Style for groups under the top level -->
            <Style TargetType="sdk:DataGridRowGroupHeader">
                <Setter Property="Background" Value="#44225566" />
                <Setter Property="Height" Value="30" />
            </Style>
        </sdk:DataGrid.RowGroupHeaderStyles>
        <sdk:DataGrid.Columns>
            <sdk:DataGridTextColumn x:Name="fldProjectTaskID" Binding="{Binding
                Path=fldProjectTaskID, Mode=TwoWay}"
                Header="Task description"
                Width="300" IsReadOnly="true" />
            <sdk:DataGridTextColumn x:Name="fldUserID" Binding="{Binding
                Path=fldUserID, Mode=TwoWay}"
                Header="User name"
                Width="300" IsReadOnly="True" />
            <sdk:DataGridTemplateColumn x:Name="fldApasxolisi" Header="Activity"
                Width="250">
                <sdk:DataGridTemplateColumn.CellEditingTemplate>
                    <DataTemplate>
                        <telerik:RadNumericUpDown x:Name="numeric_up_down"
                            Value="{ Binding
                                Path=fldApasxolisi,
                                Mode=TwoWay,
                                NotifyOnValidationError=true,
                                ValidatesOnExceptions=true,
                                TargetNullValue=''}"/>
                    </DataTemplate>
                </sdk:DataGridTemplateColumn.CellEditingTemplate>
                <sdk:DataGridTemplateColumn.CellTemplate>
                    <DataTemplate>
                        <TextBlock Text="{Binding Path=fldApasxolisi,
                            StringFormat=\{0:n2\}}" HorizontalAlignment="Right"/>
                    </DataTemplate>
                </sdk:DataGridTemplateColumn.CellTemplate>
            </sdk:DataGridTemplateColumn>
        </sdk:DataGrid.Columns>
    </sdk:DataGrid>
</Border>

```

```

        </sdk:DataGrid.Columns>
    </sdk:DataGrid>

</Border>

<StackPanel Grid.Row="1" Margin="0,5,0,0" Orientation="Horizontal"
            VerticalAlignment="Top" >
    <Button x:Name="btnAdd_fin" Click="btnAdd_fin_Click" Content="Add"
            HorizontalAlignment="Left" Width="80" Margin="0,0,0,0" />
    <Button x:Name="btnDelete_fin" Click="btnDelete_fin_Click" Content="Delete"
            HorizontalAlignment="Left" Width="80" Margin="20,0,0,0" />
</StackPanel>

</Grid>
</common:DataForm>

```

[E4_task_res_UI.cs](#)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using New_IsoTrackWebApp.Common;
using New_IsoTrackWebApp.Data;
using System.Windows.Data;
using New_IsoTrackWebApp.Repository;
using System.Collections.ObjectModel;
using System.ServiceModel.DomainServices.Client;
using IsoTrackWebApp.Library.Messages;

namespace New_IsoTrackWebApp.tblE1.tblE1_UI
{
    public partial class E4_task_res_UI : DataForm
    {
        long ProjectID;
        public E4_task_res_UI(long ProjectID)
        {
            InitializeComponent();
            this.ProjectID = ProjectID;
            load();
        }

        void load()
    }
}

