



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΨΗΦΙΑΚΗ ΕΥΣΤΡΑΤΕΙΑ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΕΥΝΑΣ ΣΤΑ ΔΙΚΤΥΟΚΕΝΤΡΑ ΕΥΣΤΡΑΤΕΙΑ

PRIVATE BANKING

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μαρία Α. Καραναυγιάτου

2009

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα: ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
Κατεύθυνση: ΔΙΚΤΥΟΚΕΝΤΡΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ PRIVATE BANKING



Καθηγητής: ΒΑΣΙΛΑΚΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ

Φοιτήτρια: ΚΑΡΑΠΑΝΑΓΙΩΤΟΥ ΜΑΡΙΑ ΜΕ/0463

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ	
ΑΡ. ΕΙΣ.	594 18 + CD
COMP.	41029
ΤΑΞΗ	332.1 ΚΑΡ
ΒΙΒΛΙΟΘΗΚΗ	



ΠΕΡΙΛΗΨΗ

Σύμφωνα με την αρχιτεκτονική SOA, στη συγκεκριμένη διπλωματική εργασία περιγράφονται τρία process model, κάθε ένα από τα οποία λειτουργεί αυτόνομα και διεκπεραιώνει μία ολοκληρωμένη επιχειρησιακή διαδικασία (business process) της διεύθυνσης Private Banking μιας τράπεζας. Ένα business process αποτελείται από επιμέρους δραστηριότητες (Activities), η υλοποίηση των οποίων γίνεται από ανεξάρτητα προγράμματα, ενώ η επικοινωνία τους επιτυγχάνεται μέσω containers. Ο συντεταγμένος και προτυποποιημένος τρόπος επικοινωνίας μεταξύ των επιχειρησιακών διαδικασιών τις καθιστά επαναχρησιμοποιήσιμες και ανεξάρτητες από τις εφαρμογές και τις υπολογιστικές πλατφόρμες στις οποίες λειτουργούν.

Πανεπιστήμιο Πειραιώς

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία είναι το αποτέλεσμα της εμπειρίας που αποκόμισα κατά τη διάρκεια της φοίτησής μου στο πρόγραμμα μεταπτυχιακών σπουδών (κατεύθυνση Δικτυοκεντικά Συστήματα) του τμήματος Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιά.

Για τον λόγο αυτό θα ήθελα να ευχαριστήσω όλους τους καθηγητές που στάθηκαν δίπλα μου και μου δώσανε όλες τις απαραίτητες γνώσεις. Θα ήθελα να ευχαριστήσω από τα βάθη της καρδιάς μου τον Καθηγητή κ. Γεώργιο Βασιλακόπουλο, για τη μεγάλη προσφορά του, στην κατανόηση και επίλυση των πολλών δυσκολιών και προβλημάτων που αντιμετώπισα.

