

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Τμήμα Μηχανικών Πληροφορικής



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Τίτλος πτυχιακής: Πληροφοριακό Σύστημα για εταιρεία Κινητής Τηλεφωνίας

Σπουδαστής

ΖΑΚΑ ΙΩΑΝΝΗΣ (ΑΜ: ΤΠ4129)

Επιβλέπων εκπαιδευτικός: Παπαδάκης Νικόλαος

Ημερομηνία παρουσίασης: Μάιος, 2020

Ευχαριστίες

Πρώτα από όλα θέλω να ευχαριστήσω τον κύριο Παπαδάκη Νικόλαο για τον χρόνο που πρόσφερε βοηθώντας για την εκπόνηση της πτυχιακής αυτής εργασίας όπως επίσης και την πλήρης αλλά και ταυτόχρονα σωστή καθοδήγηση.

Abstract

This application was designed and produced for its usage by any Telecommunication company. Its stability, flexibility and ease of use makes it a competitive product in the market. Its system capabilities allows a company to keep their branches details including each branch's employees and administrative staff. Every employee is either a technician or an administrative. Each employee makes a contract with a customer. Every customer can have one telephone contract but multiple telephone numbers. A bill is then produced dynamically based on the last bills issue date and the cost depend on each contract. Furthermore, this application is not only capable of keeping phone numbers, customer and contract details, but also antennas details. It provides for each branch unlimited number of antennas with its details (type, location etc.) but every antenna can only exist in one branch. Details of the antenna's maintenance are also included with this system. Last but not least, it provides really efficient data visualization by showing real time data reporting that can dynamically change by filters given to the user.

Σύνοψη

Η συγκεκριμένη εφαρμογή σχεδιάστηκε και κατασκευάστηκε για την χρήση της από οποιαδήποτε εταιρεία τηλεπικοινωνιών. Η σταθερότητα, η ευελιξία και η ευκολία χρήσης του, το καθιστούν ως ένα ανταγωνιστικό προϊόν στην αγορά. Οι δυνατότητες του συστήματος επιτρέπουν σε μια εταιρεία να διατηρεί τα στοιχεία των καταστημάτων της, συμπεριλαμβανομένων των υπαλλήλων και διοικητικών υπαλλήλων κάθε υποκαταστήματος. Κάθε εργαζόμενος είναι είτε τεχνικός είτε διοικητικός. Κάθε εργαζόμενος συνάπτει σύμβαση με έναν κάθε φορά πελάτη. Ο πελάτης έχει την δυνατότητα να έχει μόνο μια σύμβαση, αλλά περισσότερους από έναν αριθμό τηλεφώνου. Στη συνέχεια, ένας λογαριασμός παράγεται δυναμικά βάσει της ημερομηνίας έκδοσης των τελευταίων λογαριασμών και το κόστος εξαρτάται από κάθε σύμβαση. Επιπλέον, η εφαρμογή αυτή δεν είναι μόνο ικανή να διατηρεί τηλεφωνικούς αριθμούς, στοιχεία πελατών και συμβάσεων, αλλά και λεπτομέρειες κεραιών. Παρέχει για κάθε υποκατάστημα απεριόριστο αριθμό κεραιών με τις λεπτομέρειες που τους ακολουθούν (τύπος, θέση κλπ.) ,όμως κάθε κεραία μπορεί να υπάρχει μόνο σε ένα υποκατάστημα. Λεπτομέρειες σχετικά με τη συντήρηση της κεραιάς περιλαμβάνονται επίσης σε αυτό το σύστημα. Τελευταίο αλλά εξίσου σημαντικό, είναι ότι παρέχει πραγματικά αποτελεσματική απεικόνιση δεδομένων, δείχνοντας αναφορές σε πραγματικό χρόνο, οι οποίες μπορούν να αλλάξουν δυναμικά με φίλτρα που δίνονται στον χρήστη.

Contents

1. Εισαγωγή.....	7
1.1 Περίληψη	7
1.2 Στόχοι και κίνητρα για την Διεξαγωγή της Εργασίας	7
1.3 Δομή Εργασίας.....	8
2. Μεθοδολογία Υλοποίησης.....	9
2.1 Μέθοδος Ανάπτυξης & Ανάλυσης	9
2.1.1 Oracle Database	9
2.1.2 Oracle Application Express.....	10
2.1.3 Apache Tomcat.....	11
2.1.4 ORDS (Oracle Rest Data Services)	11
2.1.5 Oracle SQL Developer.....	12
2.1.6 Docker	13
2.1.7 HTML	13
2.1.8 Java script.....	14
2.1.9 jQuery.....	15
2.1.10 CSS.....	16
2.1.11 PL/SQL.....	17
3. Στόχοι και ενέργειες για την εκπόνηση της Πτυχιακής Εργασίας	18
4. Κύριο Μέρος της Πτυχιακής Εργασίας.....	19
4.1 Ανάλυση του Προβλήματος.....	19
4.1.2 Απαιτήσεις του Συστήματος.....	19
4.2 Σχεδιασμός Προβλήματος & Υλοποίηση	19
5. Αποτέλεσμα	49
5.1 Συμπέρασμα	49
5.2 Επεκτασιμότητα και εργασία για το μέλλον	51
Πίνακας των εικόνων	52
Βιβλιογραφία	566

1. Εισαγωγή

1.1 Περίληψη

Η εφαρμογή παρέχει υπηρεσίες διαχείρισης δεδομένων σε εταιρείες τηλεπικοινωνιών. Αρχικά είναι έτσι στημένη ώστε να είναι όσο πιο δυναμική γίνεται και να διευκολύνει τους χρήστες της. Η δομή του επιτρέπει την ομαλή διαχείριση των υποκαταστημάτων μια εταιρείας με ότι αυτό ακολουθεί όπως για παράδειγμα τους εργαζόμενους, τους πελάτες αλλά ακόμα και τις κεραίες που διαθέτει το κάθε κατάστημα. Τέλος διατηρεί ολόκληρο ιστορικό από όλα αυτά τα δεδομένα στο οποίο με την εικόνα αυτή θα μπορεί η διοίκηση να γνωρίζει τι ενέργειες θα πρέπει να ακολουθηθούν για την πιο σωστή και ομαλή λειτουργία της εταιρείας.

1.2 Στόχοι και κίνητρα για διεξαγωγή εργασίας

Πλέον μεγάλο ποσοστό των επιχειρήσεων στρίβει προς μία νέα κατεύθυνση και σε νέο πιο «μοντέρνο» μοντέλο λειτουργίας. Φυσικά και αναφέρω για την ψηφιοποίηση όλων των δεδομένων που τόσο καιρό λειτουργούσαν με στείβες από κόλλες Α4 στοιβαγμένα μέσα σε ντοσιέ. Ενίοτε κόλλες χανόντουσαν ή με την πάροδο του χρόνου αλλοιώνονταν έχοντας ως αποτέλεσμα την απώλεια σημαντικών δεδομένων. Επίσης οι ατελείωτες ώρες συμπλήρωσης φορμών με το χέρι αλλά και η διατήρηση της οργανωμένης και ασφαλής τοποθέτηση τους στα σωστά σημεία εξαντλούν τους υπαλλήλους κατά μεγάλο ποσοστό. Χάνοντας έτσι παραγωγικότητα, χρόνο και χρήματα. Αποφασίζοντας την ένταξη της ψηφιοποίησης στην επιχείρηση της μία εταιρεία θα έχει μόνο θετικά αποτελέσματα. Μερικά από αυτά είναι τα εξής:

- Βελτιωμένη ασφάλεια δεδομένων
- Βελτιωμένη παραγωγικότητα των εργαζομένων
- Μείωση κόστους
- Μειωμένα χειροκίνητα λάθη κατά την εισαγωγή δεδομένων
- Μειωμένος χρόνος που χρειάζεται για την συλλογή των δεδομένων
- Βιωσιμότητα κατά την μειωμένη χρήση του χαρτιού
- Λιγότερος αποθηκευτικός χώρος εφόσον δεν υπάρχει πλέον το χαρτί

1,000 SMBs Top 5 Business Priorities



IMGS-1

Why Businesses Need to Go Digital



IMGS-2

1.3 Δομή της εργασίας

Εν συνέχεια, στο δεύτερο κεφάλαιο θα αναλύσουμε την μεθοδολογία της υλοποίησης της εφαρμογής. Με λίγα λόγια τον τρόπο ο οποίος αναλύσαμε και αναπτύξαμε την εφαρμογή. Έπειτα στο τρίτο κεφάλαιο θα αναπτύξουμε τους στόχους και τις ενέργειες οι οποίες είναι απαραίτητη για την ολοκλήρωση της εφαρμογής. Μετά στο τέταρτο κεφάλαιο θα αναλύσουμε το κύριο μέρος της εργασίας βρίσκοντας τα προβλήματα και τις απαιτήσεις για το σύστημα μας όπως επίσης και να προγραμματίσουμε την υλοποίησης και τον τρόπο με τον οποίο εν τέλη ολοκληρώσαμε. Στο τέλος, το πέμπτο κεφάλαιο θα είναι αυτό που αναλύει τα αποτελέσματα και το συμπέρασμα της εργασίας, όπως και τους τρόπους οι οποίοι θα μπορούσαν να αναπτύξουν, να εξελίξουν και να επεκτείνουν την εφαρμογή στο μέλλον.

2. Μεθοδολογία της υλοποίησης

2.1 Μέθοδος Ανάπτυξης & Ανάλυσης

Για την υλοποίηση Της εφαρμογής χρησιμοποιήσα έναν συνδυασμό εργαλείων. Τα εργαλεία αυτά είναι τα εξής:

- Oracle Database 19c
- Oracle Application Express 19.2
- Apache Tomcat
- ORDS (Oracle Rest Data Services)
- Oracle SQL Developer
- Docker Desktop
- HTML5
- JavaScript
- jQuery
- CSS
- PL/SQL

2.1.1 Oracle Database



IMGS-3

Η Oracle Database (ή αλλιώς γνωστή και σας Oracle RDBMS ή και μόνο Oracle) είναι ένα σύστημα διαχείρισης σχεσιακής βάσης δεδομένων οπού αρχικά αναπτύχθηκε από τον Lawrence Ellison και άλλους προγραμματιστές στην Oracle Corporation το 1977. Πλέον είναι

μία από τις πιο αξιόπιστες και ευρέως χρησιμοποιούμενες μηχανές σε σχεσιακές βάσεις δεδομένων.

Το σύστημα αυτό είναι χτισμένο γύρω από ένα πλαίσιο σχεσιακής βάσης δεδομένων του οποίου δίδεται η πρόσβαση στα αντικείμενα του και στα δεδομένα του μέσω δομημένης γλώσσας ερωτήματος (SQL) στους χρήστες και στις front end εφαρμογές. Η Oracle αποτελείται από μια πλήρως κλιμακούμενη αρχιτεκτονική των σχεσιακών βάσεων οι οποίες χρησιμοποιούνται συχνά από επιχειρήσεις παγκοσμίως οι οποίες διαχειρίζονται και επεξεργάζονται τα δεδομένα τους σε ευρεία αλλά και τοπικά δίκτυα. Η Oracle έχει αναπτύξει την δικιά της δομή δικτύου για να επιτρέπει επικοινωνίες μεταξύ των δικτύων και αυτό την καθιστά ως μια απ' τις ασφαλέστερες και σταθερές βάσεις δεδομένων στην αγορά.

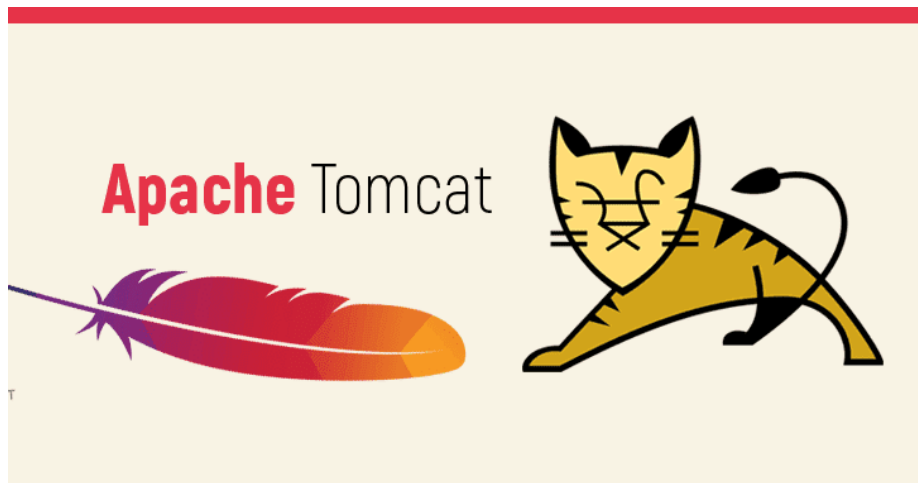
2.1.2 Oracle Application Express



IMGS-4

Το APEX , ένα εργαλείο ανάπτυξης ιστότοπων(εφαρμογών) χαμηλού επιπέδου κώδικα για τις βάσεις δεδομένων σε Oracle. Το Application Express σας δίνει την δυνατότητα σχεδιασμού, ανάπτυξης και παράτασης των όμορφων και αποκριτικών εφαρμογών που βασίζονται σε βάσεις δεδομένων στο Cloud ή σε τοπικές εγκαταστάσεις. Χρησιμοποιώντας μόνο ένα πρόγραμμα περιήγησης μπορείτε γρήγορα να αναπτύξετε επαγγελματικές εφαρμογές που είναι γρήγορες και ασφαλείς για οποιαδήποτε συσκευή, από Η/Υ έως και κινητά. Συνδυάζει τις ιδιότητες ενός εργαλείου χαμηλού κώδικα, παραγωγικότητας, ευκολίας στην χρήση και ευελιξίας με τις ιδιότητες ενός εργαλείου ανάπτυξης επιχειρησιακών εφαρμογών με ασφάλεια, ακεραιότητα αλλά και επεκτασιμότητα για τον ιστό.

2.1.3 Apache Tomcat



IMGS-5

Το Apache Tomcat είναι ένα λογισμικό το οποίο είναι εφαρμογή ανοιχτού κώδικα και αποτελείται από τεχνολογίες όπως Java Servlet, JavaServer Pages, Java Expression Language και Java WebSocket. Οι τεχνολογίες αυτές οι οποίες αποτελείται αναπτύσσονται όλες στο πλαίσιο κοινοτικής διαδικασίας της Java. Το λογισμικό αυτό αναπτύσσεται σε ένα ανοιχτό και συμμετοχικό περιβάλλον και η κυκλοφορία του γίνεται υπό της έκδοση της Apache License 2. Ο Tomcat κυκλοφόρησε για πρώτη φορά το 1998, τέσσερα χρόνια μετά την Java. Ξεκίνησε ως εφαρμογή αναφοράς για το πρώτο Java Servlet API και το JSP spec. Παρόλο που πλέον δεν είναι η εφαρμογή του οι υλοποιήσεις αναφοράς για αυτές τις τεχνολογίες, παραμένει ο πιο ευρέως χρησιμοποιούμενος εξυπηρετητής Java διαθέτοντας μια δοκιμασμένη βασική μηχανή με καλή εκτατότητα.

2.1.4 ORDS (Oracle Rest Data Services)



IMGS-6

Το ORDS επιτρέπει την καταγραφή ή την έκθεση των δεδομένων σε μία βάση δεδομένων της Oracle (σχεσιακή ή NoSQL) χρησιμοποιώντας το πανταχού παρόν πρωτόκολλο REST. Παρέχεται ως εφαρμογή Java (.war αρχείο) που μπορεί να εκτελεστεί αυτόνομα ή μέσα σε ένα container J2EE (όπως Apache Tomcat, WebLogic). Η χρήση του γίνεται με συνδυασμό της υπηρεσίας web APEX5 και την ανάλυση JSON PL/SQL API, έτσι αλλάζοντας την λειτουργικότητα της βάσης δεδομένων σε έναν πλήρη λειτουργικό κόμβο ενσωμάτωσης. Άλλα πλεονεκτήματα της χρήσης του ORDS είναι ο ενσωματωμένος έλεγχος ταυτότητας OAUTH2, η αυτόματη ανάλυση SQL σε JSON, ο αυτόματος χειρισμός για το HATEOAS όπως επίσης κι άλλα καλούδια τα οποία δεν θα αναφέρω.

2.1.5 SQL Developer



IMGS-7

Το SQL Developer είναι ένα δωρεάν, ολοκληρωμένο περιβάλλον ανάπτυξης το οποίο απλοποιεί την διαχείριση και την ανάπτυξη μιας Oracle βάσης δεδομένων. Λειτουργεί και σε παραδοσιακές βάσεις αλλά και σε βάσεις οι οποίες βρίσκονται στο Cloud. Το περιβάλλον αυτό προσφέρει την ολοκληρωμένη ανάπτυξη εφαρμογών PL/SQL ως ένα φύλλο εργασίας για την εκτέλεση ερωτημάτων (SQL queries) και σεναρίων. Επίσης προσφέρει μια κονσόλα DBA για την διαχείριση της βάσης δεδομένων, όπως επίσης περιλαμβάνει και μια ολοκληρωμένη λύση μοντελοποίησης δεδομένων μαζί και μια πλατφόρμα μετάβασης για την διακίνηση των δεδομένων από διαφορετικούς τύπους βάσεων εκτός Oracle (π.χ. MySQL, PostgreSQL κλπ.).

2.1.6 Docker



IMGS-8

Το Docker είναι ένα εργαλείο σχεδιασμένο για την διευκόλυνση της δημιουργίας, ανάπτυξης και εκτέλεσης εφαρμογών χρησιμοποιώντας δοχεία (containers). Τα δοχεία (containers) επιτρέπουν σε έναν προγραμματιστή να συσκευάσει μια εφαρμογή με όλα τα μέρη που χρειάζεται, όπως βιβλιοθήκες και άλλες εξαρτήσεις όπως και την ανάπτυξη ως ένα πακέτο. Με τον τρόπο αυτό, χάρη στο δοχείο (container), ο προγραμματιστής μπορεί να είναι βέβαιος ότι η εφαρμογή θα τρέξει σε οποιοδήποτε άλλο μηχάνημα Linux, ανεξάρτητα από τυχόν προσαρμοσμένες ρυθμίσεις που μπορεί να έχει εκείνο το μηχάνημα, κάτι που μπορεί να διαφέρει από το μηχάνημα που χρησιμοποιείται για τη γραφή και τον έλεγχο του κώδικα. Με λίγα λόγια, το Docker είναι λίγο σαν μια εικονική μηχανή. Αντίθετα από μια εικονική μηχανή (π.χ. Virtual Machine), αντί να δημιουργήσει ένα ολόκληρο εικονικό λειτουργικό σύστημα, το Docker επιτρέπει στις εφαρμογές να χρησιμοποιούν τον ίδιο πυρήνα Linux με το σύστημα στο οποίο εκτελούν και απαιτεί μόνο να στέλνονται εφαρμογές με πράγματα που δεν εκτελούνται ήδη από τον υπολογιστή τον κεντρικό. Αυτό έχει ως αποτέλεσμα την σημαντική ενίσχυση της απόδοσης και ταυτόχρονα μειώνει και το μέγεθος της εφαρμογής.

2.1.7 HTML



IMGS-9

Το HTML του οποίου το όνομα σημαίνει Hypertext Markup Language, επιτρέπει στον χρήστη να δημιουργεί και να δομεί τμήματα, επικεφαλίδες, παραγράφους συνδέσμους αλλά και blockquotes για εφαρμογές και ιστοσελίδες. Το HTML δεν θεωρείται μια γλώσσα προγραμματισμού ως αποτέλεσμα να μην έχει τη δυνατότητα δημιουργίας δυναμικών λειτουργιών. Αντιθέτως, καθιστά δυνατή την οργάνωση και τη μορφοποίηση εγγράφων, όπως και το Microsoft Word. Χρησιμοποιώντας HTML, χρησιμοποιούμε απλές δομές κώδικα (χαρακτηριστικά και ετικέτες) για την επισήμανση μιας σελίδας ιστότοπου. Γενικά, η HTML είναι μια γλώσσα σήμανσης η οποία είναι πραγματικά απλή και εύκολη στην εκμάθηση ακόμα και για πιο αρχάριους που ασχολούνται με την σχεδίαση ιστοτόπων.

2.1.8 Java script



IMGS-10

Η JavaScript είναι μια γλώσσα δέσμης ενεργειών για τον ιστό. Θεωρείτε μία ερμηνευτική γλώσσα, που αυτό σημαίνει ότι δεν έχει την ανάγκη μεταγλωττιστή για να μεταφράσει τον κώδικα της όπως για παράδειγμα C ή C ++. Ο κώδικας εκτελείται απευθείας από ένα πρόγραμμα περιήγησης ιστού (π.χ. Firefox, Chrome κ.λπ.). Η τελευταία έκδοση που κυκλοφόρησε τον Ιούνιο του 2018 είναι η ECMAScript 2018. Λειτουργεί μαζί με HTML και CSS για την δημιουργία εφαρμογών ιστού ή ιστοσελίδες. Υποστηρίζεται από τα περισσότερα σύγχρονα προγράμματα περιήγησης ιστού υπολογιστών αλλά και κινητών. Επίσης ελέγχει τα δυναμικά στοιχεία των ιστοσελίδων. Τα Application Programming Interfaces (API) υποστηρίζονται επίσης δίδοντας έτσι μεγαλύτερο εύρος λειτουργικότητας.