```

```

{
    DataGridTextColumn user_col =
        (DataGridTextColumn)FindColumn.GetByName(datagrid_tasks.Columns, "fldUserID");
    user_col.Binding.Converter = new User_converter();

    DataGridTextColumn task_col =
        (DataGridTextColumn)FindColumn.GetByName(datagrid_tasks.Columns, "fldProjectTaskID");
    task_col.Visibility = System.Windows.Visibility.Collapsed;

    Data_Source.LoadData("GetTblE4TasksResources_of_project",
        new List<Parameter> { new Parameter { ParameterName = "ProjectID", Value =
                                                                    ProjectID }

        }, p =>
    {
        PagedCollectionView pg_view = new PagedCollectionView(Data_Source.Items);

        PropertyGroupDescription prg = new
            PropertyGroupDescription("fldProjectTaskID");

        prg.Converter = new task_converter();

        pg_view.GroupDescriptions.Add(prg);

        datagrid_tasks.ItemsSource = pg_view;

        btnAdd_fin.IsEnabled = btnDelete_fin.IsEnabled =
            ApplicationContext.Team_Role == New_IsoTrackWebApp.Data.TeamRole.ProjectManager;

        datagrid_tasks.IsReadOnly = !btnAdd_fin.IsEnabled;
    });
}

public List<Web.tblE4TasksResources> Data
{
    get
    {
        return Data_Source.Items.Cast<Web.tblE4TasksResources>().ToList();
    }
}

static class FindColumn
{
    public static DataGridColumn GetByName(ObservableCollection<DataGridColumn> col,
        string name)
    {
        return col.SingleOrDefault(p =>
            (string)p.GetValue(FrameworkElement.NameProperty) == name
        );
    }
}

class User_converter : IValueConverter
{
    public object Convert(object value, Type targetType, object parameter,
        System.Globalization.CultureInfo culture)
    {
        return DataRepository.Look_Up_Value<Web.tblUsers>("Users", "fldUserID",

```



```

        System.Convert.ToInt64(value), "flduserFullName");
    }

    public object ConvertBack(object value, Type targetType, object parameter,
        System.Globalization.CultureInfo culture)
    {
        return null;
    }
}

class task_converter : IValueConverter
{
    public task_converter()
    {
    }
    public object Convert(object value, Type targetType, object parameter,
        System.Globalization.CultureInfo culture)
    {
        return DataRepository.Look_Up_Value<Web.tblE4Tasks>("Tasks",
            "fldProjectTaskID", System.Convert.ToInt64(value), "fldProjectTaskDescription");
    }

    public object ConvertBack(object value, Type targetType, object parameter,
        System.Globalization.CultureInfo culture)
    {
        return null;
    }
}

protected override Data.DataSource Get_DataSource()
{
    return new DataSource(new New_IsoTrackWebApp.Web.E1.tblE1_Context());
}

private void btnAdd_fin_Click(object sender, RoutedEventArgs e)
{
    Select_Task_Res_UI sel = new Select_Task_Res_UI(ProjectID, new ChildWindow(),
false );
    sel.Show(p =>
    {
        List<object> list = (List<object>)p;
        Web.tblUsers usr = (Web.tblUsers)list[0];
        Web.tblE4Tasks tsk = (Web.tblE4Tasks)list[1];

        if (Data_Source.Items.Cast<Web.tblE4TasksResources>().FirstOrDefault(res =>
(res.fldUserID == usr.fldUserID) && (res.fldProjectTaskID == tsk.fldProjectTaskID)) != null)
        {
            ErrorWindow.ShowError(string.Format("User {0} assigned to task {1}
already exists", usr.flduserFullName, tsk.fldProjectTaskDescription));
        }
        else
        {
            Web.tblE4TasksResources tsk_r = new Web.tblE4TasksResources();
            tsk_r.fldProjectID = ProjectID;
            tsk_r.fldProjectTaskID = tsk.fldProjectTaskID;
            tsk_r.fldUserID = usr.fldUserID;
            tsk_r.fldApasxolisi = 0;
            Data_Source.Items.Add(tsk_r);
        }
    }
}

```

```

        });
    }

    private void btnDelete_fin_Click(object sender, RoutedEventArgs e)
    {
        Web.tblE4TasksResources res = datagrid_tasks.SelectedItem as
                                                Web.tblE4TasksResources;

        if (res == null)
        {
            return;
        }
        Data_Source.Items.Remove(res);
    }
}
}

```

[tblE1_Services.cs](#)

```

namespace New_IsoTrackWebApp.Web.E1
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel;
    using System.ComponentModel.DataAnnotations;
    using System.Data;
    using System.Linq;
    using System.ServiceModel.DomainServices.EntityFramework;
    using System.ServiceModel.DomainServices.Hosting;
    using System.ServiceModel.DomainServices.Server;
    using New_IsoTrackWebApp.Web;
    using System.Data.SqlClient;
    using System.Configuration;
    using System.Data.EntityClient;
}