Τέλος εκφράζω την ευγνωμοσύνη μου στην οικογένειά μου για την υποστήριξη και βοήθειά τους σε όλη τη διάρκεια των μεταπτυχιακών σπουδών μου.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	2
ΕΥΧΑΡΙΣΤΙΕΣ	3
ΠΕΡΙΕΧΟΜΕΝΑ	4
ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ	5
ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ	6
ΚΕΦΑΛΑΙΟ 1 ^ο	7
1.1. ΕΙΣΑΓΩΓΗ	8
1.2. ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ	9
ΚΕΦΑΛΑΙΟ 2 ^ο ΣΥΣΤΗΜΑ ΡΟΗΣ ΕΡΓΑΣΙΑΣ WORKFLOW	10
2.1. ΕΙΣΑΓΩΓΗ	11
2.2. CONFIGURATION TOOL	13
2.3. WEBSHERE MQ WORKFLOW BUILDTIME	13
2.3.1. PROCESS MODEL ΠΡΟΣΕΛΚΥΣΗΣ ΠΕΛΑΤΕΙΑΣ PRIVATE BANKING	14
2.3.2. PROCESS MODEL ΑΠΟΔΟΧΗ ΠΕΛΑΤΗ PRIVATE BANKING	18
2.3.3. PROCESS MODEL ΕΝΗΜΕΡΩΣΗΣ & ΕΞΥΠΗΡΕΤΗΣΗΣ ΠΕΛΑΤΕΙΑΣ P.B.	26
2.3.4. STAFF – ΠΡΟΣΩΠΙΚΟ	29
2.3.5. ROLES – ΡΟΛΟΙ	29
2.3.6. ORGANIZATIONS – ΟΡΓΑΝΙΣΜΟΙ	30
2.3.7. PROGRAMS – ΠΡΟΓΡΑΜΜΑΤΑ	30
2.3.8. DATA STRUCTURE – ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ	32
2.3.9. EXPORT	33
2.4 WEBSHERE MQ WORKFLOW CLIENT	52
ΚΕΦΑΛΑΙΟ 3 ^ο ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	55
3.1. ΕΙΣΑΓΩΓΗ	56
3.2. ΔΙΑΧΕΙΡΙΣΗ ΡΟΛΩΝ	57
3.2.1. ΜΟΝΤΕΛΟ ΟΝΤΟΤΗΤΩΝ ΣΥΣΧΕΤΙΣΕΩΝ	58
3.2.2. ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ	60
3.3. ΔΙΑΔΙΚΑΣΙΑ ΠΡΟΣΕΛΚΥΣΗΣ ΠΕΛΑΤΕΙΑΣ PRIVATE BANKING	63
3.3.1. ΜΟΝΤΕΛΟ ΟΝΤΟΤΗΤΩΝ ΣΥΣΧΕΤΙΣΕΩΝ	65
3.3.2. ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ	69
3.4. ΑΠΟΔΟΧΗ ΠΕΛΑΤΗ PRIVATE BANKING	73
3.4.1. ΜΟΝΤΕΛΟ ΟΝΤΟΤΗΤΩΝ ΣΥΣΧΕΤΙΣΕΩΝ	75
3.4.2. ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ	83
3.5. ΕΝΗΜΕΡΩΣΗ & ΕΞΥΠΗΡΕΤΗΣΗ ΠΕΛΑΤΕΙΑΣ PRIVATE BANKING	91
3.5.1. ΜΟΝΤΕΛΟ ΟΝΤΟΤΗΤΩΝ ΣΥΣΧΕΤΙΣΕΩΝ	92
3.5.2. ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ	94
ΚΕΦΑΛΑΙΟ 4 ^ο ΑΝΑΛΥΣΗ JAVA ΚΩΔΙΚΑ	97
4.1. ΕΙΣΑΓΩΓΗ	98
4.2. ΣΥΝΔΕΣΗ ΜΕ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ORACLE 9I	98
4.3. ΔΗΜΙΟΥΡΓΙΑ ΠΙΝΑΚΑ (JTABLE)	100
4.4. CONTAINERS	101
ΚΕΦΑΛΑΙΟ 5 ^ο ΣΥΜΠΕΡΑΣΜΑΤΑ	103
5.1. ΕΙΣΑΓΩΓΗ	104
5.2. ΑΝΑΣΚΟΠΗΣΗ	104
ΒΙΒΛΙΟΓΡΑΦΙΑ	105
ΠΑΡΑΡΤΗΜΑ Α	106
ΠΕΡΙΓΡΑΦΗ ΛΟΓΙΣΜΙΚΟΥ	106

ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Διαδικασία Προσέλκυσης πελατείας.....	14
Σχήμα 2: Διαδικασία Αποδοχής Πελάτη	18
Σχήμα 3: Διαδικασία Ενημέρωσης & Εξυπηρέτησης Πελατείας Private Banking	26
Σχήμα 4: Μοντέλο Οντοτήτων – Συσχετίσεων των ρόλων μιας τράπεζας.....	58
Σχήμα 5: Σχεσιακό Σχήμα ρόλων μιας τράπεζας	60
Σχήμα 6: Μοντέλο Οντοτήτων – Συσχετίσεων Διαδικασίας Προσέλκυσης Πελατείας (τμήμα 1 ^ο).....	65
Σχήμα 7: Μοντέλο Οντοτήτων – Συσχετίσεων Διαδικασίας Προσέλκυσης Πελατείας (τμήμα 2 ^ο).....	67
Σχήμα 8: Σχεσιακό Σχήμα Διαδικασίας Προσέλκυσης Πελατείας (τμήμα 1 ^ο).....	69
Σχήμα 9: Σχεσιακό Σχήμα Διαδικασίας Προσέλκυσης Πελατείας (τμήμα 2 ^ο).....	70
Σχήμα 10: Μοντέλο Οντοτήτων – Συσχετίσεων Διαδικασίας Αποδοχής Πελάτη (τμήμα 1 ^ο).....	75
Σχήμα 11: Μοντέλο Οντοτήτων – Συσχετίσεων Διαδικασίας Αποδοχής Πελάτη (τμήμα 2 ^ο).....	77
Σχήμα 12: Μοντέλο Οντοτήτων – Συσχετίσεων Διαδικασίας Αποδοχής Πελάτη (τμήμα 3 ^ο).....	79
Σχήμα 13: Μοντέλο Οντοτήτων – Συσχετίσεων Διαδικασίας Αποδοχής Πελάτη (τμήμα 4 ^ο).....	81
Σχήμα 14: Σχεσιακό Σχήμα Διαδικασίας Αποδοχής Πελάτη (τμήμα 1 ^ο).....	83
Σχήμα 15: Σχεσιακό Σχήμα Διαδικασίας Αποδοχής Πελάτη (τμήμα 2 ^ο).....	84
Σχήμα 16: Σχεσιακό Σχήμα Διαδικασίας Αποδοχής Πελάτη (τμήμα 3 ^ο).....	85
Σχήμα 17: Σχεσιακό Σχήμα Διαδικασίας Αποδοχής Πελάτη (τμήμα 4 ^ο).....	86
Σχήμα 18: Μοντέλο Οντοτήτων – Συσχετίσεων Διαδικασίας Ενημέρωσης & Εξυπηρέτησης Πελατείας.....	92
Σχήμα 19: Σχεσιακό Σχήμα Διαδικασίας Ενημέρωσης και Εξυπηρέτησης Πελατείας.....	94

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

Ελληνικές

- ΣΣΠ: Στέλεχος Σχέσεων Πελατών
- ΥΔΛΠ: Υπεύθυνος Διαχείρισης Λογαριασμών Πελάτη
- ΥΜ: Υπεύθυνος Μονάδας
- ΔΕΣ: Διεύθυνση Επενδυτικής Στρατηγικής
- ΗΑΠ: Ηλεκτρονικό Αρχείο Πελάτη
- ΤΜΔΜΠ: Τμήμα Τήρησης & Διαχείρισης Μητρώου Πελατών

Λατινικές

- SOA - Service Oriented Architecture: Ένα πρότυπο αρχιτεκτονικής πληροφορικής που υποστηρίζει τον προσανατολισμό προς τις υπηρεσίες.
- Business Process: Επιχειρησιακή Διαδικασία
- Activity: Επιμέρους τμήμα μιας ολοκληρωμένης επιχειρησιακής διαδικασίας
- Service: Μία επαναλαμβανόμενη επιχειρησιακή διαδικασία, όπως προσέλκυση πελατείας ή αποδοχή πελατείας.
- Containers: Ανεξάρτητα προγράμματα που βοηθούν στην επικοινωνία μεταξύ των δραστηριοτήτων μιας επιχειρησιακής διαδικασίας.
- RM: Relation Manager
- AO: Account Officer
- UH: Unit Head
- IS: Investment Strategy Section
- BCCH: Business Control & Compliance Head
- CIF: Customer Id File
- PB: Private Banking
- CDU: Client Documentation Unit
- MURC: Management Unit of Registration of Customer

1.1. Εισαγωγή

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι να μελετηθεί η επίδραση της διαδικασίας της διαμόρφωσης των προτύπων ποιότητας στην ανάπτυξη της ποιότητας.

- της διαδικασίας διαμόρφωσης προτύπων
- της διαδικασίας διαμόρφωσης προτύπων
- της διαδικασίας διαμόρφωσης προτύπων

Επομένως, η παρούσα διπλωματική εργασία έχει ως σκοπό να μελετήσει την επίδραση της διαδικασίας διαμόρφωσης των προτύπων ποιότητας στην ανάπτυξη της ποιότητας. Η μελέτη αυτή θα πραγματοποιηθεί με τη βοήθεια της μεθόδου της ανάλυσης διαδικασιών (Process Analysis) και της μεθόδου της ανάλυσης των αιτιών (Cause and Effect Analysis).

Αυτή η μελέτη θα πραγματοποιηθεί με τη βοήθεια της μεθόδου της ανάλυσης διαδικασιών (Process Analysis) και της μεθόδου της ανάλυσης των αιτιών (Cause and Effect Analysis). Η μελέτη αυτή θα πραγματοποιηθεί με τη βοήθεια της μεθόδου της ανάλυσης διαδικασιών (Process Analysis) και της μεθόδου της ανάλυσης των αιτιών (Cause and Effect Analysis).