2.1.9 jQuery



IMGS-11

Το jQuery, μια μικρή, γρήγορη και πλούσια σε χαρακτηριστικά βιβλιοθήκη βασισμένη σε JavaScript. Εκτελεί ενέργειες όπως την μετακίνηση των εγγράφων HTML, τον χειρισμό συμβάντων και κινούμενες εικόνες. Επίσης λειτουργεί το Ajax με πολύ απλούστερο τρόπο με ένα εύχρηστο API που λειτουργεί σε μεγάλο πλήθος περιηγητών. Με έναν συνδυασμό ευελιξίας και εκτατοτητας, η βιβλιοθήκη αυτή άλλαξε τον τρόπο που εκατομμύρια προγραμματιστές γράφουν JavaScript.

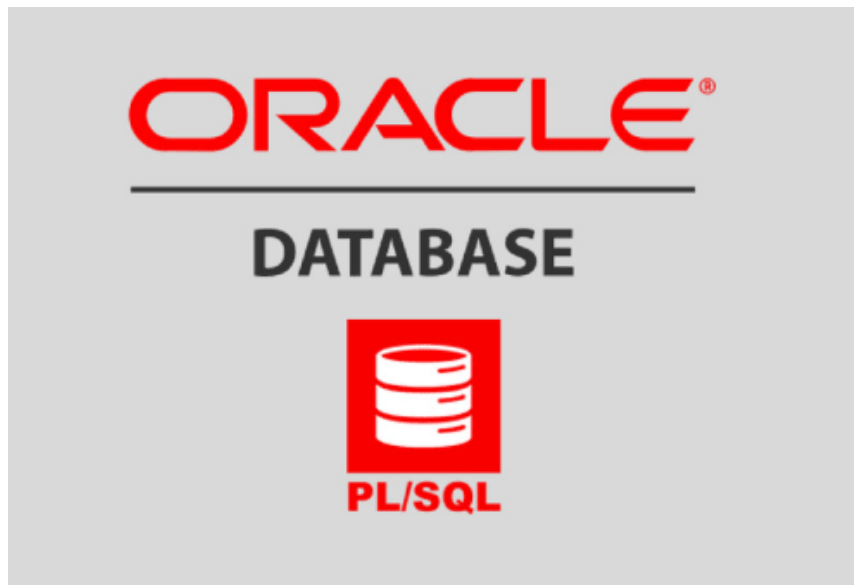
2.1.10 CSS



IMGS-12

Το CSS, μία απλή γλώσσα σχεδίασης που αποσκοπεί στην απλοποίηση της διαδικασίας κατά την δημιουργία ιστοσελίδων. Χειρίζεται την αίσθηση και την όψη μια ιστοσελίδας. Χρησιμοποιώντας το, μπορεί κάποιος να ελέγξει το χρώμα του κειμένου, το στυλ των γραμματοσειρών, την απόσταση μεταξύ παραγράφων, τον τρόπο μεγέθυνσης και διαμόρφωσης των στηλών, ποιες εικόνες φόντου ή χρώματα θα χρησιμοποιηθούν, παραλλαγές εμφάνισης ανάλογα την συσκευή με βάση το μέγεθος της οθόνης καθώς και μια ποικιλία άλλων επιδράσεων. Συνήθως συνδυάζεται με γλώσσες όπως HTML ή XHTML. Είναι εύκολο στην εκμάθηση του και μπορεί κάποιος να το καταλάβει εύκολα αλλά παρέχει ισχυρό έλεγχο στην παρουσίαση ενός εγγράφου HTML.

2.1.11 PL/SQL



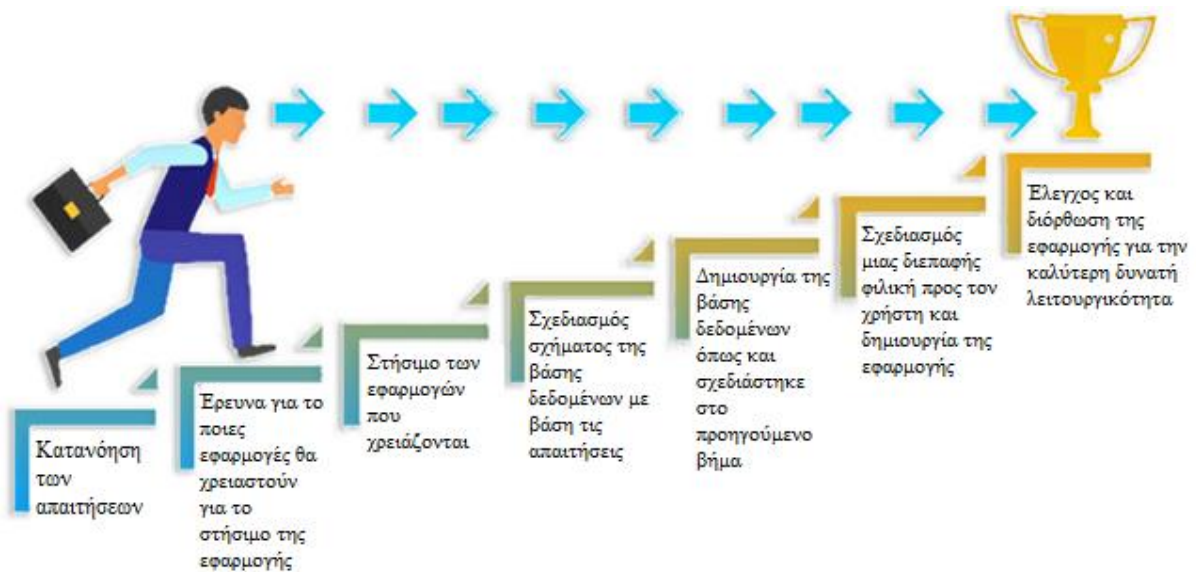
IMGS-13

Το PL/SQL είναι μια επέκταση της Δομημένης Γλώσσας Ερωτήσεων (SQL) που χρησιμοποιείται από την Oracle. Σε αντίθεση με την SQL, το PL/SQL επιτρέπει στον προγραμματιστή να γράφει κώδικα σε μια διαδικαστική μορφή. Η πλήρης ονομασία του είναι “Procedural Language extensions to SQL”. Συνδυάζει τη δύναμη επεξεργασίας της διαδικαστικής γλώσσας με τη δύναμη χειρισμού δεδομένων της SQL για την δημιουργία ισχυρών ερωτημάτων σε SQL. Αυτό σημαίνει ότι έχει την δυνατότητα να καθοδηγεί τον μεταγλωττιστή στο πως «τι πρέπει να κάνει» μέσω της SQL και το «πως να το κάνει» μέσω του διαδικαστικού του τρόπου. Παρόμοιος με άλλες γλώσσες βάσεων δεδομένων, δίνει περισσότερο έλεγχο στους προγραμματιστές με τη χρήση βρόχων, συνθηκών, αντικειμένων αλλά και προσανατολισμένων εννοιών.

3. Στόχοι και ενέργειες για την εκπόνηση της Πτυχιακής Εργασίας

Για το ομαλό αλλά και το σωστό στήσιμο της εφαρμογής θα πρέπει να ακολουθηθούν κάποια συγκεκριμένα βήματα με συγκεκριμένη σειρά. Τα βήματα αυτά και η σειρά τους σας παρουσιάζονται παρακάτω:

- i. Κατανόηση των απαιτήσεων
- ii. Έρευνα των εφαρμογών που θα χρειαστούν για το στήσιμο της εφαρμογής
- iii. Στήσιμο των εφαρμογών
- iv. Σχεδιασμός σχήματος της βάσης δεδομένων
- v. Δημιουργία της βάσης δεδομένων
- vi. Σχεδιασμός μιας διεπαφής φιλική προς τον χρήστη και δημιουργία της
- vii. Έλεγχος και διορθώσεις στην εφαρμογή για την καλύτερη δυνατή λειτουργικότητα



IMGS-14

4. Κύριο Μέρος της Πτυχιακής Εργασίας

4.1 Ανάλυση του Προβλήματος

Προκειμένου να στήσω την εφαρμογή θα πρέπει πρώτα να κατανοήσω τις απαιτήσεις που έχουν ζητηθεί. Επίσης θα πρέπει να δημιουργήσω ένα αρχικό σχήμα της βάσης με τα απαραίτητα tables τα οποία θα διατηρούν τα δεδομένα. Σε ένα πρόγραμμα όπως το xl της Microsoft θα σχεδιάσω τα απαραίτητα tables. Στην συνέχεια σχεδιάσα το κομμάτι της διεπαφής την οποία θα βλέπει και θα χρησιμοποιεί ο χρήστης της εταιρείας. Μόλις ολοκληρώσω με τους σχεδιασμούς θα πρέπει να ξεκινήσω με το στήσιμο της βάσης με τα απαραίτητα δεδομένα με βάση το xl της Microsoft που δημιούργησα. Επί προσθέτως θα δημιουργήσω την εφαρμογή βασιζόμενος στα σχέδια και αυτό. Μόλις δημιουργήσω και την εφαρμογή θα πρέπει να γίνουν οι απαραίτητοι έλεγχοι ώστε να διαπιστωθεί ότι όλα λειτουργούν σωστά και για την εύρεση τυχόν bugs και διόρθωσή τους. Στο τέλος θα γίνει ξανά έλεγχος σε βάση και εφαρμογή για να δω εάν υπάρχουν περιθώρια διόρθωσης για την πιο γρήγορη δυνατή λειτουργία και ταυτόχρονα και την πιο σωστή.

4.1.2 Απαιτήσεις του Συστήματος

Όπως αναφέρθηκε και προηγουμένως χρειάστηκαν διάφορες τεχνολογίες για το ολοκληρωμένο στήσιμο τις οποίες αναλύσαμε. Συνοπτικά χρησιμοποιήσαμε

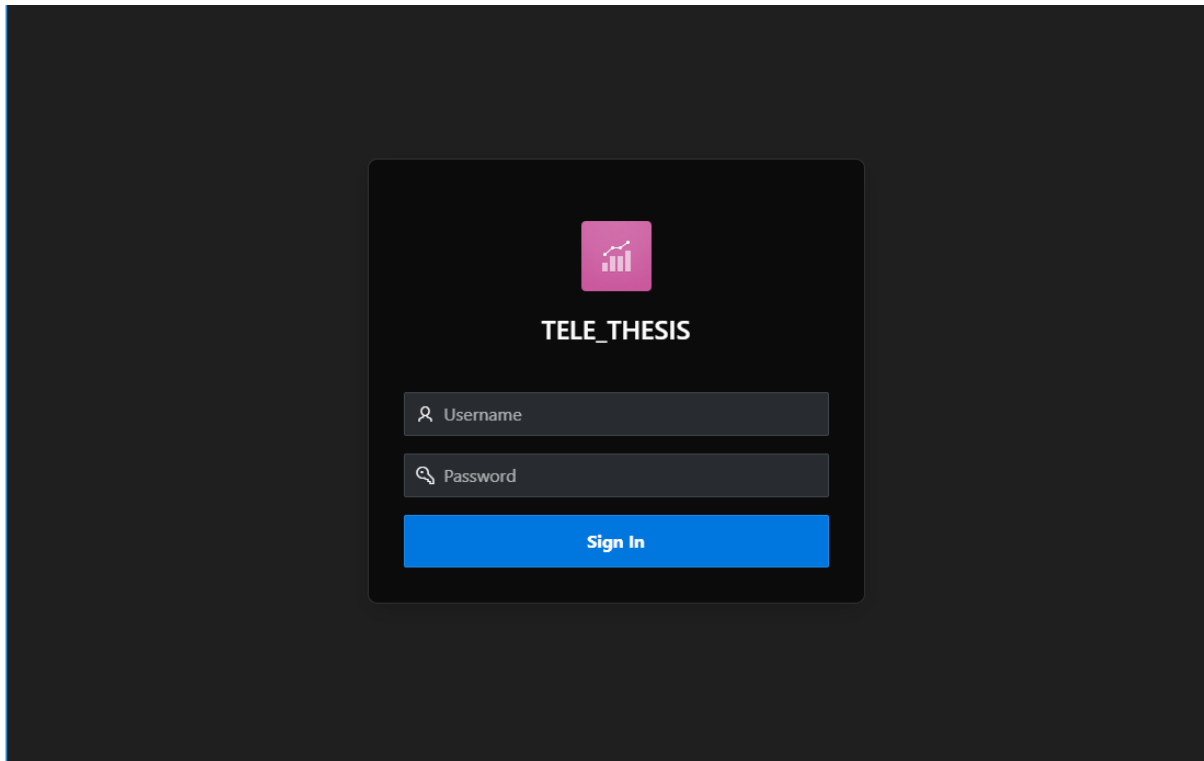
- Τα εργαλεία της Oracle για επεξεργασία και αποθήκευση δεδομένων
- APEX της Oracle ως εργαλείο για το στήσιμο της εφαρμογής (ιστοσελίδας)
- Apache Tomcat για να σηκώσω την εφαρμογή (ιστοσελίδα)
- ORDS (Oracle Rest Data Services) σαν API για την επικοινωνία του APEX με την βάση δεδομένων
- Docker Desktop σαν portable λογισμικό το οποίο σηκώνει τα εργαλεία της Oracle, Apex, Tomcat
- HTML γλώσσα σχεδιασμού σε συνδυασμό με JavaScript, jQuery, CSS
- PL/SQL για την δημιουργία SQL διαδικασιών
- Java SQLDeveloper για τον έλεγχο και την διαχείριση της βάσης δεδομένων \

4.2 Σχεδιασμός Προβλήματος & Υλοποίηση

Πρώτα από όλα και μετά από την μελέτη των απαιτήσεων που μου είχαν διατεθεί σχεδιάσα την δομή της βάσης η οποία είναι η εξής:

Στην συνέχεια ακολούθησα τα εξής βήματα για την δημιουργία των οθονών των οποίων θα βλέπουν και θα χρησιμοποιούν οι χρήστες της εφαρμογής μας. Τα βήματα που ακολουθούν:

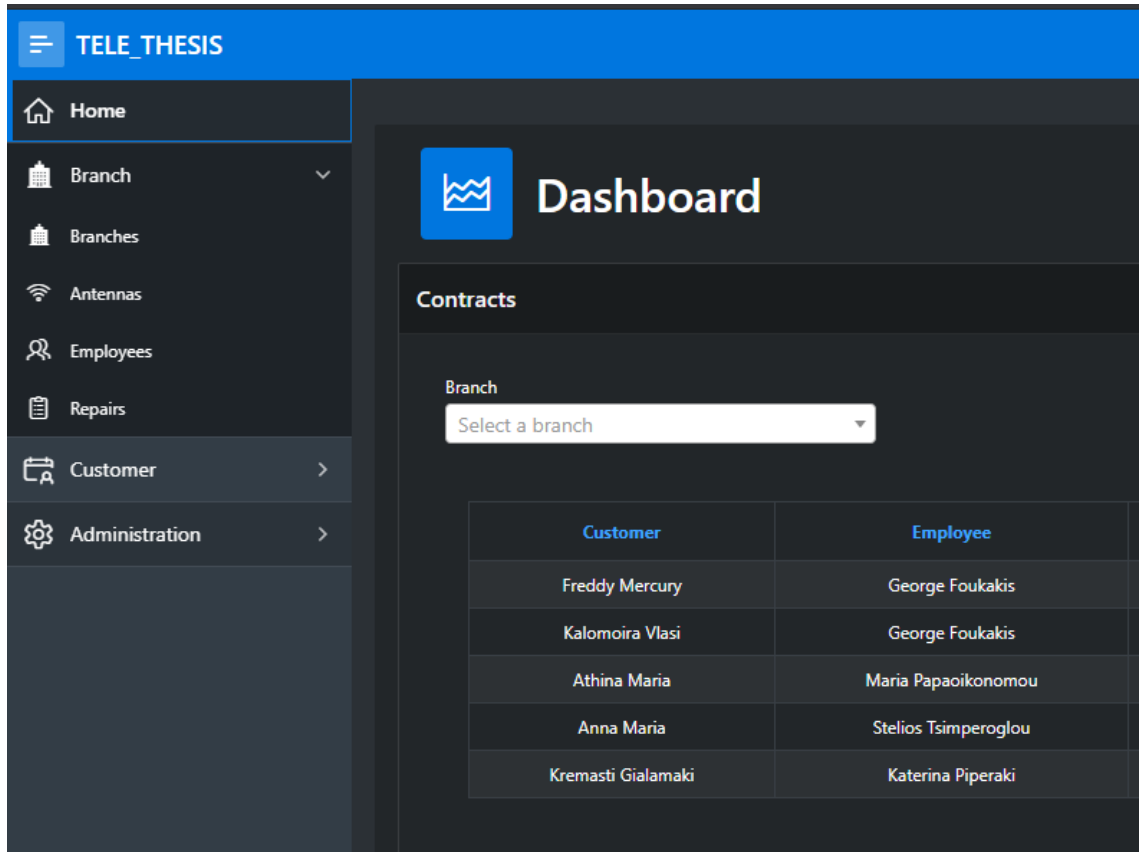
Αρχικά δημιούργησα μία σελίδα για την είσοδο στην εφαρμογή μας με τις απαραίτητες διαδικασίες αυθεντικοποίησης των χρηστών



IMGS-16

1. Έπειτα δημιούργησα ένα Navigation Bar μενού το οποίο αποτελείτε από τις συντομεύσεις των σελίδων της εφαρμογής

a. Branch



IMGS-17

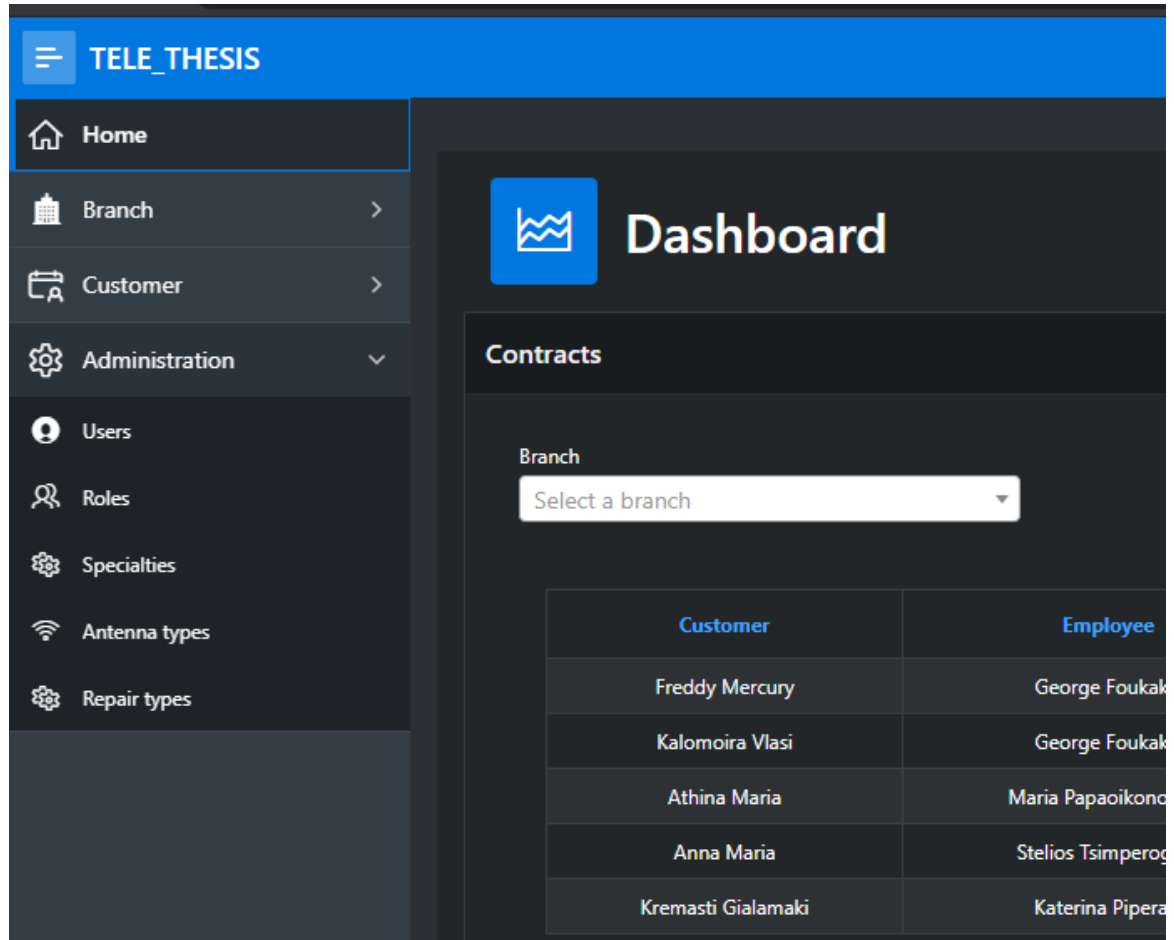
b. Customer

The screenshot displays the TELE_THESIS dashboard. The top navigation bar is blue with the text 'TELE_THESIS' and a hamburger menu icon. The left sidebar is dark grey and contains the following menu items: Home, Branch, Customer, Customers, Telephone numbers, Billing packages, Bills, Contracts, Usage per Number, and Administration. The main content area is dark grey and features a 'Dashboard' header with a blue icon. Below the header is a 'Contracts' section with a 'Branch' dropdown menu set to 'Select a branch'. A table below the dropdown lists contracts with columns for 'Customer' and 'Employee'.