```

```

// Implements application logic using the IsotrackEntities context.
// TODO: Add your application logic to these methods or in additional methods.
// TODO: Wire up authentication (Windows/ASP.NET Forms) and uncomment the following to
disable anonymous access
// Also consider adding roles to restrict access as appropriate.
// [RequiresAuthentication]
[EnableClientAccess()]
public class tblE1_Service : LinqToEntitiesDomainService<IsotrackEntities>
{
    #region my services

    #region Get_TblE1_By_ID

    [Invoke]
    public bool Insert_New_TblE1(tblE1 new_e1, long UserID)
    {
        InsertTblE1(new_e1);
        this.ObjectContext.SaveChanges();
        tblE1 e1 = ObjectContext.tblE1.FirstOrDefault(t => t.fldProjectID ==
            ObjectContext.tblE1.Max(p => p.fldProjectID));
        add_new_resource(e1, UserID);
        return true;
    }

    void add_new_resource(tblE1 e1, long UserID)
    {
        Web.tblE2Resources rs = new tblE2Resources();
        rs.fldProjectID = e1.fldProjectID;
        rs.fldUserID = UserID;
        rs.fldUserMandays = 0;
        rs.fldTeamRoleID = 1; // project director
        rs.fldUserProjectTypeID = 1;

        ObjectContext.tblE2Resources.AddObject(rs);

        ObjectContext.SaveChanges();
    }

    public IQueryable<tblE1> Get_TblE1_By_ID(long ProjectID)
    {
        var list = from p in ObjectContext.tblE1
            where p.fldProjectID == ProjectID
            select p;

        return list;
    }

    public IQueryable<tblE1> GetProjectsOfCurrentUser(long UserID)
    {
        var list1 = from p in ObjectContext.viewUserRoles
            where p.fldUserID == UserID
            select p;
        bool full_rights = (list1.Count() > 0) && (list1.ToList().Find(p => p.fldRoleID
            == 1 || p.fldRoleID == 2) != null);

        var list = from p in ObjectContext.tblE1
            where (full_rights ||

```

```

                (from r in ObjectContext.vwE2Resources
                 where r.fldProjectID == p.fldProjectID
                 select r.fldUserID).Contains(UserID))
            select p;

        return list.AsQueryable();
    }

#endregion

#region Get_tblE1_By_UserID

public IQueryable<tblE1> Get_tblE1_By_UserID(long UserID)
{
    var list1 = from p in ObjectContext.viewUserRoles
                where p.fldUserID == UserID
                select p;

    bool full_rights = (list1.Count() > 0) && (list1.ToList().Find(p => p.fldRoleID
        == 1 || p.fldRoleID == 2) != null);

    var list = from p in ObjectContext.tblE1
                where (full_rights ||
                    (from r in ObjectContext.vwE2Resources
                     where r.fldProjectID == p.fldProjectID
                     select r.fldUserID).Contains(UserID))
                select p;

    return list.AsQueryable();
}

#endregion

#region Get_recipients_of_project

public IQueryable<tblE1ProjectRecipient_Link> Get_recipients_of_project(long
ProjectID)
{
    var list = from p in this.ObjectContext.tblE1ProjectRecipient_Link
                where p.fldProjectID == ProjectID
                select p;

    return list;
}

#endregion

#region Get_recipients

public IQueryable<tblRecipients> Get_recipients()
{
    var list = from p in this.ObjectContext.tblRecipients
                select p;

    return list;
}