ΚΕΦΑΛΑΙΟ 1^ο

Πανεπιστήμιο Πειραιώς

1.1. Εισαγωγή

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι η πλήρης περιγραφή τριών διαδικασιών της διεύθυνσης Private Banking μιας τράπεζας, συγκεκριμένα:

- της διαδικασίας προσέλκυσης πελατείας,
- της διαδικασίας αποδοχής πελάτη και
- της διαδικασίας ενημέρωσης και εξυπηρέτησης πελατείας.

Επίσης, η εργασία αυτή απεικονίζει επαρκώς τον τρόπο με τον οποίο οι διάφορες οντότητες των τμημάτων της τράπεζας αλληλεπιδρούν μεταξύ τους, για την ανταλλαγή πληροφοριών. Οι λειτουργικές απαιτήσεις της εφαρμογής βασίζονται στα Business Processes που περιγράφονται στο παρόν έγγραφο.

Ακόμη περιγράφεται πλήρως η βάση δεδομένων, που σχεδιάστηκε για τις παραπάνω διαδικασίες της διεύθυνσης Private Banking. Αναφέρονται τα μοντέλα δεδομένων της Βάσης Δεδομένων, τα οποία διακρίνονται στα διαγράμματα οντοτήτων – σχέσεων και στην περιγραφή των πινάκων της σχεσιακής βάσης.

1.2. Ιστορική Αναδρομή

Οι εξελίξεις στον τομέα της Πληροφορικής είναι ραγδαίες. Στην αρχή, υπήρχαν τα μεγάλα υπολογιστικά κέντρα όπου μπορούσαν να εξυπηρετήσουν βασικές εταιρικές διαδικασίες. Στη συνέχεια, με την τεχνολογία client/ server κατάφεραν να μειώσουν σημαντικά το κόστος υποδομής και να καλύψουν σχεδόν κάθε τμήμα μιας εταιρίας. Στις μέρες μας, βρισκόμαστε στην εποχή του Web, της «απόλυτης δικτύωσης» και του «ενιαίου client περιβάλλοντος».

Ταυτόχρονα, όμως με τις ραγδαίες εξελίξεις αυξανόταν και οι απαιτήσεις των εταιριών από την τεχνολογία, όπου διαφορετικά συστήματα, εφαρμογές, δεδομένα και εργαλεία συνθέτουν ένα πολύπλοκο περιβάλλον, το οποίο γίνεται πολυπλοκότερο με την εύκολη δικτύωση που προσέφερε το Internet. Στα προβλήματα της τεχνολογίας προστέθηκαν και αυτά του επιχειρησιακού περιβάλλοντος, όπου οι απαιτήσεις μιας ολόενα παγκόσμιας αγοράς οδήγησαν τους σύγχρονους οργανισμούς στην ανακάλυψη νέων τρόπων για να μεγαλώσουν την επιχείρησή τους και να βελτιώσουν την αποδοτικότητά τους.

Η απαίτηση λοιπόν για άμεσα προσαρμόσιμα και ευέλικτα επιχειρησιακά συστήματα και υποδομές τεχνολογίας που τα υποστηρίζουν οδήγησαν στην προσέγγιση ενός επιπέδου αφαίρεσης (abstraction layer), όπου δεν έχουμε τη δημιουργία ολοκληρωμένων αυτόνομων και ενιαίων συστημάτων, άλλα ένα σύνολο μικρότερων οντοτήτων, που κάθε μία έχει συγκεκριμένο τρόπο επικοινωνίας με οποιαδήποτε άλλη οντότητα. Ο ενιαίος κώδικας πλέον χωρίζεται σε ανεξάρτητες προγραμματιστικές οντότητες που διεκπεραιώνουν ολοκληρωμένες, επί μέρους διαδικασίες. Οι οντότητες αυτές επικοινωνούν με το εξωτερικό περιβάλλον με τυποποιημένο τρόπο, είναι επαναχρησιμοποιήσιμες και δεν επηρεάζονται από την υποδομή στην οποία «τρέχουν».