Customer	Employee
Freddy Mercury	George F...
Kalomoira Vlasi	George F...
Athina Maria	Maria Papa...
Anna Maria	Stelios Tsim...
Kremasti Gialamaki	Katerina P...

IMGS-18

c. Administration

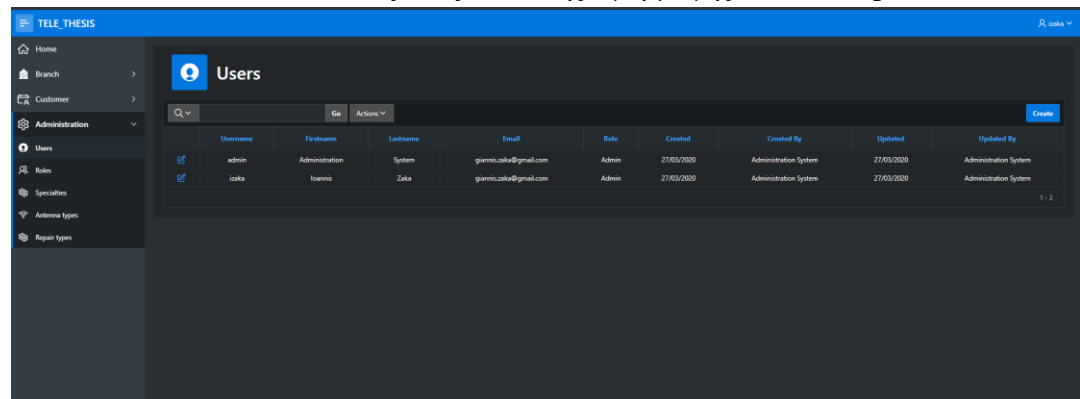


IMGS-19

2. Στην συνέχεια για το μενού Administration έφτιαξα τις σελίδες για

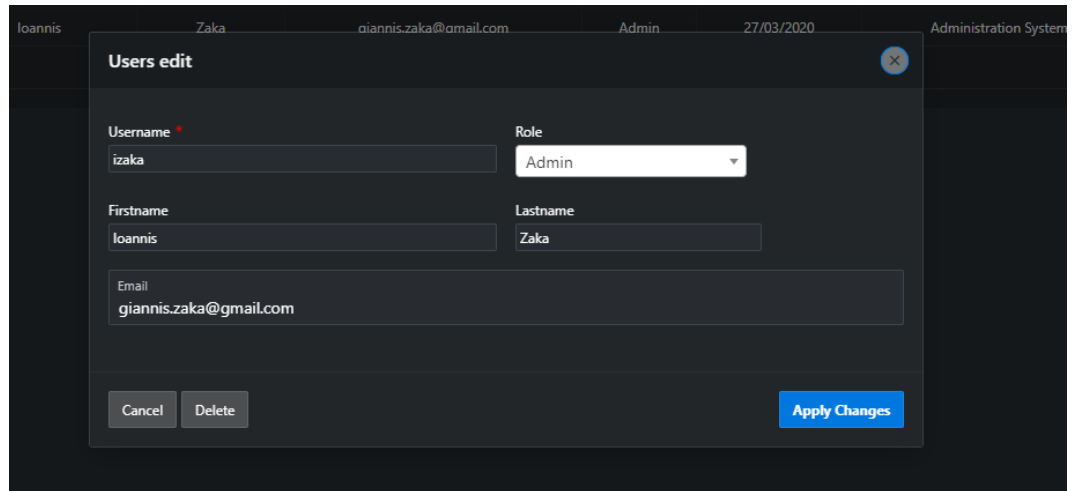
a. Users

i. Σελίδα που αναδεικνύει όλους τους users της εφαρμογής σε ένα Report



IMGS-20

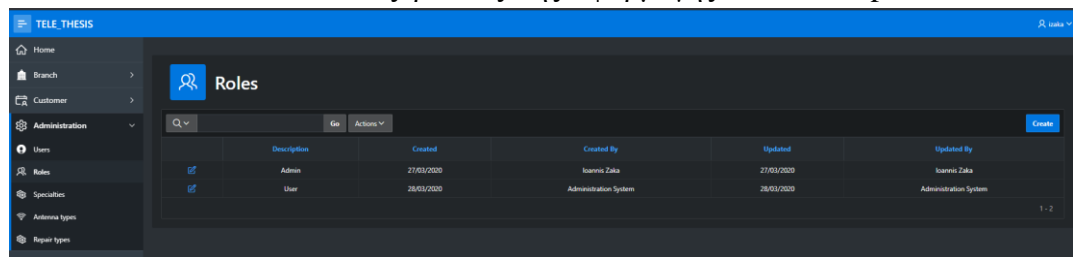
ii. Σελίδα με τις λεπτομέρειες ανά χρήστη τις οποίες μπορούμε να επεξεργαστούμε



IMGS-21

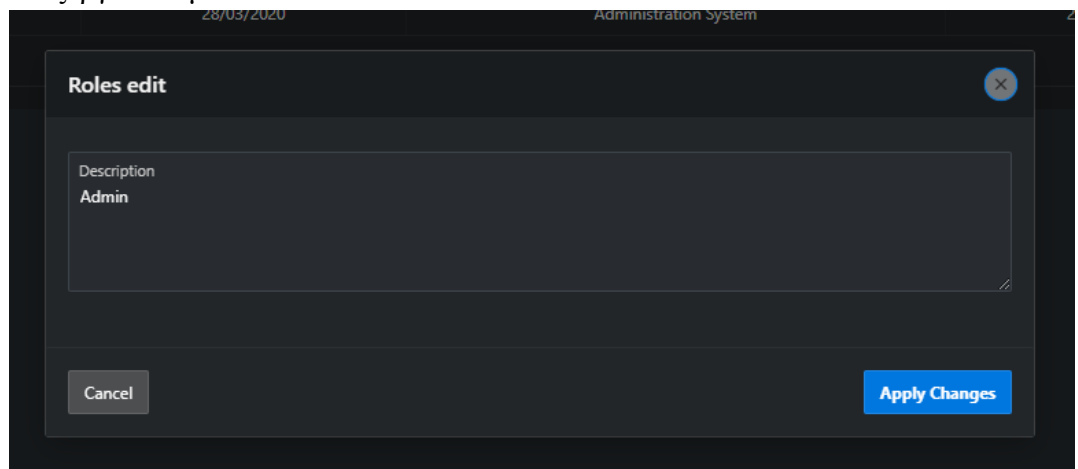
b. Roles

i. Σελίδα που αναδεικνύει τους ρόλους της εφαρμογής σε ένα Report



IMGS-22

ii. Σελίδα με τις λεπτομέρειες ανά ρόλο τις οποίες μπορούμε να επεξεργαστούμε



IMGS-23

c. Specialties

- i. Σελίδα που αναδεικνύει τις ειδικότητες σε ένα Report. Αργότερα αυτά θα εμφανίζονται σαν λίστα για να τα επιλέξει ο χρήστης

Description	Created	Created By	Updated	Updated By
Ηλεκτρολόγων Μηχανικών	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Μηχανικών Ενεργειακής Τεχνολογίας (Ηλεκτρολογικής Κατεύθυνσης)	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Μηχανικών Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Μηχανικών Πληροφορικής και Τεχνολογίας Υπολογιστών	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Μηχανικών Πληροφορικής και Επισκευών	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Μηχανικών Εργαστηρίων Πληροφορικής και Πολυμέσων	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Μηχανικών Πληροφορικής	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Μηχανικών Η/Υ	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Μηχανικών Λογισμικού	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Μηχανικών Διπλών	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Μηχανικών Αυτοματισμού	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Ηλεκτρονικών μηχανικών	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Μηχανικών Ηλεκτρονικών υπολογιστών Συστημάτων	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka

IMGS-24

- ii. Σελίδα με τις λεπτομέρειες ανά ειδικότητα τις οποίες μπορούμε να επεξεργαστούμε

Μηχανικών Τεχνολογίας, Πληροφορικής και Τηλεπικοινωνιών 27/03/2020

Specialities edit

Description
Μηχανικών Ενεργειακής Τεχνολογίας (Ηλεκτρολογικής Κατεύθυνσης)

Cancel Delete Apply Changes

Ηλεκτρονικών μηχανικών 27/03/2020

IMGS-25

d. Antenna types

- i. Σελίδα που αναδεικνύει όλους τους τύπους των κεραιών σε ένα Report. Αργότερα αυτά θα εμφανίζονται σαν λίστα για να τα επιλέξει ο χρήστης

Description	Created	Created By	Updated	Updated By
Horn	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Parabolic	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Yagi	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Monopole	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Dipole	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
Power mount	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
g	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
self-supporting lattice	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
roof-top	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
church steeple	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
wood utility pole	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka
guied lattice	27/03/2020	Ioannis Zaka	27/03/2020	Ioannis Zaka

IMGS-26

- ii. Σελίδα με τις λεπτομέρειες ανά τύπου κεραίας τις οποίες μπορούμε να επεξεργαστούμε

Antenna types edit

Description
self-supporting lattice

Cancel Delete Apply Changes

IMGS-27

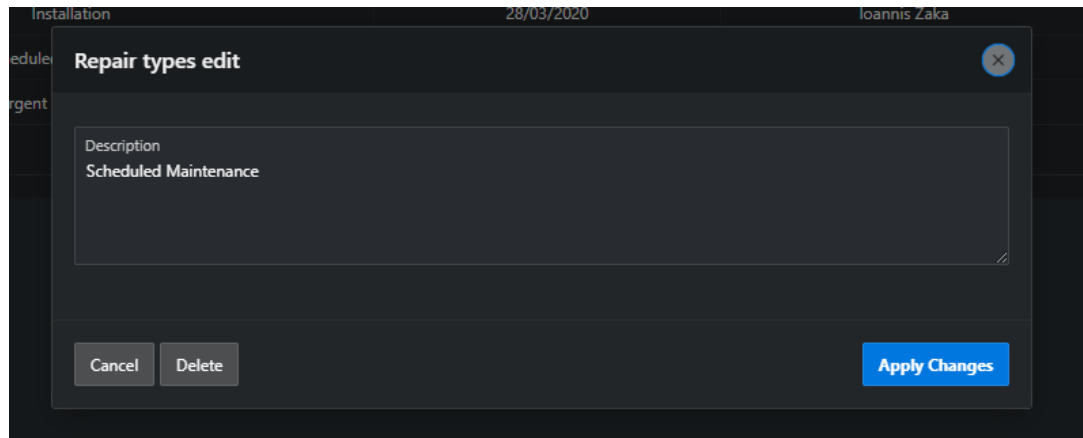
- e. Repair types

- i. Σελίδα που αναδεικνύει όλους τους τύπους επισκευής σε ένα Report. Αργότερα αυτά θα εμφανίζονται σαν λίστα για να τα επιλέξει ο χρήστης

Description	Created	Created By	Updated	Updated By
Test	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Inspection	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Installation	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Scheduled Maintenance	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Urgent Maintenance	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka

IMGS-28

- ii. Σελίδα με τις λεπτομερείς ανά τύπο επισκευής τις οποίες μπορούμε να επεξεργαστούμε

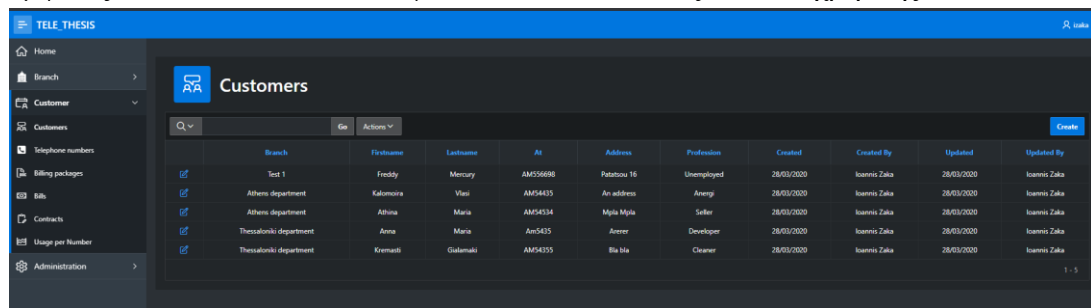


IMGS-29

3. Το μενού Customers αποτελείται από τις εξής σελίδες:

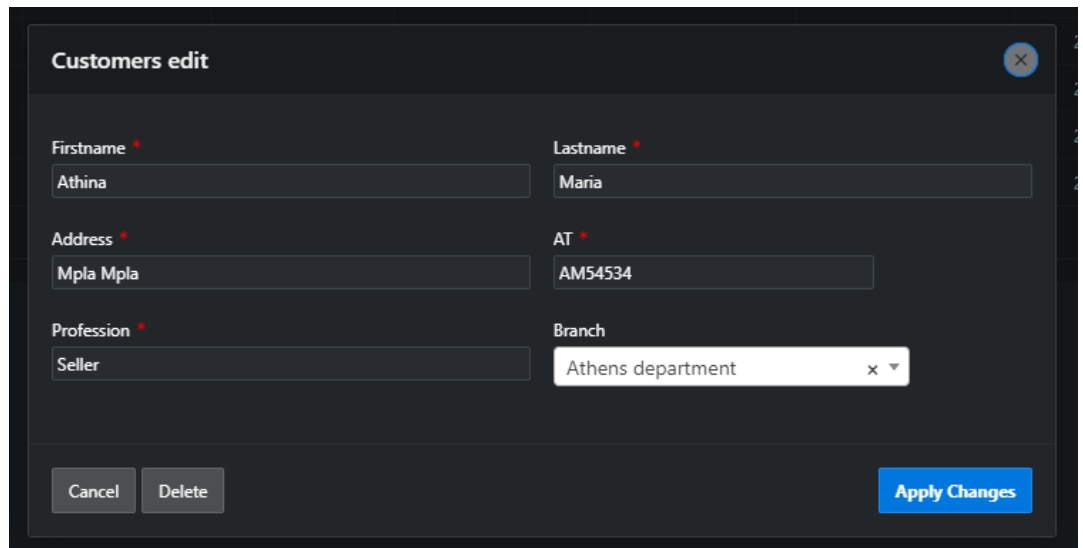
a. Customers

i. Σελίδα που αναδεικνύει τους πελάτες σε ένα Report. Αργότερα αυτά θα εμφανίζονται σαν λίστα για να τα επιλέξει ο χρήστης



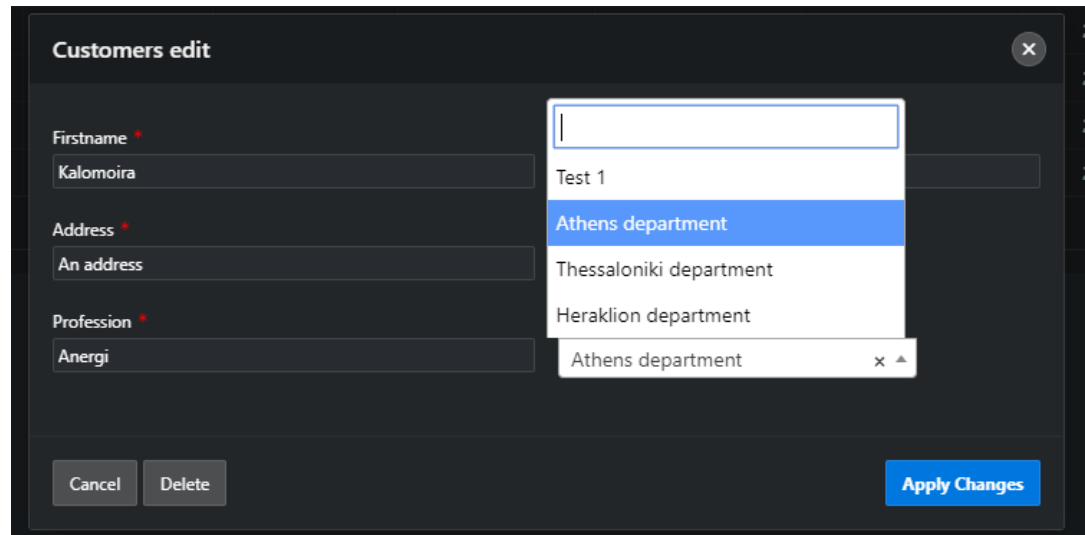
IMGS-30

ii. Σελίδα που δείχνει τις λεπτομέρειες του πελάτη τις οποίες μπορούμε να επεξεργαστούμε



IMGS-31

iii. Το πεδίο Branch αποτελείτε από μία λίστα των διαθέσιμων Branches



IMGS-32

b. Telephone numbers

i. Σελίδα που αναδεικνύει τους αριθμούς τηλεφώνων σε ένα Report. Αργότερα αυτά θα εμφανίζονται σαν λίστα για να τα επιλέξει ο χρήστης

The screenshot shows the 'Telephone numbers' report page. The table contains the following data:

Owner Id	Type	Call Number	Pub1	Pub2	Created	Created By	Updated	Updated By
Freddy Mercury	Pay as you go	432432423	4324	432	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Kalomoira Vasi	Contract	694754608	543534	543543	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Athina Maria	Contract	6998954878	543553	5435345	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Azra Maria	Contract	694584878	5345	534534	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Kremati (Galatini)	Pay as you go	154687945	65466	64564	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka

IMGS-33

ii. Σελίδα με τις λεπτομέρειες των τηλεφωνικών αριθμών τις οποίες μπορούμε να επεξεργαστούμε

The screenshot shows a 'Telephone number edit' form with the following fields and values:

- Type: Contract
- Call Number: 6999854878
- Puk1: 543553
- Puk2: 5435345
- Owner: Athina Maria

Buttons at the bottom: Cancel, Delete, and Apply Changes.

IMGS-34

- iii. Το πεδίο Owner αποτελείται από μία λίστα με τους διαθέσιμους Customers

The screenshot shows the 'Telephone number edit' form with the Owner dropdown menu open. The list of customers is:

- Freddy Mercury
- Kalomoira Vlasi
- Athina Maria (highlighted)
- Anna Maria
- Kremasti Gialamaki

The other fields and values are the same as in the previous screenshot: Type: Contract, Call Number: 6999854878, Puk1: 543553, Puk2: 5435345. Buttons at the bottom: Cancel, Delete, and Apply Changes.