```

```

#endregion

#region GetTblE1ProjectParticipant_Link_of_project

public IQueryable<tblE1ProjectParticipant_Link>
    GetTblE1ProjectParticipant_Link_of_project(long ProjectID)
{
    var list = from p in this.ObjectContext.tblE1ProjectParticipant_Link
                where p.fldProjectID == ProjectID
                select p;
    return list;
}

#endregion

public IQueryable<tblUserProjectType> get_project_types()
{
    return (from p in ObjectContext.tblUserProjectType select p);
}

#region Get_Participants

public IQueryable<tblParticipants> Get_Participants()
{
    return (from p in ObjectContext.tblParticipants
            select p);
}

public IQueryable<tblE3FinancialDeliverables> GetFinancials(long ProjectID)
{
    var list = from p in this.ObjectContext.tblE3FinancialDeliverables
                where p.fldProjectID == ProjectID
                select p;
    return list;
}

public IQueryable<tblE3TechnicalDeliverables> GetTechnicals(long ProjectID)
{
    var list = from p in this.ObjectContext.tblE3TechnicalDeliverables
                where p.fldProjectID == ProjectID
                select p;
    return list;
}

public IQueryable<tblE43Years> Get43Years_of_project(long ProjectID)
{
    var list = from p in this.ObjectContext.tblE43Years
                where p.fldProjectID == ProjectID
                select p;
    return list;
}

#endregion

#region Get_TblE2_By_ID

```

```

#endregion

public IQueryable<tblE2> Get_TblE2_By_ID(long ProjectID)
{
    var list = from p in ObjectContext.tblE2
                where p.fldProjectID == ProjectID
                select p;

    return list;
}

#region GetTblE5_for_project

public IQueryable<tblE5> GetTblE5_for_project(long ProjectID)
{
    var list = ObjectContext.tblE5.Include("tblE5Deliverables");
    return (from p in list
            where p.fldProjectID == ProjectID
            select p);
}

#endregion

#region

private string get_conn_string(string entityConnectionString)
{
    // create a entity connection string from the input
    EntityConnectionStringBuilder entityBuilder = new
        EntityConnectionStringBuilder(entityConnectionString);

    // read the db connectionstring
    return entityBuilder.ProviderConnectionString;
}

[Invoke]
public bool DeleteByID(long ID)
{
    string error = null;
    SqlConnection conn = new SqlConnection("Data Source=Eobey-PC\\Sql_Server;Initial
        Catalog=Isotrack;Integrated Security=True;MultipleActiveResultSets=True");
    conn.Open();
    try
    {
        {
            SqlTransaction trans = conn.BeginTransaction();
            try
            {
                SqlCommand cmd = conn.CreateCommand();
                cmd.Transaction = trans;
                delete(cmd, "tblE1ProjectParticipant_Link", ID);
                delete(cmd, "tblE3TechnicalDeliverables", ID);
                delete(cmd, "tblE3FinancialDeliverables", ID);
                delete(cmd, "tblE5Deliverables", ID);
                delete(cmd, "tblE4TasksResources", ID);
                delete(cmd, "tblE4Tasks", ID);
                delete(cmd, "tblE43Years", ID);
                delete(cmd, "tblE5", ID);
            }
        }
    }
}

```

```

        delete(cmd, "tblE2", ID);
        delete(cmd, "tblE1", ID);
        trans.Commit();
    }
    catch (Exception ex)
    {
        error = ex.Message;
        trans.Rollback();
    }
    finally
    {
    }
}
finally
{
    conn.Close();
}

return true;
}

[Invoke]
public int Get_TeamRole(long ProjectID, long UserID)
{
    var list =ObjectContext.tblE2Resources.Where(r => r.fldProjectID == ProjectID
        && r.fldUserID == UserID);
    if (list.FirstOrDefault() == null)
    {
        return -1;
    }
    return Convert.ToInt32(list.FirstOrDefault().fldTeamRoleID);
}

void delete(SqlCommand cmd, string tablename, long master_id)
{
    cmd.CommandText = string.Format("DELETE {0} WHERE fldProjectID = {1}",
        tablename, master_id);
    cmd.ExecuteNonQuery();
}

#endregion

#endregion
}

```