Η νέα αυτή προσέγγιση επικεντρώνεται στην υπηρεσία (service), και τις απαιτήσεις που αυτή έχει από το περιβάλλον και αντίστροφα. Συνεπώς, δεν κατασκευάζεται πλέον μία εφαρμογή διαχείρισης πελατείας μιας τράπεζας, άλλα διαδικασίες, όπως «Προσέλκυση Πελατείας», «Αποδοχή Πελατών» και «Ενημέρωση και Εξυπηρέτηση Πελατών», που θα μπορούν να λειτουργήσουν είτε για τις ανάγκες μιας τράπεζας, είτε για αυτές ενός επενδυτικού ομίλου.

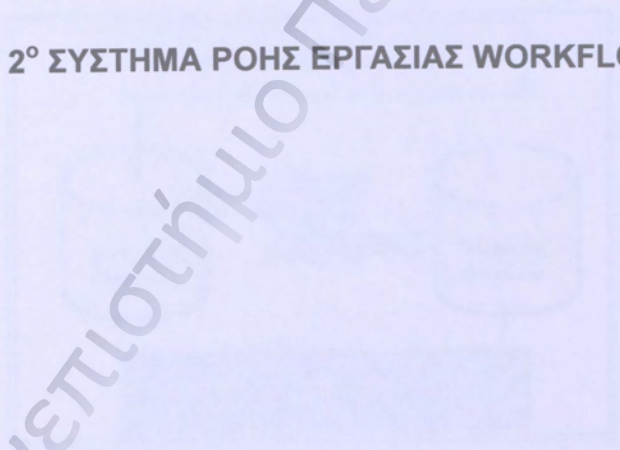
2.1. Εισαγωγή

Η εφαρμογή Oracle Service Bus (OSB) αποτελεί τον κεντρικό κόμβο της Πύλης Εργασίας - Workflow Engine, η οποία παρέχει τη δυνατότητα να συνδέονται οι εφαρμογές με το Workflow. Τα εργαλεία που χρησιμοποιούνται για την ανάπτυξη της OSB είναι:

- WebSphere 9.0 (OSB 11.1.2)
- WebSphere 9.0 (OSB 11.1.2)
- JDK 1.6.0

Η παρούσα μελέτη αφορά στην ανάπτυξη και την εγκατάσταση της OSB, καθώς και στην σύνδεση με το Workflow.

ΚΕΦΑΛΑΙΟ 2^ο ΣΥΣΤΗΜΑ ΡΟΗΣ ΕΡΓΑΣΙΑΣ WORKFLOW



Σύμφωνα με το [Oracle Service Bus](#), η OSB είναι η κεντρική πλατφόρμα που παρέχει την επικοινωνία μεταξύ των εφαρμογών και των υπηρεσιών. Η OSB αποτελεί τον κεντρικό κόμβο της Πύλης Εργασίας - Workflow Engine, η οποία παρέχει τη δυνατότητα να συνδέονται οι εφαρμογές με το Workflow. Τα εργαλεία που χρησιμοποιούνται για την ανάπτυξη της OSB είναι:

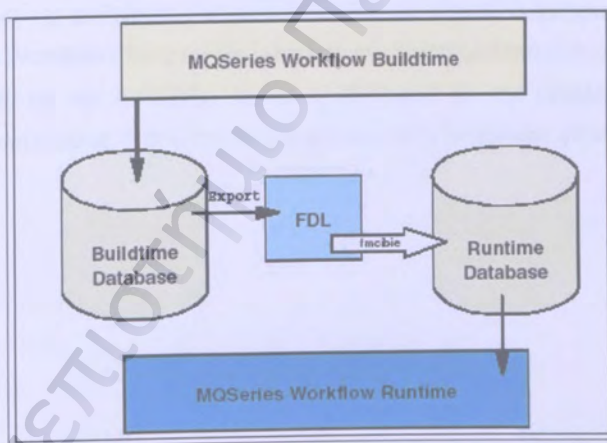
- το OSB
- το OSB
- το OSB
- το OSB

2.1. Εισαγωγή

Η εφαρμογή Private Banking είναι βασισμένη στην τεχνολογία της Ροής Εργασίας - Workflow. Σύμφωνα με αυτήν την τεχνολογία, υλοποιείται πρώτα το μοντέλο Ροής Εργασίας και στη συνέχεια δημιουργείται το πρόγραμμα που θα καθοδηγείται από το Workflow. Τα εργαλεία που χρησιμοποιήθηκαν είναι :