IMGS-35

c. Billing packages

- i. Σελίδα που αναδεικνύει τα πακέτα προσφορών σε ένα Report. Αργότερα αυτά θα εμφανίζονται σαν λίστα για να τα επιλέξει ο χρήστης

Description	Free Minutes	Free Messages	Fixed	Cost per minute	Cost per message	Created	Created By	Updated	Updated By
Student 1000	1000	1000	5	.1	.2	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Student 300	300	300	20	.2	.25	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Business 200	200	200	40	.4	.3	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka

IMGS-36

- ii. Σελίδα με τις λεπτομέρειες των πακέτων προσφορών τις οποίες μπορούμε να επεξεργαστούμε

IMGS-37

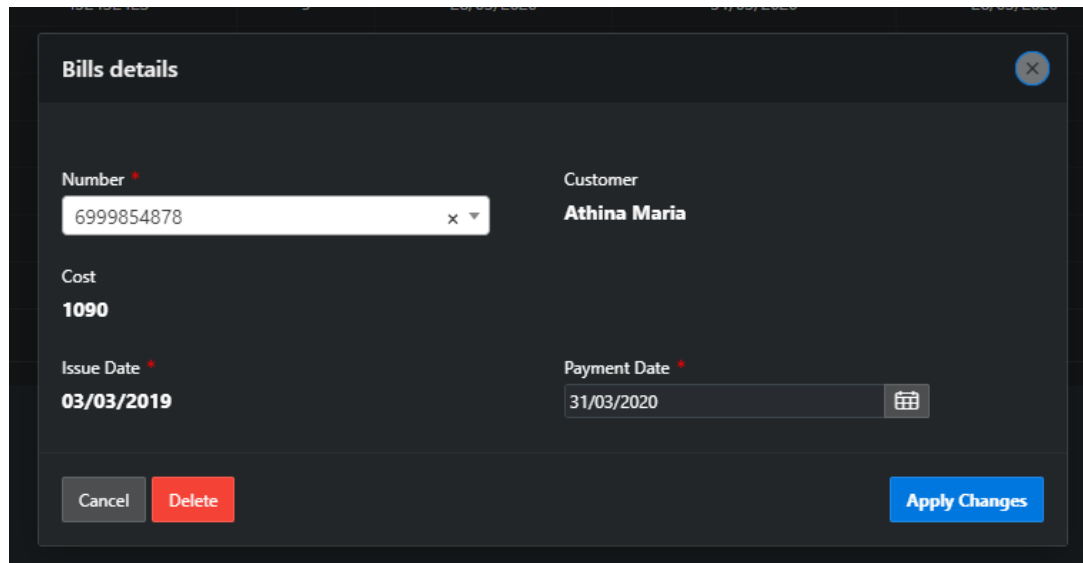
d. Bills

- i. Σελίδα που αναδεικνύει όλους τους λογαριασμούς σε ένα Report

Customer	Call Number	Cost	Issue Date	Payment Date	Created	Created By	Updated	Updated By
Freddy Mercury	432432423	5	28/03/2020	31/03/2020	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Freddy Mercury	432432423	5	31/03/2020	30/04/2020	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Kalomenia Vlassi	6947384658	145	05/03/2019	30/06/2019	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Kalomenia Vlassi	6947384658	405	28/03/2020	28/03/2020	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Athina Maria	6999554878	1090	03/03/2019	31/03/2020	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Athina Maria	6999554878	2050	28/03/2020	31/03/2020	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Anna Maria	6945384378	95	28/03/2020	31/03/2020	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka

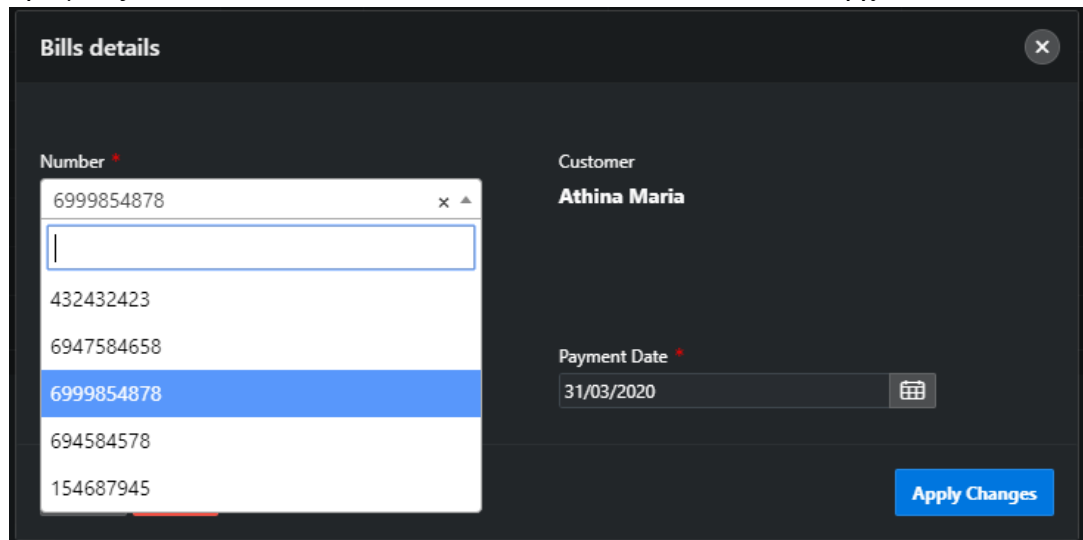
IMGS-38

- ii. Σελίδα με τις λεπτομέρειες των λογαριασμών τις οποίες μπορούμε να επεξεργαστούμε. Το cost βγαίνει αυτόματα από διαδικασίες που τρέχουν πίσω στην βάση δεδομένων.



IMGS-39

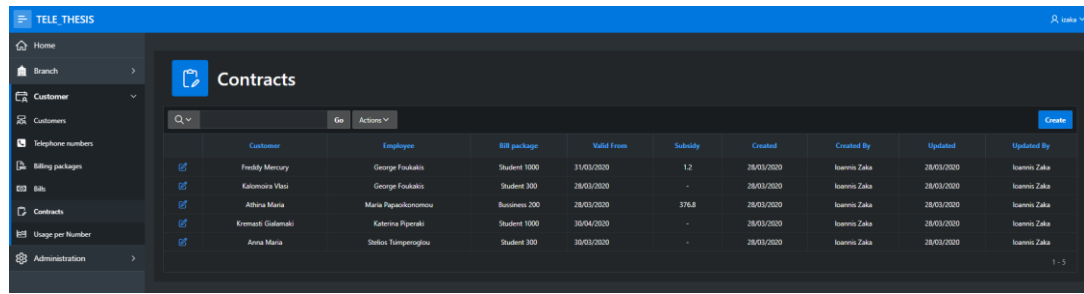
- iii. Το πεδίο Number αποτελείτε από τους διαθέσιμους τηλεφωνικούς αριθμούς που υπάρχουν



IMGS-40

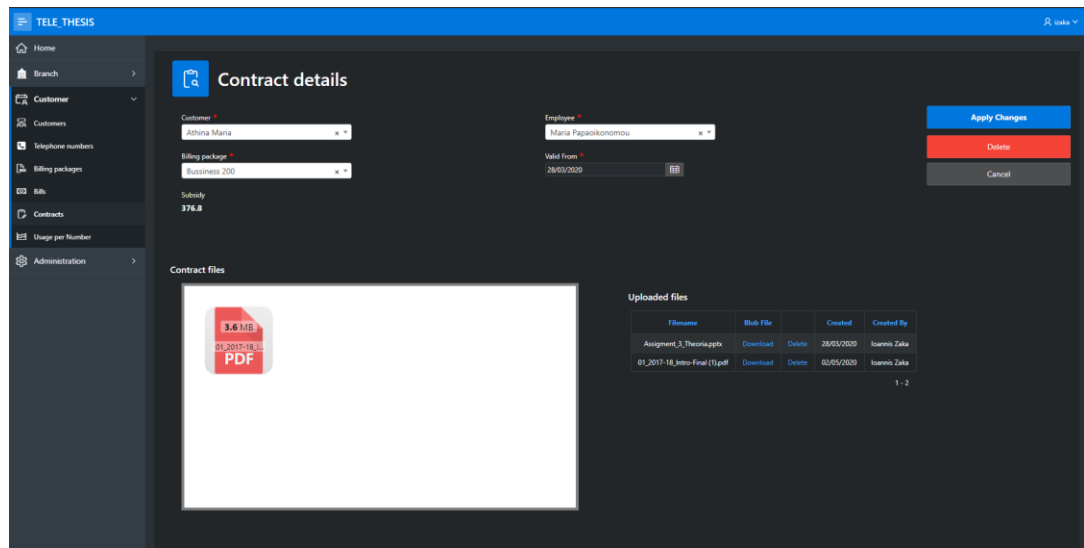
e. Contracts

- i. Σελίδα που αναδεικνύει όλα τα συμβόλαια σε Report. Αργότερα αυτά θα εμφανίζονται σαν λίστα για να τα επιλέξει ο χρήστης



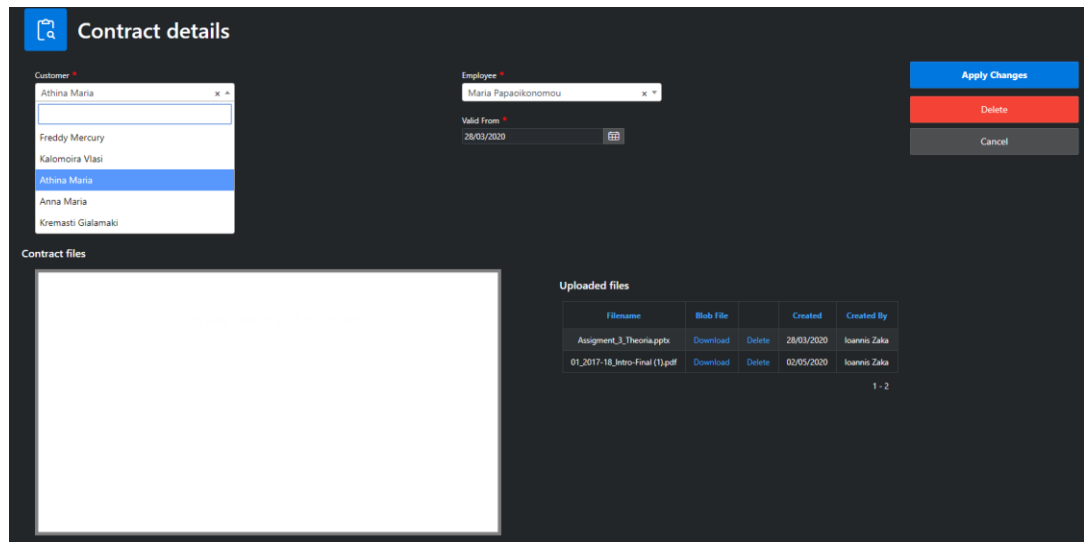
IMGS-41

- ii. Σελίδα με τις λεπτομέρειες των συμβολαίων τις οποίες μπορούμε να επεξεργαστούμε



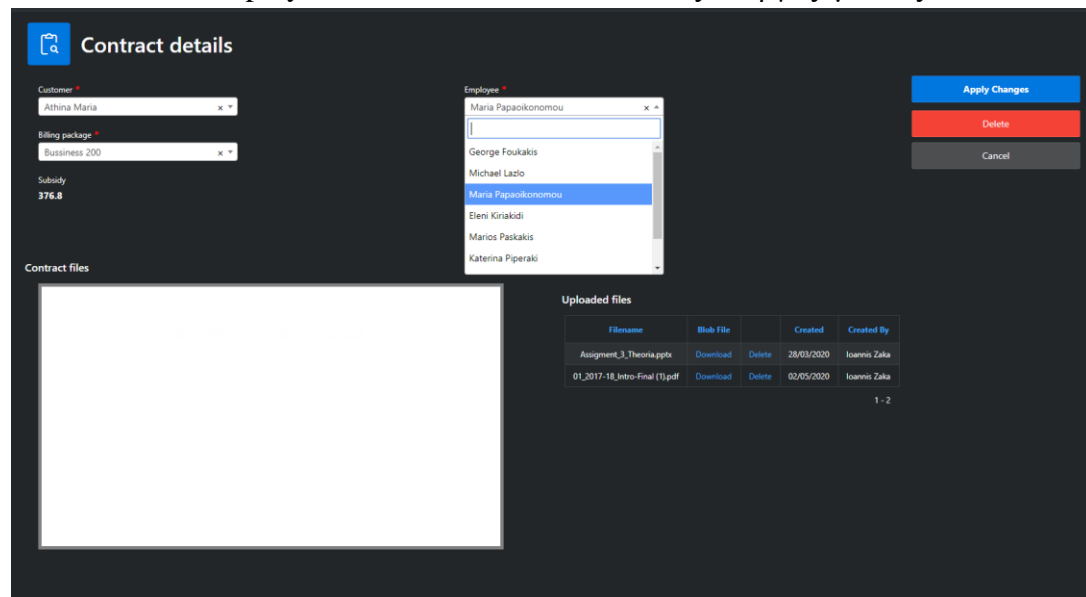
IMGS-42

- iii. Το πεδίο Customer αποτελείται από τους διαθέσιμους πελάτες



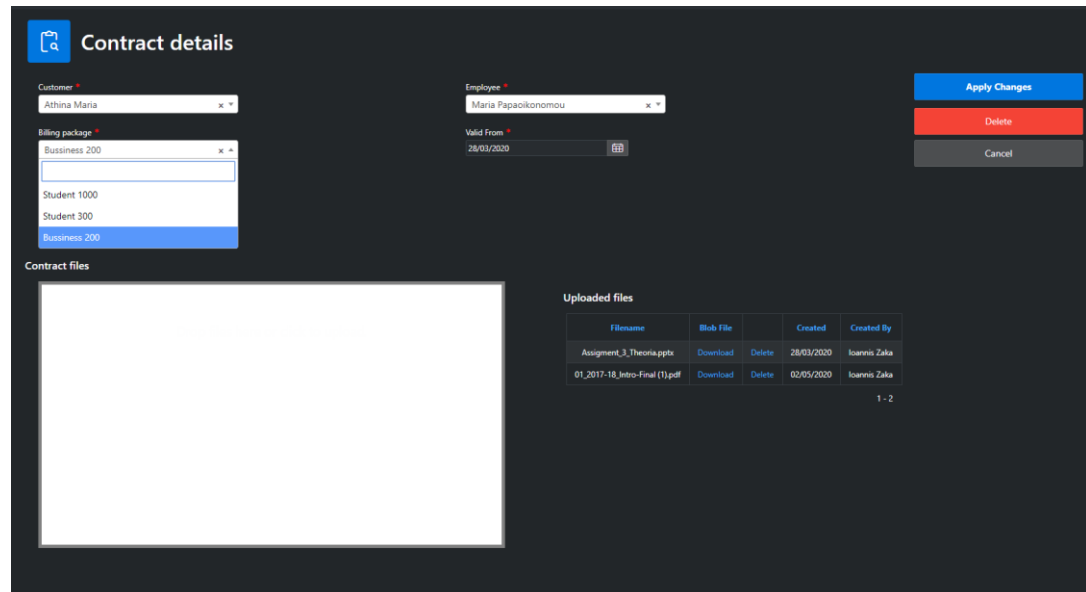
IMGS-43

- iv. Το πεδίο Employee αποτελείτε από τους εργαζόμενους



IMGS-44

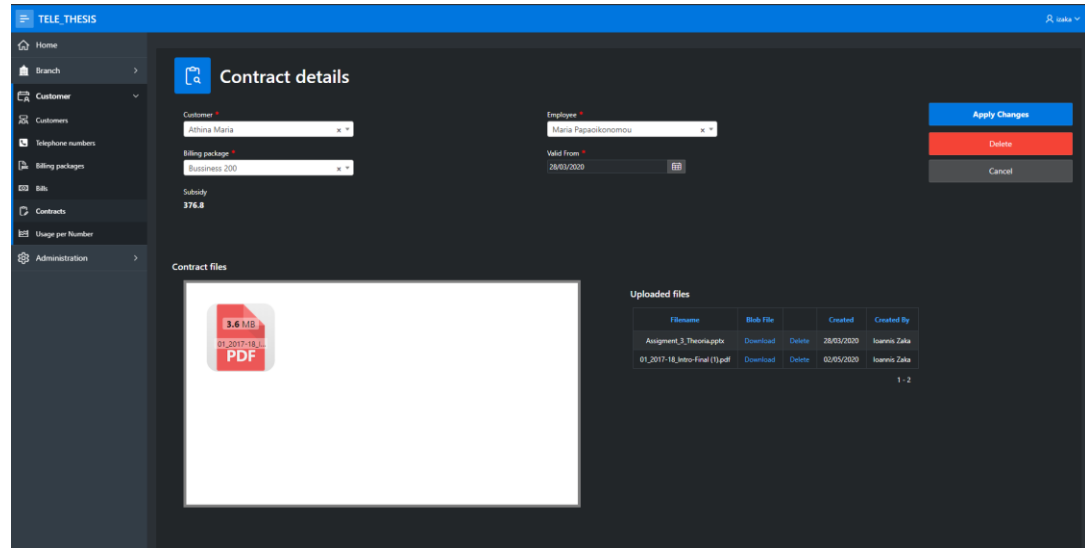
- v. Το πεδίο Billing Package αποτελείτε από τα διαθέσιμα πακέτα προσφορών



IMGS-45

- vi. Στο πεδίο Contract files μπορούμε είτε με ένα απλό κλικ στο λευκό πεδίο είτε με Drag and Drop στο λευκό πεδίο να ανεβάσουμε αρχεία. Έπειτα τα ανεβασμένα αρχεία αναδεικνύονται στο Report δεξιά με τις

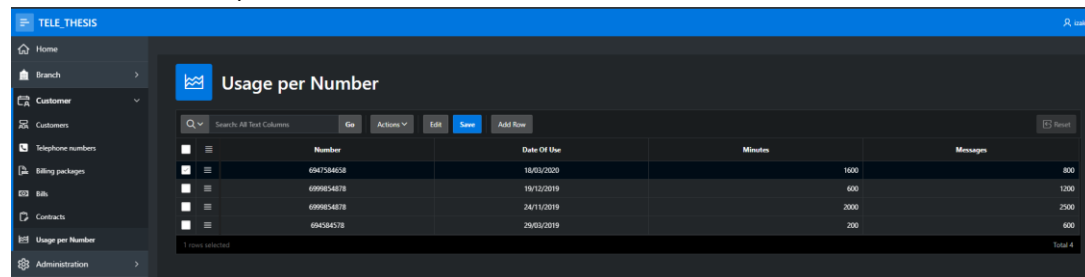
επιλογές Download για το κατέβασμα τους ή Delete για την διαγραφή τους



IMGS-46

f. Usage per Number

- i. Σελίδα που αναδεικνύει την χρήση Μηνυμάτων και Λεπτών ομιλίας ανά αριθμό σε ένα Report. Το συγκεκριμένο αυτό Report είναι Editable region στο οποίο μπορούμε να επεξεργαστούμε τις λεπτομέρειες ανά Row με ένα απλό κλικ



IMGS-47

4. Στο μενού Branch έχουμε τις εξής επιλογές σελίδων:

a. Branches

- i. Σελίδα που αναδεικνύει τα καταστήματα σε ένα Report. Αργότερα αυτά θα εμφανίζονται σαν λίστα για να τα επιλέξει ο χρήστης

Branch Employees	Description	Address	Administrative	Created	Created By	Updated	Updated By
	Test 1	Theodorou 2		28/03/2020	Ioannis Zaka	28/03/2020	IoannisZaka
	Athens department	Ano Kipseli - patatsou 16	Maria Papaoikonomou	28/03/2020	Ioannis Zaka	28/03/2020	IoannisZaka
	Thessaloniki department	Saniouli House 4	Katerina Piperaki	28/03/2020	Ioannis Zaka	28/03/2020	IoannisZaka
	Heraklion department	Estauromenou 19	Giorgos Papanikolaou	28/03/2020	Ioannis Zaka	28/03/2020	IoannisZaka

IMGS-48

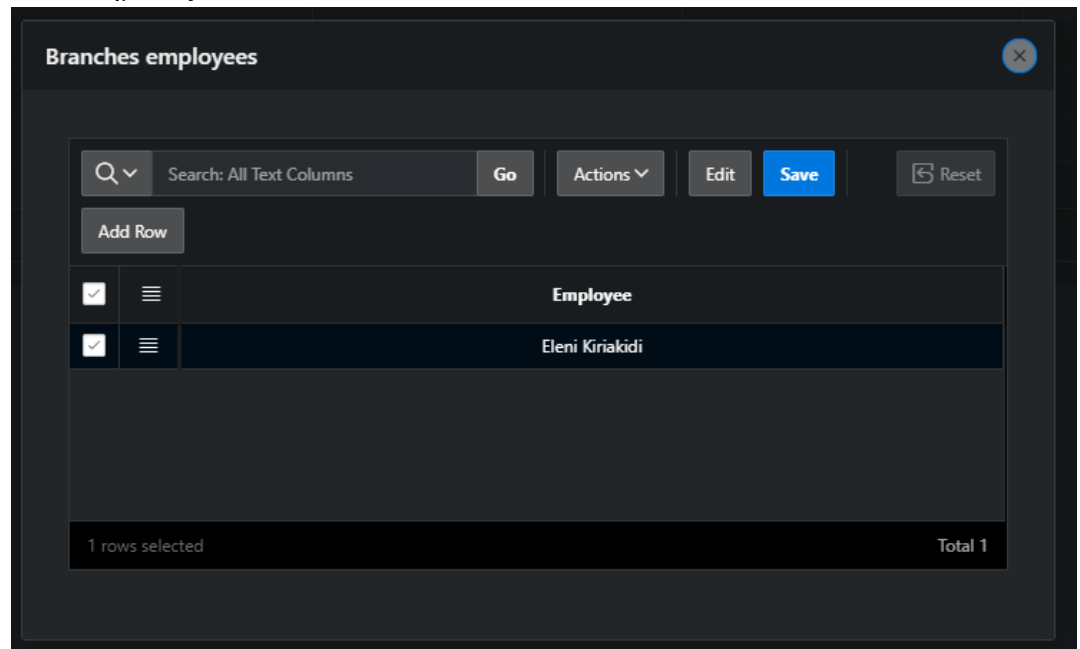
- ii. Σελίδα με τις λεπτομέρειες ανά κατάστημα τις οποίες μπορούμε να επεξεργαστούμε

IMGS-49

- iii. Το πεδίο Administrative αποτελείται από μία λίστα των διαθέσιμων εργαζομένων οι οποίοι είναι διοικητικά στελέχη