- WebSphere MQ Workflow Buildtime
- WebSphere MQ Workflow Client
- JDK 1.5_03

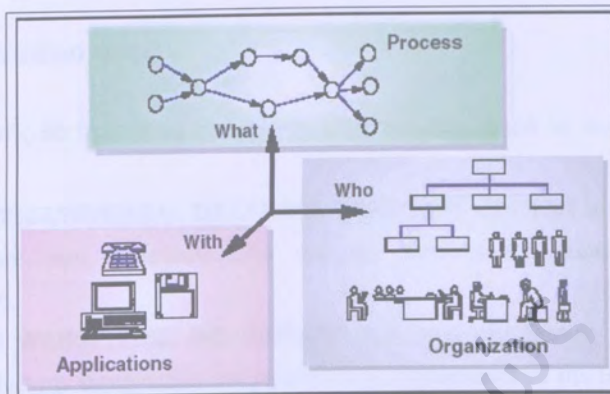
Η σειρά με την οποία γίνεται η επικοινωνία των παραπάνω εργαλείων, φαίνεται στο σχήμα που ακολουθεί :



Εικόνα 1: Επικοινωνία εργαλείων Workflow

Σε πρώτο στάδιο δημιουργείται το μοντέλο δραστηριοτήτων με το MQSeries Workflow Buildtime μέσω του οποίου καθορίζονται τα εξής:

- τα άτομα,
- οι ρόλοι τους,
- τα προγράμματα και
- οι δομές δεδομένων.



Εικόνα 2: Σύνδεση των τμημάτων του Buildtime

Οι παραπάνω ρυθμίσεις αποθηκεύονται στην IBM DB2 βάση δεδομένων του Buildtime. Στη συνέχεια από το Buildtime κάνοντας export δημιουργείται ένα αρχείο με κατάληξη **.fdl** που περιέχει όλες τις ρυθμίσεις του μοντέλου. Για να μπορεί ο χρήστης να ανοίξει τον WebSphere MQ Workflow Client και να εκτελέσει τις δραστηριότητες που του ανήκουν, θα πρέπει το αρχείο με την κατάληξη **.fdl** να γίνει import με την εντολή **fmcibie** και τις απαραίτητες παραμέτρους, έτσι ώστε να ενεργοποιηθεί ο MQSeries Workflow Runtime.

2.2. Configuration tool

Αρχικά, ο χρήστης θα πρέπει να εγκαταστήσει με την εξής σειρά τα ακόλουθα προϊόντα της IBM :

1. Το **IBM DB2 UNIVERSAL DATABASE PERSONAL EDITION** (Αποτελεί την Βάση Δεδομένων που χρησιμοποιείται για την αποθήκευση των δεδομένων του Workflow).
2. Τον **IBM WEBSHERE MQ SERVER** (Αποτελεί τον Server του συστήματος, οποίος δίνει την δυνατότητα στο Workflow να επικοινωνεί με την Βάση του).
3. Το **WEBSHERE MQ WORKFLOW** (Εργαλείο με το οποίο ο χρήστης υλοποιεί το μοντέλο Ροής Εργασίας).

2.3. WebSphere MQ Workflow Buildtime

Μετά την επεξεργασία του Configuration tool, ο χρήστης μπορεί να μπει στο Buildtime του MQSeries Workflow και να κατασκευάσει την διεργασία του μοντέλου (process). Η εισαγωγή του στο πρόγραμμα γίνεται ακολουθώντας την διαδρομή Start → Programs → IBM WebSphere MQ Workflow → WebSphere MQ Workflow Buildtime – FMC. Κατά την είσοδο του στο πρόγραμμα, ο χρήστης θα πρέπει να συμπληρώσει τα πεδία ως ακολούθως:

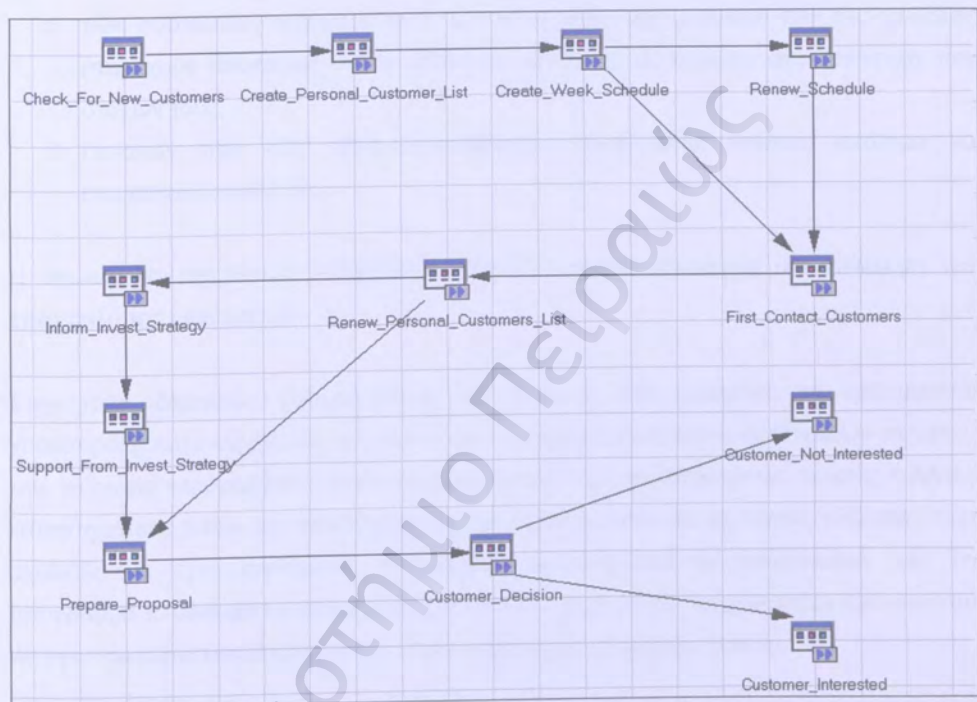
USERNAME = **ADMIN** και
PASSWORD = **password**.

Στο συγκεκριμένο εργαλείο απαιτείται να οριστούν τα εξής :

- **Process Model:** το μοντέλο της διεργασίας
- **Staff:** Το προσωπικό
- **Roles:** Τους ρόλους που θα έχει το προσωπικό
- **Data Structures:** Δομές δεδομένων του μοντέλου
- **Organizations:** Οργανισμοί
- **Programs:** Τα προγράμματα που θα τρέχουν συγχρόνως με τις επιμέρους δραστηριότητες του μοντέλου.

2.3.1. Process Model Προσέλκυσης Πελατείας Private Banking

Σ' αυτή την ενότητα δημιουργείται το Process Model προσέλκυσης πελατείας με τις επιμέρους Activities. Το μοντέλο της εργασίας είναι το ακόλουθο:



Σχήμα 1: Διαδικασία Προσέλκυσης πελατείας

Το παραπάνω μοντέλο αποτελείται από 12 διεργασίες. Αρχικά ο Relationship Manager RM εκτελεί τη διεργασία [Check_For_New_Customers], με την οποία παρακολουθεί ημερησίως τη βάση δεδομένων (Αρχείο Δυνητικών Πελατών), όπου περιλαμβάνονται όλοι οι διαθέσιμοι υποψήφιοι πελάτες (Prospect Clients List) είτε αυτοί είναι φυσικά πρόσωπα είτε νομικά, και επιλέγει με ποιους θα ασχοληθεί.

Στη δεύτερη διεργασία [Create_Personal_Customer_List] ο RM καταρτίζει την προσωπική του λίστα με τους πιθανούς υποψήφιους πελάτες βάσει:

1. της Βάσης Δεδομένων των υποψήφιων πελατών.
2. των απευθείας επαφών του (καλλιέργεια σχέσεων, γεύματα εργασίας, εκδηλώσεις της Τράπεζας, κλπ)

3. των συστάσεων πελατών από υφιστάμενους πελάτες του Private Banking.
4. των συστάσεων από άλλες πηγές της Τράπεζας ή του Ομίλου (Δ/ντες Κατ/των, Corporate Business Centers, Θυγατρικές Ομίλου) για εντελώς νέους πελάτες ή για ήδη υπάρχοντες πελάτες, τους οποίους πρέπει να διαχειρίζεται η Διεύθυνση Private Banking.
5. των συστάσεων πελατών από τον επικεφαλής της μονάδας UH της μονάδας (παρέχουν συστάσεις στους RMs της μονάδας) με έμφαση στην επίτευξη των στόχων τους.
6. Πελατών που ήδη πραγματοποιήθηκε επαφή αλλά κρίνεται σκόπιμο να επαναπροσεγγισθούν.