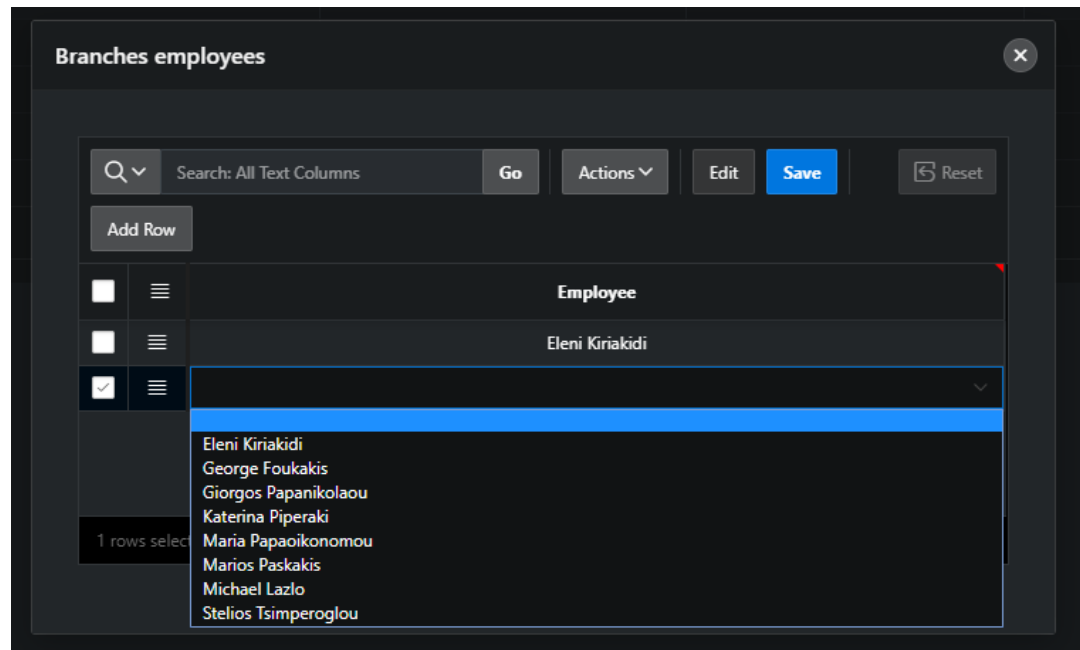
IMGS-50

- iv. Στην σελίδα Branches employees (Εργαζόμενοι καταστημάτων) αναδεικνύεται μία λίστα που περιλαμβάνει τους εργαζόμενους του καταστήματος



IMGS-51

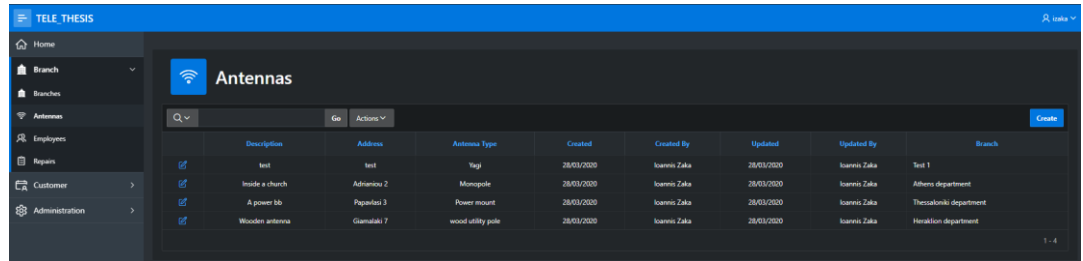
- v. Το πεδίο Employee αποτελείτε από μία λίστα διαθέσιμων εργαζομένων



IMGS-52

b. Antennas

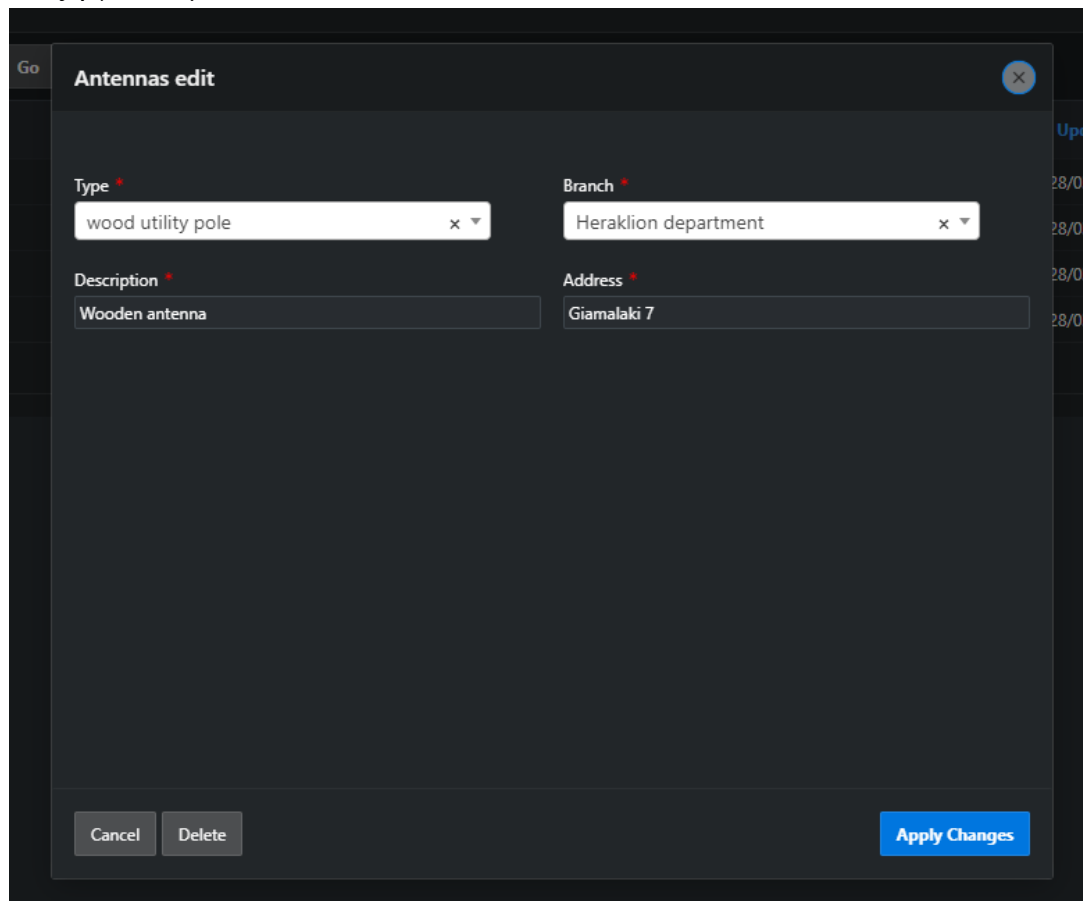
- i. Σελίδα που αναδεικνύει όλες τις κεραίες σε ένα Report. Αργότερα αυτά θα εμφανίζονται σαν λίστα για να τα επιλέξει ο χρήστης



Description	Address	Antenna Type	Created	Created By	Updated	Updated By	Branch
test	test	Yagi	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka	Test 1
Inside a church	Adrianou 2	Monopole	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka	Athens department
A power bb	Papafisi 3	Power mount	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka	Thessaloniki department
Wooden antenna	Giamalaki 7	wood utility pole	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka	Heraklion department

IMGS-53

- ii. Σελίδα με τις λεπτομέρειες ανά κεραία τις οποίες μπορούμε να επεξεργαστούμε



Go Antennas edit

Type *
wood utility pole x

Branch *
Heraklion department x

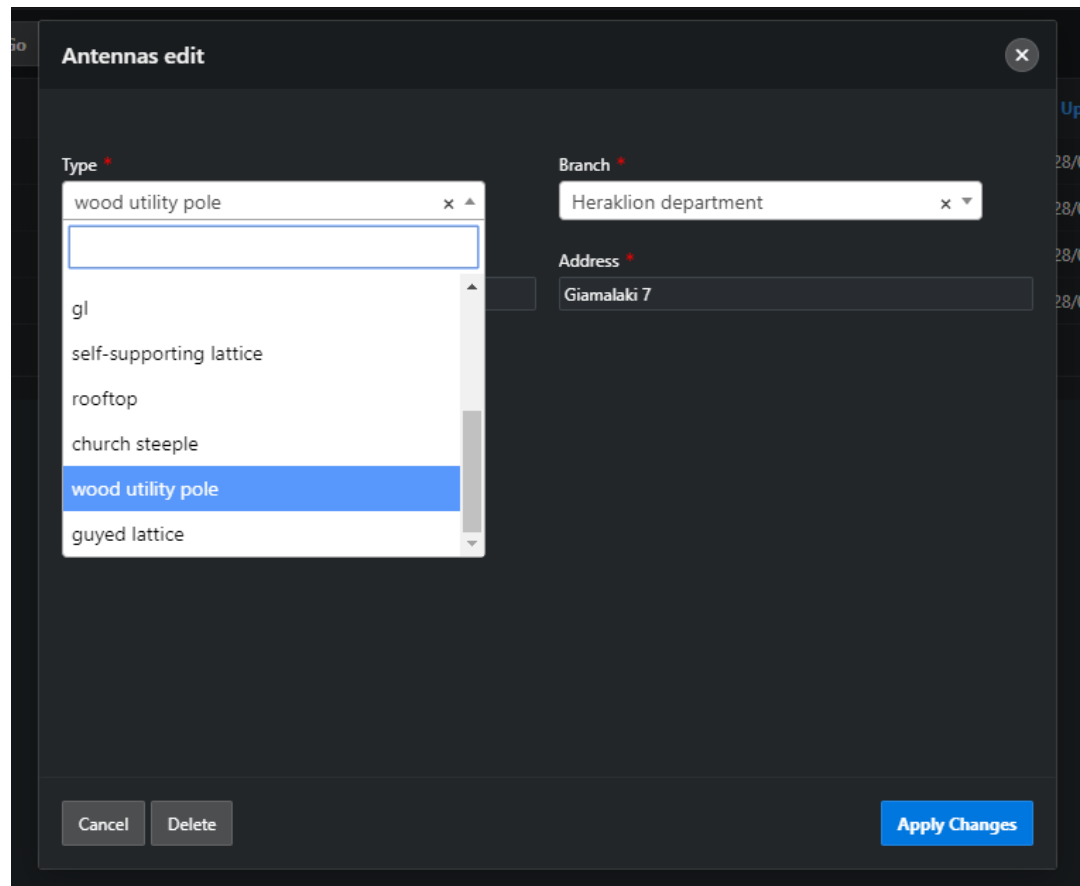
Description *
Wooden antenna

Address *
Giamalaki 7

Cancel Delete Apply Changes

IMGS-54

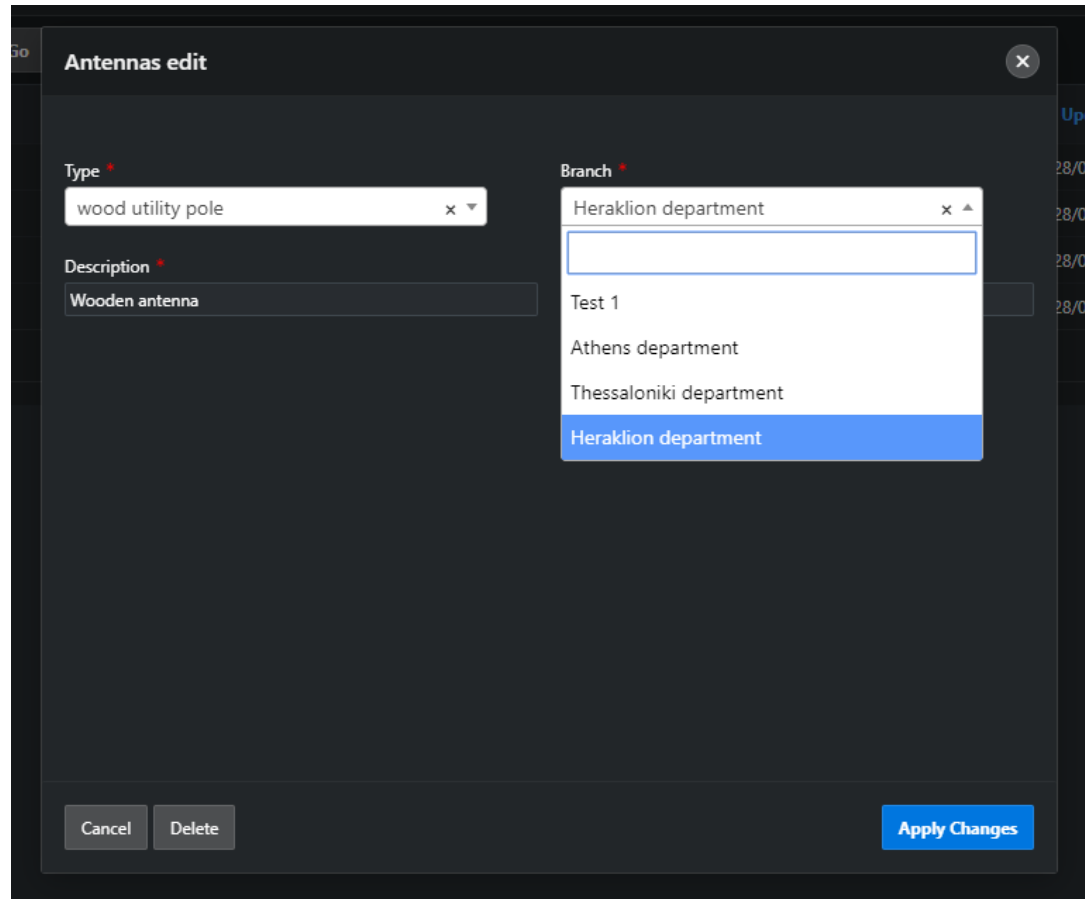
iii. Το πεδίο Type αποτελείται από μία λίστα τύπων κεραιών



The image shows a software dialog box titled "Antennas edit". It contains three main input fields: "Type", "Branch", and "Address". The "Type" field is currently open, displaying a dropdown list of antenna types. The list includes "gl", "self-supporting lattice", "rooftop", "church steeple", "wood utility pole" (which is highlighted in blue), and "guyed lattice". The "Branch" field is set to "Heraklion department" and the "Address" field is set to "Giamalaki 7". At the bottom of the dialog, there are three buttons: "Cancel", "Delete", and "Apply Changes".

IMGS-55

iv. Το πεδίο Branch αποτελείται από μία λίστα των διαθέσιμων καταστημάτων



IMGS-56

c. Employees

i. Σελίδα που αναδεικνύει όλους τους εργαζομένους σε Report

Branch	Firstname	Lastname	AT	Type	Salary	Created	Created By	Updated	Updated By
Test 1	Eleni	Kiriakidi	AP52453	Technician	980	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Athens department	Stefos	Toumpoglou	GK45478	Technician	900	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Thessaloniki department	Katerina	Piperaki	FK85734	Administrative	1300	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
Heraklion department	Maria	Papaikononitou	AK48756	Administrative	1200	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
-	George	Foukalis	AM48987	Technician	798	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
-	Georgios	Papantidakis	AN57887	Administrative	1600	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
-	Michiel	Lado	AM42342	Administrative	1600	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka
-	Marcos	Poukalis	UT84885	Technician	800	28/03/2020	Ioannis Zaka	28/03/2020	Ioannis Zaka

IMGS-57

ii. Σελίδα με τις λεπτομέρειες των εργαζομένων τις οποίες μπορούμε να επεξεργαστούμε

The screenshot shows a dark-themed 'Employees edit' form. It contains the following fields: 'Firstname' with the value 'Stelios', 'Lastname' with 'Tsimperoglou', 'AT' with 'GK45478', and 'Salary' with '900'. The 'Type' field is a dropdown menu currently set to 'Technitian'. At the bottom, there are three buttons: 'Cancel', 'Delete', and 'Apply Changes'.

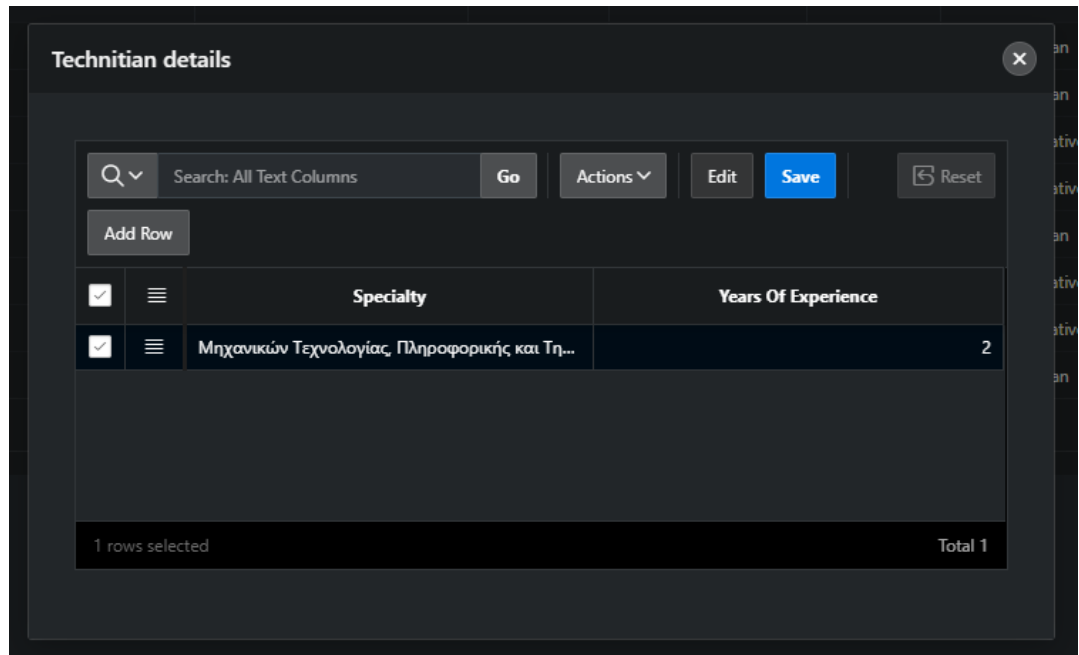
IMGS-58

iii. Το πεδίο Type αποτελείτε από μία λίστα τύπων εργαζομένων

This screenshot is identical to the previous one, but the 'Type' dropdown menu is open, showing a list of options. The visible options are 'Technitian' (which is highlighted in blue) and 'Administrative'. The 'Apply Changes' button is partially visible at the bottom right.

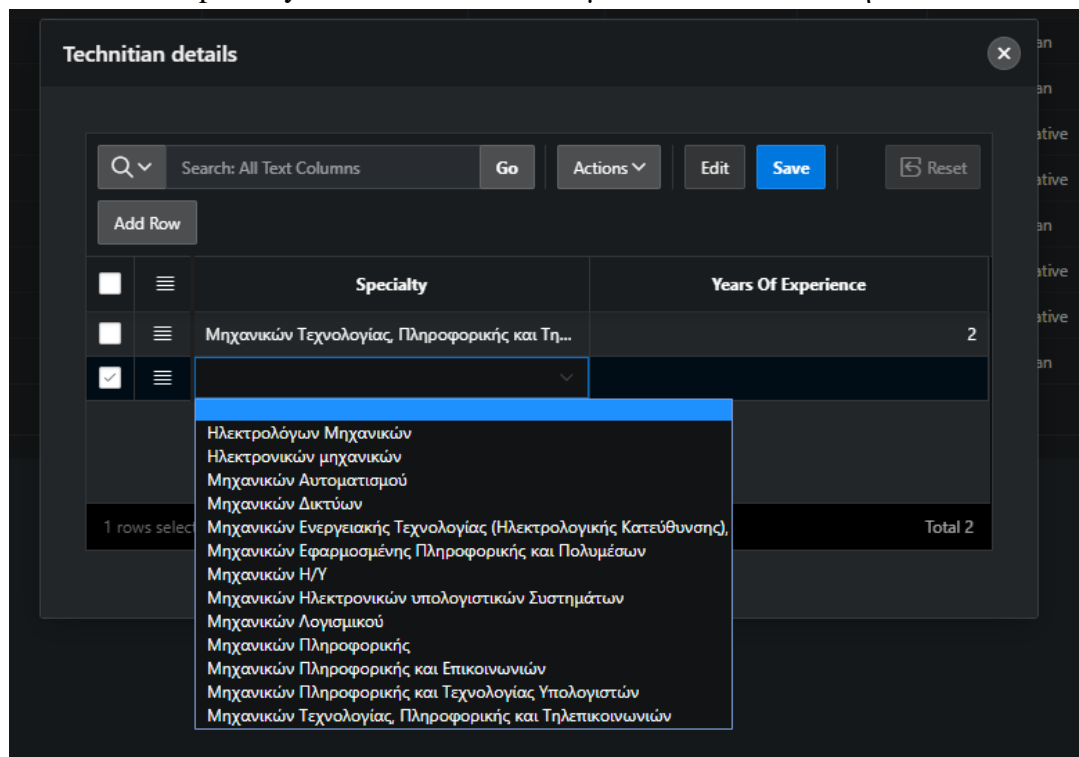
IMGS-59

iv. Οι εργαζόμενοι οι οποίοι ανήκουν στην κατηγορία «Τεχνικοί» έχουν την εξής σελίδα η οποία αναδεικνύει κάποιες έξτρα πληροφορίες



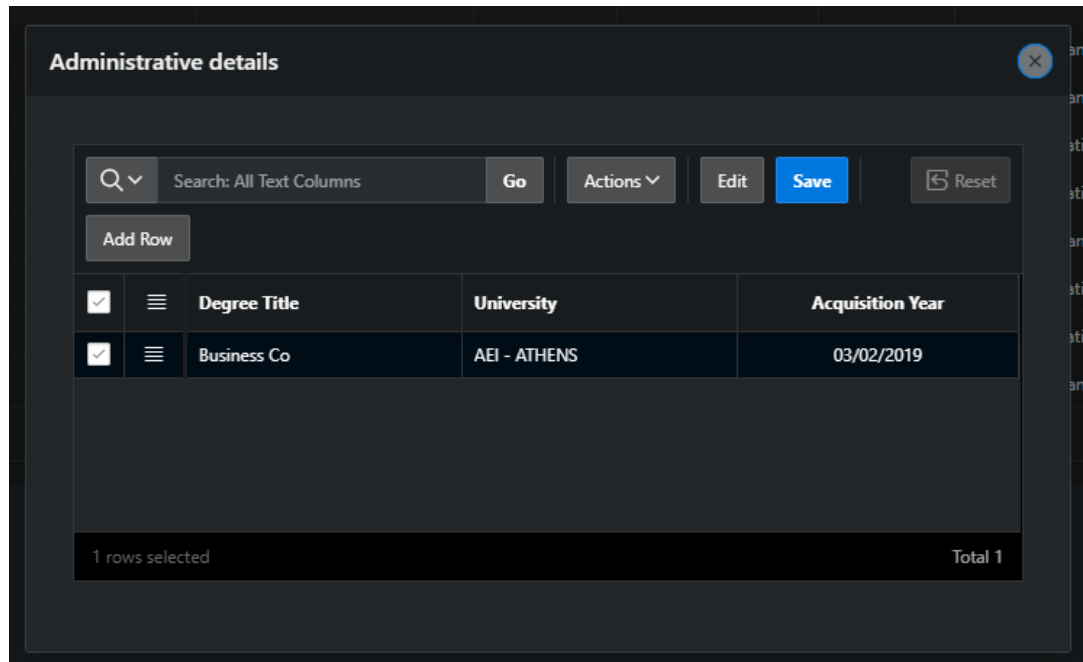
IMGS-60

- v. Το πεδίο Specialty αποτελείτε από μία λίστα ειδικοτήτων



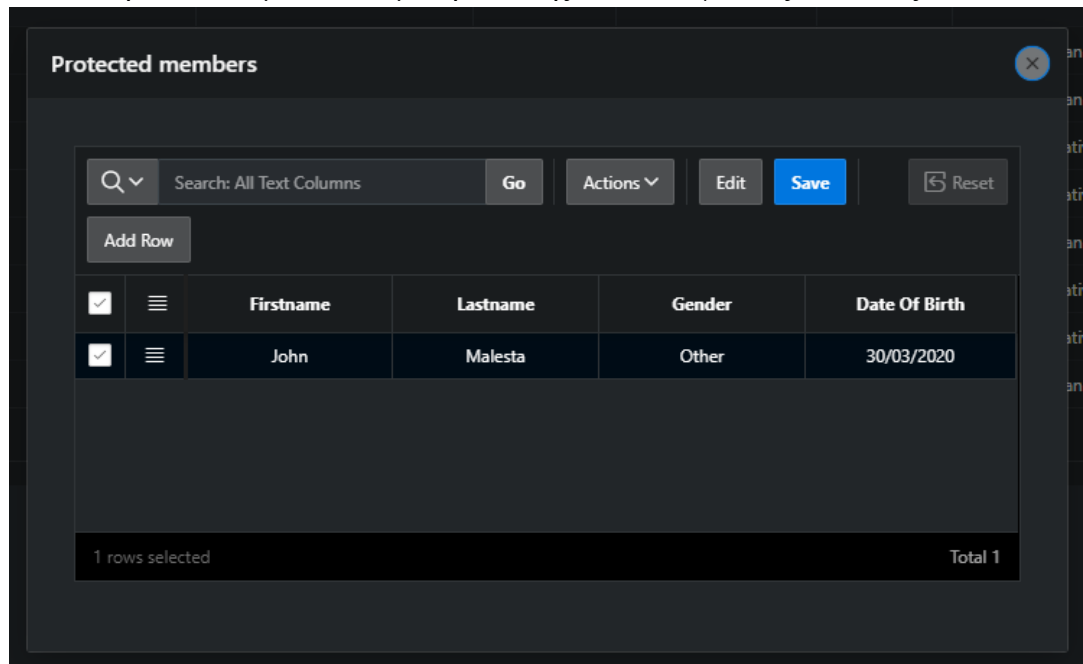
IMGS-61

- vi. Επίσης όπως οι «Τεχνικοί» έτσι και οι «Διοικητικοί» τύποι εργαζομένων έχουν δικιά τους σελίδα με έξτρα δεδομένα



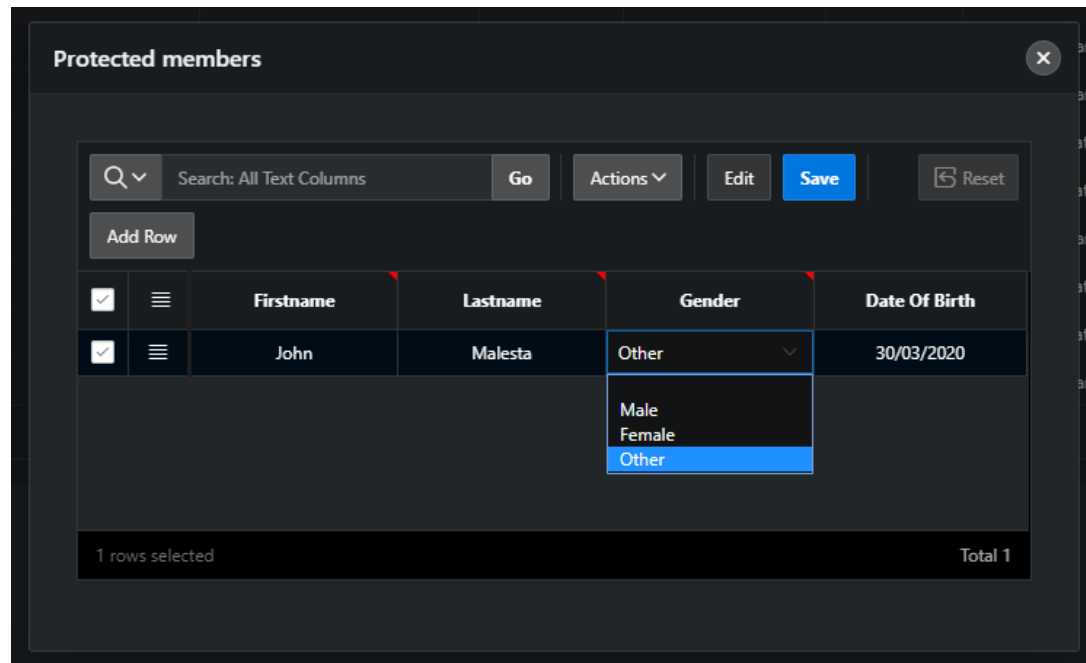
IMGS-62

- vii. Όλοι οι εργαζόμενοι έχουν και μία σελίδα στην οποία αναδεικνύονται τα προστατευόμενα μέλη της οικογένειάς τους



IMGS-63

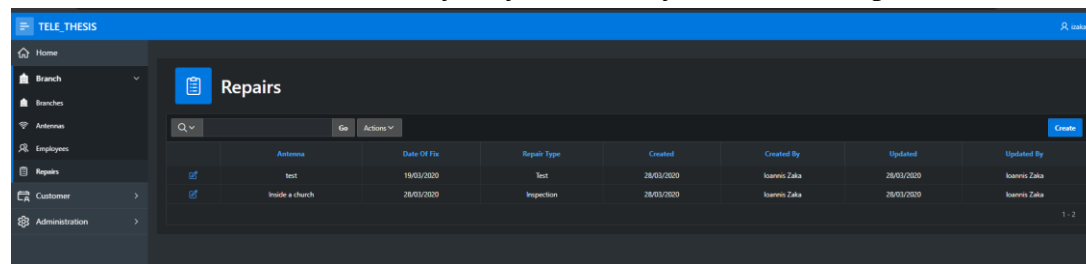
viii. Το πεδίο Gender αποτελείται από μία λίστα με διαθέσιμα Genders



IMGS-64

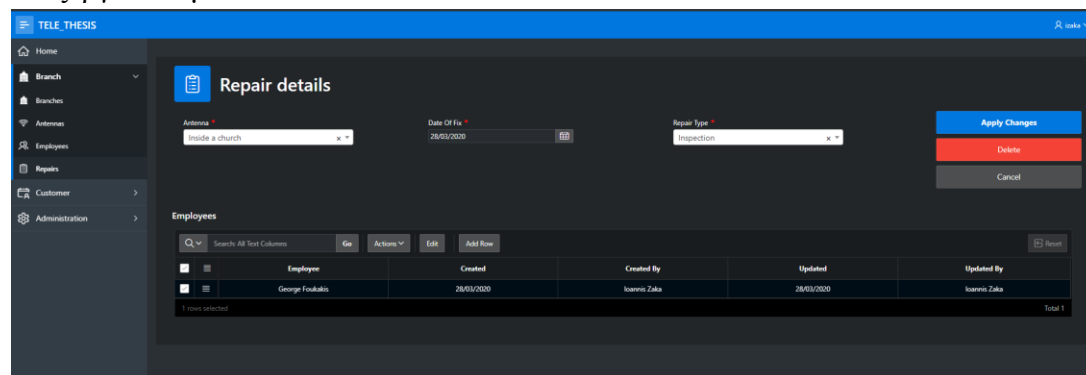
d. Repairs

i. Σελίδα που αναδεικνύει όλες τις επισκευές σε ένα Report



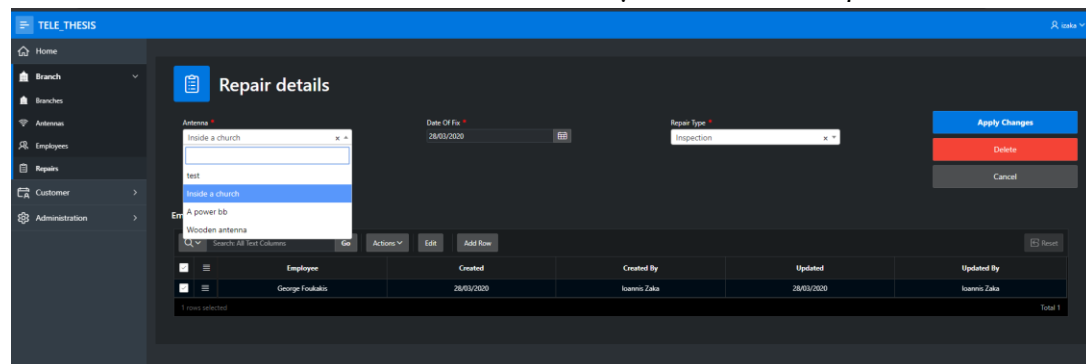
IMGS-65

ii. Σελίδα με τις λεπτομέρειες των επισκευών τις οποίες μπορούμε να επεξεργαστούμε



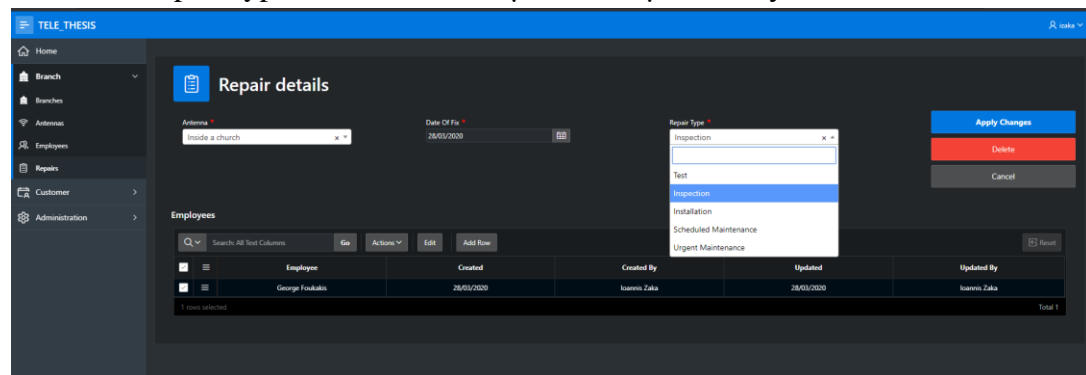
IMGS-66

iii. Το πεδίο Antenna αποτελείται από μία λίστα κεραιών



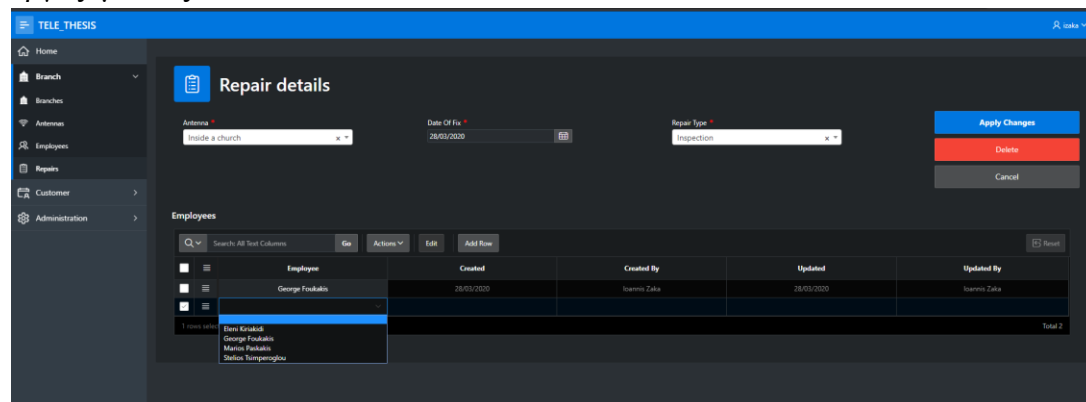
IMGS-67

iv. Το πεδίο Repair type αποτελείται από μία λίστα με τύπους επισκευών

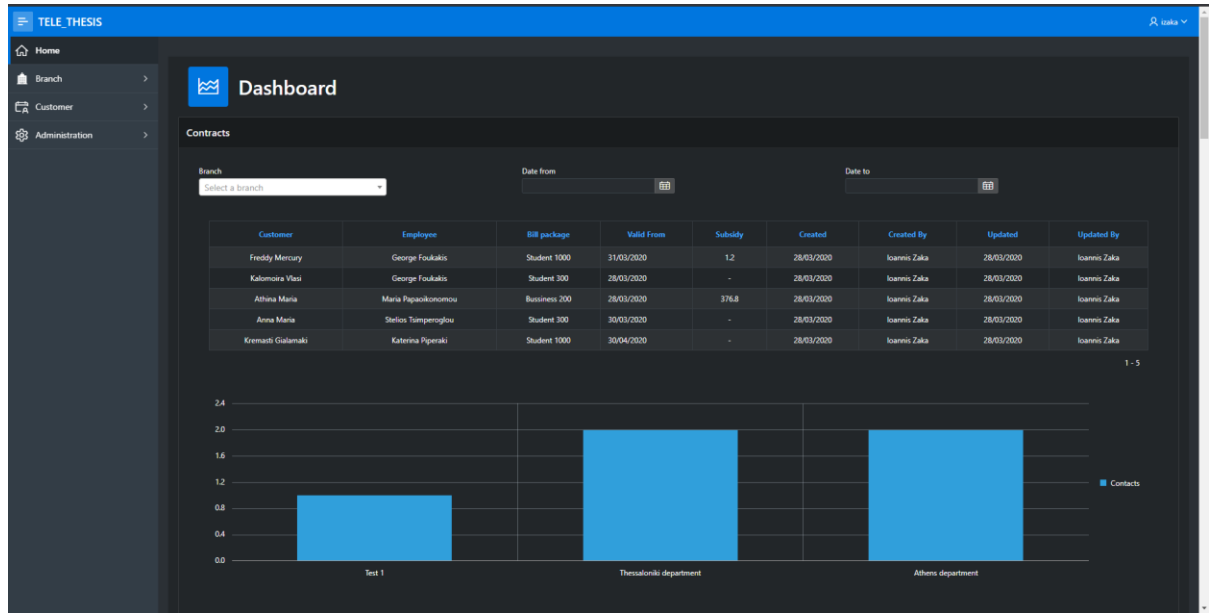


IMGS-68

v. Το πεδίο Employee αποτελείται από μία λίστα με τους τεχνικούς εργαζομένους



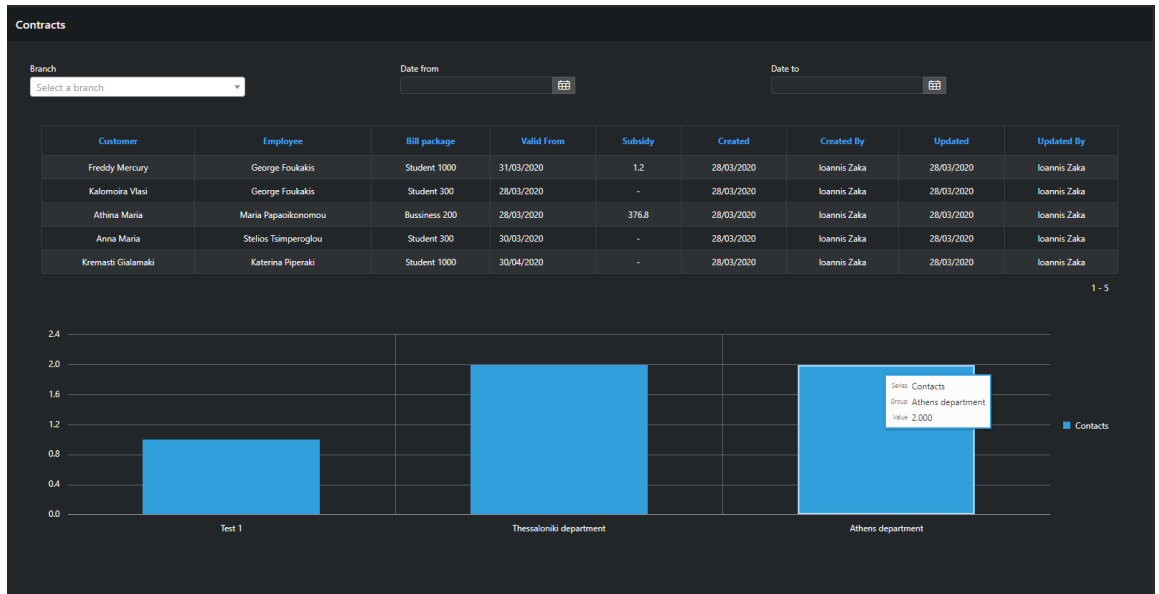
IMGS-69



IMGS-70

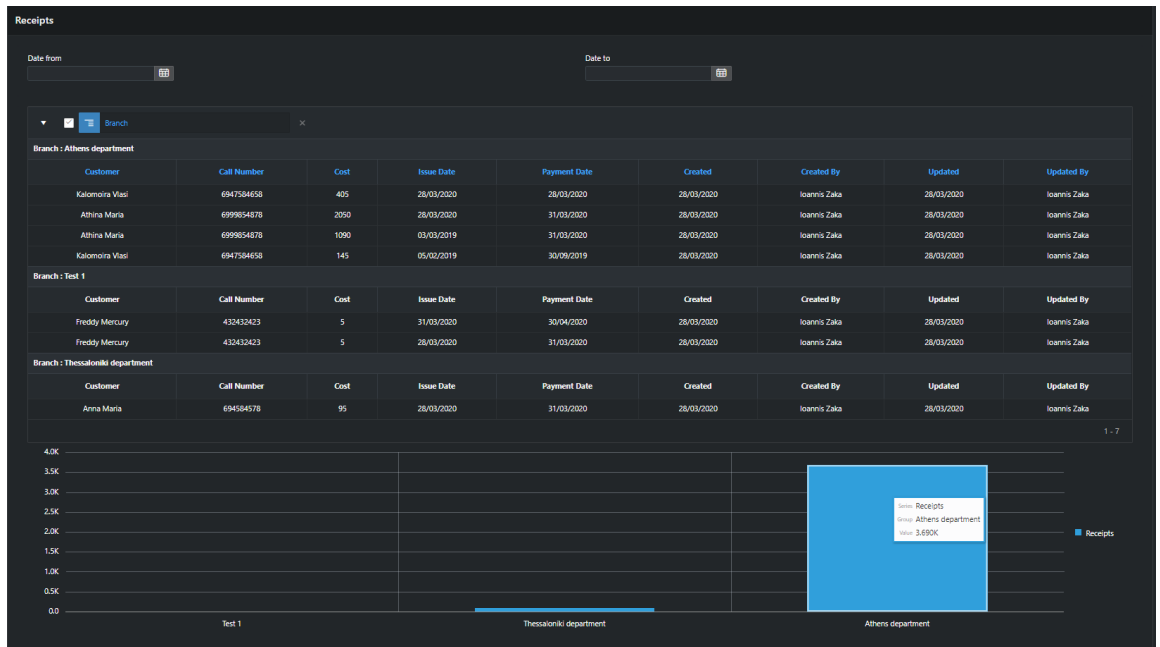
Η κεντρική σελίδα αποτελείται από διάφορα regions τα οποία δείχνουν συνοπτικές πληροφορίες με βάση κάποια φίλτρα. Αυτά τα Regions είναι:

- Συμβόλαια, τα οποία μπορούμε να τα φιλτράρουμε ανά Branch(κατάστημα) και ανά ημερομηνίες (Date from – Date to)



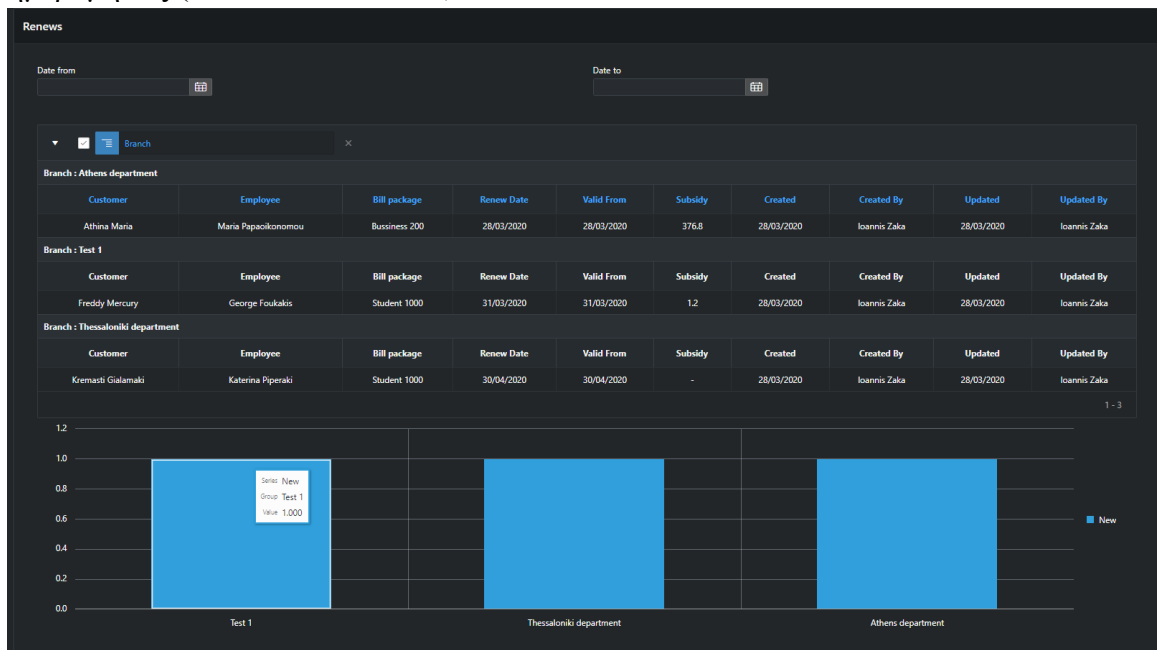
IMGS-71

- Λογαριασμοί, τα οποία μπορούμε να τα φιλτράρουμε ανά ημερομηνίες (Date from – Date to)



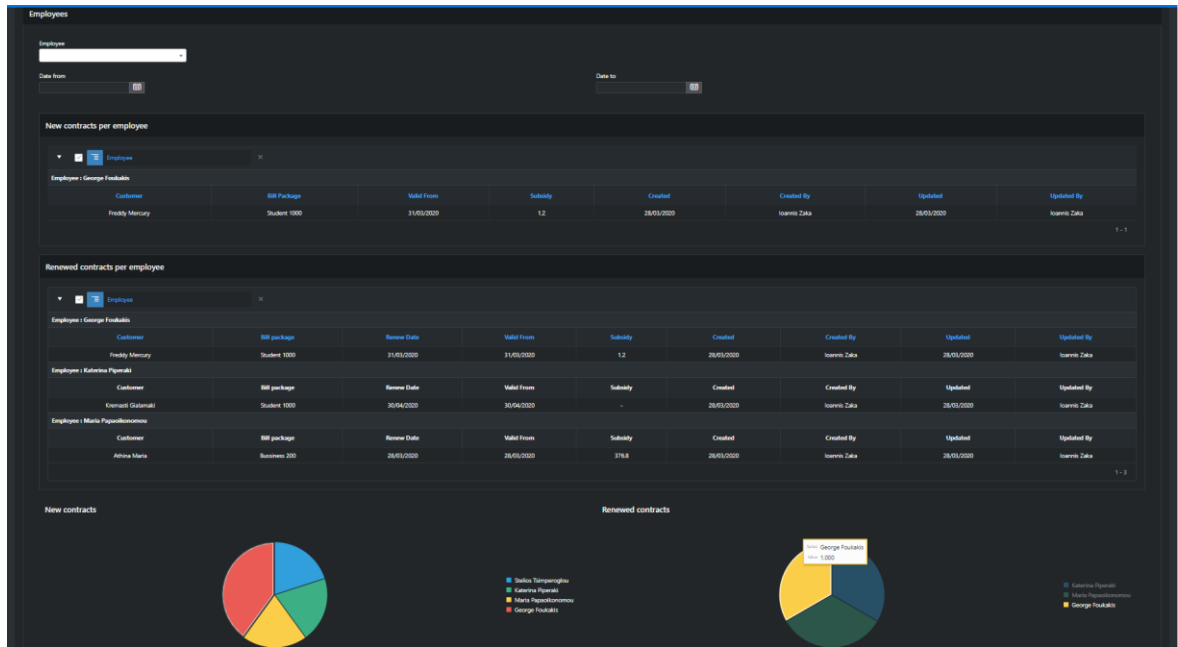
IMGS-72

c. Ανανεώσεις λογαριασμών τα οποία μπορούμε να φιλτράρουμε ανά ημερομηνίες (Date from – Date to)



IMGS-73

d. Νέα συμβόλαια και ανανεώσεις συμβολαίων που έκαναν οι εργαζομένοι τα οποία μπορούμε να τα φιλτράρουμε με βάση τους Εργαζομένους και ανά ημερομηνίες (Date from – Date to)



IMGS-74

5. Αποτέλεσμα

5.1 Συμπέρασμα

Ο στόχος της εφαρμογής μας έχει ολοκληρωθεί με βάση τις απαιτήσεις που μας έχουν παραδοθεί. Πλέον η εφαρμογή μας μπορεί να αποθηκεύει σημαντική πληροφορία αλλά και να την επεξεργαστεί. Ο κύριος λόγος που αποφάσισα να αναπτύξω την εφαρμογή μου σε APEX Oracle είναι επειδή το APEX προσφέρει την πιο γρήγορη και αξιόπιστη επικοινωνία ανάμεσα σε βάση δεδομένων (backend) με το Frontend κομμάτι. Οι μεταφορά των δεδομένων γίνεται άμεσα, άρα και γρήγορα και ταυτόχρονα με ασφάλεια σε σύγκριση με άλλες τεχνολογίες που χρειάζονται ενδιάμεσες εφαρμογές όπως Nodejs κ.α. Αυτό έχει ως αποτέλεσμα να είναι μια εφαρμογή που βολεύει και καλύπτει τις βασικές ανάγκες των χρηστών της ώστε η καθημερινή ρουτίνα τους να είναι πιο απλή κι πιο εύκολη δίχως ταλαιπωρίες.

Πιο συγκεκριμένα, η αναγκαίες πληροφορίες που έπρεπε να αποθηκευτούν είναι οι εξής:

- Για κάθε υποκατάστημα της εταιρίας κινητής τηλεφωνίας το όνομα και την διεύθυνση
- Για κάθε υπάλληλο της εταιρίας κινητής τηλεφωνίας το όνομα, το επώνυμο, τον αριθμό ταυτότητας, τον μισθό, τα προστατευόμενα μέλη της οικογένειας τους (ονοματεπώνυμο, φύλο και ηλικία). Οι υπάλληλοι χωρίζονται σε δύο κατηγορίες
 - Σε διοικητικούς για τους οποίους αποθηκεύομαι πληροφορία για το πτυχίο τους (τίτλο και πανεπιστήμιο που το έδωσε και ημερομηνία κτήσης)
 - Σε τεχνικούς για τους οποίους αποθηκεύομαι πληροφορία για την ειδικότητα και τα έτη προϋπηρεσίας
- Για τις κεραίες που έχει η εταιρία κινητής τηλεφωνίας αποθηκεύομαι το όνομα, την διεύθυνση και τον τύπο της κεραίας
- Για κάθε πελάτη της εταιρίας κινητής τηλεφωνίας το όνομα, την διεύθυνση, τον αριθμό ταυτότητας και το επάγγελμα
- Για κάθε αριθμό που έχει η εταιρία αποθηκεύομαι τον αριθμό κλήσης, τον κωδικό αριθμό (ruk1 και 2) και αν είναι σύνδεση ή καρτοκινητό
- Για τα διάφορα πακέτα χρέωσης που υπάρχει αποθηκεύουμε πληροφορίες όπως το όνομα, τα λεπτά της δωρεάν ομιλίας και τα μηνύματα, το πάγιο και την τιμή χρέωση για τον παραπάνω χρόνο ομιλίας και το ίδιο για τα μηνύματα
- Για κάθε βλάβη/επιδιόρθωση που λαμβάνει χώρα στο δίκτυο της εταιρίας αποθηκεύομαι το πότε έγινε, σε ποια κεραία, τι είδους ήταν και ποιοι υπάλληλοι την επιδιορθώσανε

- Για κάθε λογαριασμό αποθηκεύομαι το ποσό χρέωσης, την ημερομηνία έκδοσης, την ημερομηνία πληρωμής και σε ποιον αριθμό και πελάτη αντιστοιχεί

Επίσης σαν περιορισμούς έχουμε τις εξής πληροφορίες:

- Ένας πελάτης ανήκει σε ένα και μόνο ένα υποκατάστημα.
- Ένα υποκατάστημα έχει τουλάχιστον 2 υπαλλήλους
- Ένα υποκατάστημα έχει ένα και μόνο ένα υπάλληλο ως διευθυντή και ένας υπάλληλος είναι το πολύ σε ένα υποκατάστημα διευθυντής
- Ένας αριθμός συνδέεται με ένα το πολύ πελάτη αλλά ένας πελάτης μπορεί να έχει πολλούς αριθμούς
- Ένας υπάλληλος μπορεί να επιδιορθώνει πολλές κεραίες και μια κεραία μπορεί να επιδιορθώνεται από πολλούς υπαλλήλους
- Μια κεραία ανήκει σε ένα και μόνο υποκατάστημα και ένα υποκατάστημα μπορεί να έχει πολλές κεραίες
- Ένας πελάτης που έχει σύνδεση έχει υπογράψει ένα και μόνο ένα συμβόλαιο με κάποιον υπάλληλο της εταιρίας και αυτό θα ανήκει σε ένα από τα πακέτα σύνδεσης

Οι διαδικασίες που υποστηρίζει η εφαρμογή μας είναι:

- Εισαγωγή και διαγραφή από τις πιο πάνω κατηγορίες
- Interface για την εισαγωγή του χρόνου ομιλίας και των μηνυμάτων που έγινε με την χρήση κάποιου αριθμού οπότε μετά θα υπολογίζεται το ποσόν χρέωσης ως
 - $\text{Ποσόν} = \text{πάγιο} + [(\text{λεπτά_ομιλίας} - \text{δωρεάν_λεπτά}) / 60] * \text{extra} + [\text{αριθμός_μηνυμάτων} - \text{αριθμός_δωρεάν_μηνυμάτων}] * \text{χρε_μηνυμάτος}$
- Αυτόματη μείωση του παγίου κατά 10% όταν συμπληρωθεί ένας χρόνος σύνδεσης
- Για κάθε ανανέωση συμβολαίου θα υπολογίζεται το ποσόν της επιδότησης και θα το εμφανίζει στην οθόνη. Το ποσό της επιδότησης προκύπτει ως εξής
 - $\text{Ποσόν_επιδότησης} = \text{άθροισμα_λογαριασμών_τελευταίου_χρόνου} * 0.12$
- Αναφορά συγκεντρωτική περιέχοντας τα συμβόλαια κι τα καρτοκινητά που κάνει καθ'ένα υποκατάστημα
- Συγκεντρωτική αναφορά για τις εισπράξεις κάθε καταστήματος σε ένα χρονικό διάστημα
- Συγκεντρωτική αναφορά για τις ανανεώσεις κάθε καταστήματος σε ένα χρονικό διάστημα
- Συγκεντρωτική αναφορά για κάθε υπάλληλο όσον αφορά
 - τα νέα συμβόλαια που κάνει
 - τις ανανεώσεις

5.2 Επεκτασιμότητα και εργασία για το μέλλον

Όπως κάθε εφαρμογή έτσι και η δική μας έχει την δυνατότητα και συνίσταται να επεκταθεί και να βελτιωθεί. Μερικά παραδείγματα αναλύονται παρακάτω:

- Επέκταση εφαρμογής σε «Πολλαπλές εταιρίες»

Δηλαδή, να υπάρξει η δυνατότητα να μπορούν να χρησιμοποιούν την εφαρμογή πάνω από μία εταιρίες αλλά ταυτόχρονα να μην επηρεάζει η μία την άλλη. Η κάθε εταιρία θα βλέπει πάντα τα δικά της στοιχεία και μόνο αυτά

- Αυτοματοποίηση εφαρμογής στην «Καταγραφή χρήσης και κοστολόγηση»

Έτσι όπως λειτουργεί αυτήν την στιγμή η εφαρμογή θα πρέπει να μπου οι εργαζόμενοι και να εισάγουν με το χέρι τα μηνύματα και τα λεπτά ομιλίας ανά πελάτη, αριθμό και ημερομηνία. Θα μπορούσε να φτιαχτεί μια αυτόματη καταγραφή η οποία δεν παραβιάζει τα προσωπικά δεδομένα των χρηστών και καταγράφει την καθημερινή χρήση των χρηστών.

- Επέκταση εφαρμογής σε «Διαφορετικούς τύπους εταιριών»

Η εφαρμογή μας έχει στηθεί για μία εταιρία κινητής τηλεφωνίας αλλά μπορεί κάλλιστα να τροποποιηθεί με πολύ εύκολο τρόπο και να την χρησιμοποιήσει οποιοδήποτε τύπου εταιρίας ενδιαφέρεται για την χρήση της.

Πίνακας των εικόνων

IMG-1	Business Priorities	Σελίδα 8
IMG-2	Why businesses need to go digital	Σελίδα 8
IMG-3	ORACLE LOGO	Σελίδα 9
IMG-4	APEX LOGO	Σελίδα 10
IMG-5	APACHE TOMCAT LOGO	Σελίδα 11
IMG-6	ORDS LOGO	Σελίδα 12
IMG-7	SQL DEVELOPER LOGO	Σελίδα 12
IMG-8	DOCKER DESKTOP LOGO	Σελίδα 13
IMG-9	HTML LOGO	Σελίδα 14
IMG-10	JAVASCRIPT LOGO	Σελίδα 14
IMG-11	<u>jQuery</u> LOGO	Σελίδα 15
IMG-12	CSS LOGO	Σελίδα 16
IMG-13	PL/SQL LOGO	Σελίδα 17
IMG-14	ΣΤΟΧΟΙ ΚΑΙ ΕΝΕΡΓΕΙΕΣ ΕΙΚΟΝΑ	Σελίδα 18
IMG-15	DATABASE SCHEMA	Σελίδα 20
IMG-16	LOGIN PAGE	Σελίδα 21
IMG-17	MENU BRANCH IMG1	Σελίδα 22
IMG-18	MENU CUSTOMER IMG2	Σελίδα 23

IMG-19	MENU ADMINISTRATION IMG3	Σελίδα 24
IMG-20	USERS REPORT	Σελίδα 24
IMG-21	USERS EDIT	Σελίδα 25
IMG-22	ROLES REPORT	Σελίδα 25
IMG-23	ROLES EDIT	Σελίδα 25
IMG-24	SPECIALITIES REPORT	Σελίδα 26
IMG-25	SPECIALITIES EDIT	Σελίδα 26
IMG-26	ANTENNA TYPES REPORT	Σελίδα 27
IMG-27	ANTENNA TYPES EDIT	Σελίδα 27
IMG-28	REPAIR TYPES REPORT	Σελίδα 27
IMG-29	REPAIR TYPES EDIT	Σελίδα 28
IMG-30	CUSTOMERS REPORT	Σελίδα 28
IMG-31	CUSTOMERS EDIT	Σελίδα 28
IMG-32	CUSTOMERS – BRANCH LIST	Σελίδα 29
IMG-33	TELEPHONE NUMBERS REPORT	Σελίδα 29
IMG-34	TELEPHONE NUMBERS EDIT	Σελίδα 30
IMG-35	TELEPHONE NUMBERS – OWNER LIST	Σελίδα 30
IMG-36	BILLING PACKAGES REPORT	Σελίδα 31
IMG-37	BILLING PACKAGES EDIT	Σελίδα 31

IMG-38	BILLS REPORT	Σελίδα 31
IMG-39	BILLS EDIT	Σελίδα 32
IMG-40	BILLS – NUMBER LIST	Σελίδα 32
IMG-41	CONTRACTS REPORT	Σελίδα 33
IMG-42	CONTRACTS EDIT	Σελίδα 33
IMG-43	CONTRACTS - CUSTOMER LIST	Σελίδα 33
IMG-44	CONTRACTS – EMPLOYEE LIST	Σελίδα 34
IMG-45	CONTRACTS – BILLING PACKAGE LIST	Σελίδα 34
IMG-46	CONTRACTS – FILE UPLOAD	Σελίδα 35
IMG-47	USAGE PER NUMBER REPORT	Σελίδα 35
IMG-48	BRANCHES REPORT	Σελίδα 36
IMG-49	BRANCHES EDIT	Σελίδα 36
IMG-50	BRANCHES – ADMINISTRATIVE LIST	Σελίδα 36
IMG-51	BRANCHES EMPLOYEES	Σελίδα 37
IMG-52	BRANCHES EMPLOYEES LIST	Σελίδα 37
IMG-53	ANTENNAS REPORT	Σελίδα 38
IMG-54	ANTENNAS EDIT	Σελίδα 38
IMG-55	ANTENNAS – TYPE LIST	Σελίδα 39
IMG-56	ANTENNAS – BRANCH LIST	Σελίδα 40

IMG-57	EMPLOYEES REPORT	Σελίδα 40
IMG-58	EMPLOYEES EDIT	Σελίδα 41
IMG-59	EMPLOYEES – TYPE LIST	Σελίδα 41
IMG-60	TECHNITIAN DETAILS	Σελίδα 42
IMG-61	TECHNITIAN – SPECIALTY LIST	Σελίδα 42
IMG-62	ADMINISTRATIVE DETAILS	Σελίδα 43
IMG-63	PROTECTED MEMBERS	Σελίδα 43
IMG-64	PROTECTED MEMBERS – GENDER LIST	Σελίδα 44
IMG-65	REPAIRS REPORT	Σελίδα 44
IMG-66	REPAIRS EDIT	Σελίδα 44
IMG-67	REPAIRS – ANTENNA LIST	Σελίδα 45
IMG-68	REPAIRS – REPAIR TYPE LIST	Σελίδα 45
IMG-69	REPAIRS – EMPLOYEE LIST	Σελίδα 45
IMG-70	HOME	Σελίδα 46
IMG-71	CONTRACTS	Σελίδα 46
IMG-72	BILLS	Σελίδα 47
IMG-73	CONTRACT RENEWS	Σελίδα 47
IMG-74	NEW CONTRACTS	Σελίδα 49

Βιβλιογραφία

Bryla, Ioney, B.B., .K.L. (2013). Oracle Database 12c The Complete Reference - September 2013. (978-0-07-180175-1 ed.). India: McGraw-Hill Osborne Media.

Ian Abramson, Michael S. Abbey, Michael Corey, I.A. ., .M.S.A., .M.C. (2004). Oracle Database 10g: A Beginner's Guide. New York Chicago San Francisco Lisbon London Madrid Mexico City Milan New Delhi San Juan Seoul Singapore Sydney Toronto: McGraw-Hill/Osborne.

Marcel van der Plas, Michel van Zoest, M.P., .M.Z. (2013). Oracle APEX Cookbook: Second Edition. (2nd ed.). London: Packt Publishing Ltd.

John Scottnick, Buytaertkaren, Cannellmartin, d'souzadoug, Gaultdimitri, Gielisroel, Hartmandenes, Kubicekraj, Mattamaldan, Mcghanfrancis, Mignaulttom, Petrusjorge, Rimblaschristoph, Ruepprich,

J.S., N.B., K.C., M.D.S., D.G., D.G., R.H., D.K., R.M., D.M., F.M., T.P., J.R., C.R. (2015). Oracle REST Data Services. New York: Apress, Berkeley, CA - Online ISBN 978-1-4842-0484-9 - Print ISBN 978-1-4842-0485-6.

Charles Anderson, C.A. (2015). Docker [Software engineering]. : IEEE Software , Print ISSN: 0740-7459 Electronic ISSN: 1937-4194.

D Merkel, D.M. (2014). Docker: lightweight linux containers for consistent development and deployment. : Linux journal.

Chanoch Wiggers, Ben Galbraith, Vivek Chopra, Sing Li, Debashish Bhattacharjee, Amit Bakore, Romin Irani, Sandip Bhattacharya, Chad Fowler, C.W., .B.G., .V.C., .S.L., .D.B., .A.B., .R.I., .S.B., .C.F. (2004). Professional Apache Tomcat. (1st ed.). Indianapolis, IN 46256: Wiley Publishing, Inc 10475 Crosspoint Boulevard - ISBN: 0-7645-4372-5.

David Sawyer McFarland, D.S.M. (2011). JavaScript & JQuery: The Missing Manual. (2nd ed.). United States of America: O'Reilly Media, Inc, 1005 Gravenstein Highway North, Sebastopol, CA 95472 - ISBN: 978-4-449-3-9902-3.

Danny Goodman, D.G. (2002). Dynamic HTML: The Definitive Reference: A Comprehensive Resource for HTML, CSS, DOM & JavaScript. (2nd ed.). United States of America: O'Reilly Media, Inc, 1005 Gravenstein Highway North, Sebastopol, CA 95472 - ISBN: 0-596-00316-1.

Steven Feuerstein, Andrew Odewahn, S.F. .A.O. (2000). Oracle PL/SQL Programming: A Developer's Workbook: Oracle Development Languages. (1st ed.). United States of

America: O'Reilly Media, Inc, 1005 Gravenstein Highway North, Sebastopol, CA 95472 - ISBN: 1-56592-674-9.

Andrew odewahn, A.O. (1999). Oracle Web Applications; PL/SQL Developer's Introduction September 1999. (1st ed.). United States: O'Reilly & Associates, Inc 103A Morris St Sebastopol, CA United States- SBN:978-1-56592-687-5.

Wc johnson, a sirikit, W.C.J. .A.S. (2002). Service quality in the Thai telecommunication industry: a tool for achieving a sustainable competitive advantage. (1st ed.). : .

References

{1} Cyrille modiano, C.M. (2017). DBA-SCRIPTS. Retrieved 7 May, 2020, from <https://www.dba-scripts.com/articles/oracle-internals/oracle-rest-data-services/>

{2} Computer hope, C.H. (c2020). Computer Hope. Retrieved 7 May, 2020, from <https://www.computerhope.com/jargon/h/html.htm>

{3} Mazhar imam khan, puneetjain97, M.I.K.,.P.J. (c2020). GeeksforGeeks. Retrieved 7 May, 2020, from <https://www.geeksforgeeks.org/plsql-introduction/>

{4} Margaret rouse, M.R. (c2020). SearchOracle. Retrieved 7 May, 2020, from <https://searchoracle.techtarget.com/definition/PL/SQL>

{5} Red hat, R.H. (c2019). OpenSource. Retrieved 7 May, 2020, from <https://opensource.com/resources/what-docker>

{6} Connecticut siting council, C.S.C. (2019). Data. Retrieved 7 May, 2020, from <https://data.ct.gov/Government/Telecommunications-Towers-and-Antennas/n7zh-5dbr>

{7} John papiewski, J.P. (c2020). SmallBusiness. Retrieved 7 May, 2020, from <https://smallbusiness.chron.com/types-antennas-telecommunication-32506.html>

Παράρτημα Πτυχιακής εργασίας

```
create or replace PACKAGE BODY TELE_AUTH AS

Procedure Login_user(p_username in VARCHAR2 , p_password in VARCHAR2) AS
Cursor fetchUser is
Select *
from tel_users
where Upper(username)= Upper(p_username)
and Upper(password)=Upper(p_password);

userRec tel_users%Rowtype;
BEGIN
Open fetchUser;
Fetch fetchUser Into lUserRec;
Close fetchUser;

INSERT into xxx_logger(err_msg) values(' user_id '||lUserRec.Id||' appl_username '||lUserRec.Username);

If lUserRec.Id Is Null Then
apex_util.set_authentication_result(1);
Raise_application_error(-20096,'Invalid Username');
-- elsif lUserRec.is_approved = -1 Then
-- apex_util.set_authentication_result(1);
-- Raise_application_error(-20096,'User not approved yet');
else
APEX_UTIL.SET_SESSION_STATE('APPL_USER_ID',lUserRec.Id);
APEX_UTIL.SET_SESSION_STATE('APPL_USERNAME',lUserRec.Username);
APEX_UTIL.SET_SESSION_STATE('APPL_ROLE_ID',lUserRec.Role_id);
end if;
END Login_user;

FUNCTION Login (p_username IN VARCHAR2, p_password IN VARCHAR2) Return Boolean AS
Begin
Login_user (UPPER (p_username), UPPER(p_password));
Return TRUE;
Exception
When Others Then Return FALSE;
END Login;

Function Register_User (p_firstname in VARCHAR, p_lastname in VARCHAR2, p_username in VARCHAR2,p_email in Varchar2, p_password in Varchar2, p_confirm_pass in Varchar2 ) return Varchar2
as
result varchar2(4000) := '1';
lexist number;
Begin
select count(1) into lexisit from tel_users where username= p_username or email=p_email;

insert into xxx_logger(err_msg)values('p_firstname: '||p_firstname||' p_lastname: '||p_lastname||' p_username:'||p_username||' p_email'||p_email||' p_password '||p_password||' p_confirm

if lexisit = 0 then

if p_firstname is not null and /*p_firstname != '' and*/
p_lastname is not null and /*p_lastname != '' and*/
p_username is not null and /*p_username != '' and*/
p_email is not null and /*p_email != '' and*/
p_password is not null and /*p_password != '' and*/
p_confirm_pass is not null /*and p_confirm_pass != ''*/
then
insert into xxx_logger(err_msg)values('perasa thn id me ta nulls');
if Upper(p_password) = Upper(p_confirm_pass) then

insert into xxx_logger(err_msg)values('perasa to if me ta pass');
if instr(p_email,'@') != 0 then
insert into xxx_logger(err_msg)values('perasa to if me to email');
begin
insert into tel_users
(username,password,lastname,firstname,email)
values
(p_username,p_password,p_lastname,p_firstname,p_email);
exception when others then
result := sqlerrm;
end;
else
result := 'Invalid Email';
end if;
elsif Upper(p_password) != Upper(p_confirm_pass) then
```

```

p_lastname is not null and /'p_lastname != '' and'/
p_username is not null and /'p_username != '' and'/
p_email is not null and /'p_email != '' and'/
p_password is not null and /'p_password != '' and'/
p_confirm_pass is not null /'and p_confirm_pass != ''/
then
insert into xxx_logger(err_msg)values('perasa thn id me ta nulls');
if Upper(p_password) = Upper(p_confirm_pass) then

insert into xxx_logger(err_msg)values('perasa to if me ta pass');
if instr(p_email,'@') != 0 then
insert into xxx_logger(err_msg)values('perasa to if me to email');
begin
insert into tel_users
values
(p_username,p_password,p_lastname,p_firstname,p_email);
exception when others then
result := sqlerrm;
end;
else
result := 'Invalid Email';
end if;
elseif Upper(p_password) != Upper(p_confirm_pass) then
result := 'Password and confirm password do not match';

end if;
end if;
else
result := 'Username or email already exist';
end if;
return result;
End Register_User;
*/
Function FormatDate (n_email in Varchar2) return Varchar2 as

```

```

create or replace PACKAGE BODY TELE_CALC AS

function cr_bill_cost(numberID in number,issueDate in Date) return number AS
maxDate date;
sMinutes number;
sMessages number;
fMinutes number;
fMessages number;
cPMinute number;
cPMessage number;
cFixed number;
pCustomer number;
totalCost number;

contractDATE date;
finalMinutes number;
finalMessages number;

BEGIN
--Get last issue date
begin
select issue_date into maxDate from
(SELECT issue_date
FROM tel_bills WHERE number_id = numberID and issue_date < issueDate
order by issue_date desc)
where rownum=1;
exception when no_data_found then
maxDate := to_date('01/01/1900','dd/mm/yyyy');
end;

--Get total minutes since last issue date
begin
select sum(minutes) into sMinutes
from tel_minutes_talked
where number_id = numberID
and date_of_use > maxDate
and date_of_use <= issueDate;
exception when no data found then

```

```

and date_of_use <= issueDate;
exception when no_data_found then
    sMinutes := 0;
end;

--Get total messages since last issue date
begin
    select sum(messages) into sMessages
    from tel_minutes_talked
    where number_id = numberID
    and date_of_use > maxDate;
exception when no_data_found then
    sMessages := 0;
end;

--Get the customer
select owner_id
into pCustomer
from tel_number_details
where id = numberID;

--Get the billing packages values that consist of this number
select a.FREE_MINUTES,a.FREE_MESSAGES,a.FIXED,a.COST_MINUTES,a.COST_MESSAGES,b.valid_from
into fMinutes,fMessages,cFixed,cPMinute,cPMessage,contractDate
from tel_billing_packages a, tel_contracts b
where b.customer_id = pCustomer
and b.bill_pckg_id = a.id;

totalCost := 0;
finalMinutes := sMinutes - fMinutes;
finalMessages:= sMessages- fMessages;

--If minutes used exceed the free minutes then calculate the cost of the extra minutes used
if finalMinutes > 0 then
    totalCost := totalCost + (finalMinutes * cPMinute );
end if;
--If messages used exceed the free messages then calculate the cost of the extra messages used
if finalMessages > 0 then
    totalCost := totalCost + (finalMessages * cPMessage);

```

```

    if finalMessages > 0 then
        totalCost := totalCost + (finalMessages * cPMessage);
    end if;
    --Add the fixed price into the total cost
    cFixed := calc_fixed(cFixed,contractDate);
    totalCost := totalCost + cFixed;

    return totalCost;
END cr_bill_cost;

function calc_fixed(fixedCost in Number,contractDate in Date) return number as
pYears number;
pFixedCost number := fixedCost;
begin
    begin
        select trunc(months_between(sysdate,contractDate)/12) into pYears from dual;
    exception when others then
        pYears := 0;
    end;

    for i in 1..pYears loop
        pFixedCost := pFixedCost - (pFixedCost * 0.1);
    end loop;

    return pFixedCost;
end;

function calc_subs(customerID in Number, lValidFrom date) return number as
tCost number;
begin
    insert into xxx_logger(err_msg) values ('meta tin function');
    select sum(cost) into tCost
    from tel_bills
    where number_id in (select nd.id
                        from tel_number_details nd
                        where owner_id = customerID)
    and issue_date between
        add months(trunc(lValidFrom, 'month'). -12)

```

```

begin
insert into xxx_logger(err_msg) values ('meta tin function');
select sum(cost) into tCost
from tel_bills
where number_id in (select nd.id
                    from tel_number_details nd
                    where owner_id = customerID)
and issue_date between
    add_months(trunc(lValidFrom, 'month'), -12)
and
    trunc(lValidFrom+30, 'month');
insert into xxx_logger(err_msg) values ('ekana select auto: ' || tCost);
tCost := tCost * 0.12;

return tCost;

end;
END TELE_CALC;

```

```

create or replace package body TELE_Utils is
    ApacheUrlDocs Varchar2(1024) := 'http://db2.ap.t.gr:8080/ords/project_files/';
    -----
    Function fGetTokenizerString( pString      In Varchar2 ,
                                pToken       In Varchar2,
                                pAddNullValues in Boolean default False )

    Return StringArray Is
    lStringArray StringArray;
    lremString   Varchar2(32000) := pString;
    IsLast       Boolean := False;
    IndexArray   Binary_Integer := 0;
    Procedure pAddString ( pString In Varchar2 ) Is
    Begin
        If ( not pAddNullValues ) Then
            If pString Is Null Then Return; End If;
        End if;
        IndexArray := IndexArray +1;
        lStringArray(IndexArray) := pString;
    End pAddString;
    Begin
    If pString Is Null --Or Nvl(Length(pToken),2) > 1
    Then Return lStringArray;
    End If;
    ---
    If (pString Is Not Null And Instr(pString,pToken) = 0 ) Then
        pAddString(pString);
        Return lStringArray;
    End If;

    Loop
    If Not IsLast
    Then
        pAddString(Substr(lremString,1,instr(lremString,pToken)-1));
        lremString := Substr(lremString,instr(lremString,pToken)+length(pToken));
    Else
        pAddString(lremString);
        Exit;
    End If;
    If Nvl(Instr(lremString,pToken),0) = 0 Then IsLast := True;

```

```

        End If;
    If Nvl(Instr(lremString,pToken),0) = 0 Then IsLast := True;
    End If;
End Loop;

Return lStringArray;
End fGetTokenizerString;

Function fConstructFileName(pTabAlias In Varchar2 ,
                           pId      In Number  ,
                           pFileName In Varchar2
                           ) Return Varchar2 Is
Begin
    Return lower(pTabAlias)||'_'||pId||'.'||
           Substr(pFileName,Instr(pFileName,'.',-1)+1);
End fConstructFileName;
-----
Function fSaveBlobToFile ( pBlobFile In Blob , pFileName In Varchar2 ) Return Number Is
l_file UTL_FILE.FILE_TYPE;
l_buffer RAW(32767);
l_amount BINARY_INTEGER := 32767;
l_pos INTEGER := 1;
l_blob_len INTEGER;

Begin

    -- Get LOB locator
    l_blob_len := DBMS_LOB.getlength(pBlobFile);
    l_pos:= 1;
    -- Open the destination file.
    l_file := UTL_FILE.fopen('PROJECT_FILES',pFileName,'wb', 32767);

    -- Read chunks of the BLOB and write them to the file
    -- until complete.
    WHILE l_pos < l_blob_len LOOP
    DBMS_LOB.read(pBlobFile, l_amount, l_pos, l_buffer);
    UTL_FILE.put_raw(l_file, l_buffer, TRUE);
    l_pos := l_pos + l_amount;
    END LOOP.

```

```

END LOOP;

-- Close the file.
UTL_FILE.fclose(l_file);

Return l;

Exception When Others Then
Dbms_Output.put_line(Sqlerrm);
-- Close the file if something goes wrong.
IF UTL_FILE.is_open(l_file) THEN
    UTL_FILE.fclose(l_file);
END IF;
raise_application_error(-20666,Sqlerrm);
Return 0;
End fSaveBlobToFile;
-----

▶ ⊞ Procedure pDeleteFile ( pFileName In Varchar2 ) Is
Begin
    utl_file.fremove('PROJECT_FILES',pFileName);
Exception When Others Then
    Dbms_Output.put_line(Sqlerrm);
End pDeleteFile;

-----

▶ ⊞ Function getFilename(pFilePath In Varchar2) Return Varchar2 Is
    v_file VARCHAR2(100);
Begin
    -- Parse string for UNIX system
    ⊞ If INSTR(pFilePath, '/') > 0 then
        v_file := SUBSTR(pFilePath,
            (INSTR(pFilePath, '/', -1, 1) + 1),
            length(pFilePath));
        -- Parse string for Windows system
    elsif INSTR(pFilePath, '\') > 0 then
        v_file := SUBSTR(pFilePath,
            (INSTR(pFilePath, '\', -1, 1) + 1),
            length(pFilePath));
        -- If no slashes were found, return the original string

```

```

length(pFilePath));
    -- If no slashes were found, return the original string
else
    v_file := pFilePath;
end if;
Return v_file;
End getFilename;

--
FUNCTION ClearSting (pVal in VARCHAR2) RETURN VARCHAR2
IS
    v_val VARCHAR2(4000);
BEGIN
    Select Replace(Translate(pVal,
                            '!@#%&^*()+={ } | \ : ; , . < > ? / ~ ` " ' ,
                            '#####'),
                '^',
                ' ')
        into v_val
    from Dual;
    return (v_val);
END;
--
FUNCTION getIRSql (
    p_app_id      IN   NUMBER,
    p_page_id     IN   NUMBER,
    p_all_cols    IN   BOOLEAN DEFAULT True
)
RETURN VARCHAR2
IS
    v_report_id   NUMBER;
    v_region_id   NUMBER;
    v_report      apex_ir.t_report;
    v_query       VARCHAR2 (32767);
    v_column      VARCHAR2 (4000);
    v_position    NUMBER;

BEGIN
    SELECT region_id
    -----

```



```

BEGIN
SELECT region_id
  INTO v_region_id
  FROM apex_application_page_regions
 WHERE application_id = p_app_id
       AND page_id = p_page_id
       AND source_type = 'Interactive Report';

v_report_id :=
  apex_ir.get_last_viewed_report_id (p_page_id      => p_page_id,
                                    p_region_id     => v_region_id
                                   );

v_report :=
  apex_ir.get_report (p_page_id      => p_page_id,
                    p_region_id     => v_region_id,
                    p_report_id     => v_report_id
                   );
v_query := v_report.sql_query;

FOR i IN 1 .. v_report.binds.COUNT
LOOP
  v_query :=
    REPLACE (v_query,
            ':' || v_report.binds (i).NAME,
            '|| v_report.binds (i).VALUE || '|| '
           );
END LOOP;

IF p_all_cols
THEN
  FOR c IN (SELECT *
            FROM apex_application_page_ir_col
            WHERE application_id = p_app_id AND page_id = p_page_id
            ORDER BY display_order)
  LOOP
    v_column := v_column || ', ' || c.column_alias;
  END LOOP;

  v_column := LTRIM (v_column, ', ');

```

```

        v_column := LTRIM (v_column, ', ');
        v_position := INSTR (v_query, '(');
        v_query := SUBSTR (v_query, v_position);
        v_query := 'SELECT ' || v_column || ' FROM ' || v_query;
    END IF;

    RETURN v_query;
EXCEPTION
    WHEN OTHERS
    THEN
        v_query := SQLERRM;
        RETURN v_query;
END getIRSql;
-----
Function fCommaToArray(pCommaStr In Varchar2,pSeperator In Varchar2 Default ',') Return strarray Is
lArr strarray := strarray();
lSt StringArray;
Begin
    If pCommaStr Is Null
    Then Return lArr;
    Elself Instr(pCommaStr,pSeperator) = 0
    Then
        lArr.Extend;
        lArr(lArr.Count) := pCommaStr;
    Else
        lSt := fGetTokenizerString(pCommaStr,pSeperator);
        For x In 1..lSt.Count
        Loop
            lArr.Extend;
            lArr(lArr.Count) := lSt(x);
        End Loop;
    End If;
    Return lArr;
End fCommaToArray;
-----
create or replace TRIGGER BD_TEL_BRANCH_EMP
BEFORE DELETE ON TEL_BRANCH_EMP
for each row
declare
bCOUNT number;
BEGIN
    begin
        select emp_no into bCOUNT from tel_branches where id = :new.BRANCH_ID;
    exception when others then
        bCount := 0;
    end;
    if bCount <= 2 then
        raise_application_error(-20001,'Records can not be deleted');
    else
        null;
    end if;

END;
```

```

create or replace TRIGGER BD_TEL_CONTRACTS
BEFORE DELETE ON TEL_CONTRACTS
for each row
BEGIN
    delete from tel_upload_files where component_id = 1 and ref_id = :new.id;
END;

```

```

create or replace TRIGGER BIU_TEL_BRANCH_EMP
BEFORE INSERT OR UPDATE ON TEL_BRANCH_EMP
for each row
declare
bCOUNT number;
BEGIN
    IF INSERTING THEN
        IF :new.ID IS NULL THEN
            SELECT SEQ_BRANCH_EMP .NEXTVAL INTO :new.ID FROM dual;
        END IF;
        :new.CREATED := SYSDATE;
        :new.CREATED_BY := nvl(v('APPL_USER_ID'),1);
        :new.UPDATED := SYSDATE;
        :new.UPDATED_BY := nvl(v('APPL_USER_ID'),1);
        begin
            select emp_no into bCOUNT from tel_branches where id = :new.BRANCH_ID;
        exception when others then
            bCount := 0;
        end;
        bCount := bCount + 1;
        begin
            update tel_branches set emp_no = bCOUNT where id = :new.BRANCH_ID;
        exception when others then
            insert into XXX_LOGGER (err_msg) values ('something went wrong with emp_no count');
        end;

        ELSIF UPDATING THEN
            :new.UPDATED := SYSDATE;
            :new.UPDATED_BY := nvl(v('APPL_USER_ID'),1);
        END IF;
    END;

```

```

create or replace TRIGGER BIU_TEL_CONTRACTS
BEFORE INSERT OR UPDATE OR DELETE ON TEL_CONTRACTS
for each row
declare
tCost number;
BEGIN
  IF INSERTING THEN
    IF :new.ID IS NULL THEN
      SELECT SEQ_CONTRACTS.NEXTVAL INTO :new.ID FROM dual;
    END IF;
    :new.CREATED := SYSDATE;
    :new.CREATED_BY := nvl(v('APPL_USER_ID'),1);
    :new.UPDATED := SYSDATE;
    :new.UPDATED_BY := nvl(v('APPL_USER_ID'),1);

    ELSIF UPDATING THEN
      :new.UPDATED := SYSDATE;
      :new.UPDATED_BY := nvl(v('APPL_USER_ID'),1);
    if :old.valid_from != :new.valid_from then
      :new.SUBSIDY := tele_calc.calc_subs(:new.CUSTOMER_ID, :new.VALID_FROM);
      insert into tel_renews (contract_id,renew_date)
      values (:new.id,:new.valid_from);
    end if;
  END IF;
END;

```