Η δημιουργία της λίστας υποψηφίων πελατών γίνεται πάντα με την επίβλεψη του επικεφαλής της μονάδας UH.

Στην τρίτη διεργασία [Create_Week_Schedule] ο RM καταρτίζει το εβδομαδιαίο χρονοπρόγραμμα ενεργειών, σύμφωνα με την προσωπική λίστα υποψηφίων πελατών του, το οποίο περιλαμβάνει επισκέψεις στο Δίκτυο (πρώτη επίσκεψη σε πελάτες ή Δ/ντες καταστημάτων, follow up, επισκέψεις, follow up calls, κλπ) και τις λοιπές ενέργειες που σχεδιάζει να πραγματοποιήσει προκειμένου να ενισχύσει το πελατολόγιό του. Το πρόγραμμα περιλαμβάνει επίσης τον χρόνο που εκτιμά ο RM ότι χρειάζεται προκειμένου να προετοιμαστεί κατάλληλα για την παρουσίαση στον εκάστοτε πελάτη.

Στην τέταρτη διεργασία [Renew_Schedule] ο επικεφαλής του τμήματος UH πριν ελέγξει και επικυρώσει το πρόγραμμα, συζητά με τον RM με στόχο την αναθεώρηση και προτεραιοποίηση των ενεργειών του.

Στο επόμενο στάδιο [First_Contact_Customer], ο RM προετοιμάζει την πρώτη επικοινωνία με τον υποψήφιο πελάτη σύμφωνα με το χρονοπρόγραμμα ενεργειών του. Αφού διενεργήσει την πρώτη επικοινωνία (τηλεφωνικώς ή μέσω διερευνητικής συνάντησης) με τον υποψήφιο πελάτη συμπληρώνει το αντίστοιχο call report. Εάν πραγματοποιηθεί συνάντηση, ο RM παρουσιάζει φάκελο με πλήρη ενημερωτικά έγγραφα για το Private Banking καθώς και επιπλέον ενημερωτικά φυλλάδια για τις υπηρεσίες και τα προϊόντα της Τράπεζας. (Παρουσίαση Total Wealth Management). Ο RM κατά την πρώτη επαφή διερευνά τις ανάγκες του πελάτη οργανώνοντας την με βάση τις ακόλουθες τέσσερις περιοχές:

1. Τους στόχους του πελάτη (client objectives).
2. Τις συναλλακτικές ανάγκες του (transactional needs).
3. Τις πιστωτικές ανάγκες του (credit needs).
4. Τις επενδυτικές ανάγκες του (investment needs).

Συμπληρώνει το ερωτηματολόγιο προσδιορισμού επενδυτικό προφίλ. (investment questionnaire).

Βάσει της πρώτης επικοινωνίας/ επαφής με τον υποψήφιο πελάτη, ο RM προτεραιοποιεί τη λίστα υποψήφίων πελατών του [Renew_Personal_Customers_List], καταλήγοντας στους περισσότερο πιθανούς πελάτες (μέσα στο επόμενο τρίμηνο) και συντονίζει τις επόμενες συναντήσεις μαζί τους. Έτσι, μετά την ολοκλήρωση αυτής της δραστηριότητας ο RM έχει καταρτίσει την τελική λίστα υποψήφίων πελατών.

Στη συνέχεια, εάν κριθεί απαραίτητο, ο RM ενημερώνει τη Διεύθυνση Επενδυτικής Στρατηγικής (Investment Strategy Section) [Inform_Invest_Strategy] για τις επενδυτικές προτιμήσεις των υποψήφίων πελατών με σκοπό την προετοιμασία και αποστολή προτάσεων.

Σημείωση: Στην περίπτωση που ο πελάτης – κατά την πρώτη επικοινωνία/ επαφή – έχει δηλώσει συγκεκριμένη ανάγκη, ο RM προσκαλεί και εξειδικευμένα στελέχη της Τράπεζας (Αναλυτές, Dealers, Διαχειριστές αμοιβαίων κεφαλαίων, κλπ). Συγκεκριμένα, οι RMs συνεργάζονται κατά αυτόν τον τρόπο με τη Διεύθυνση Επενδυτικής Στρατηγικής με σκοπό την καλύτερη εξυπηρέτηση του πελάτη.

Στη διεργασία [Support_From_Invest_Strategy] η Διεύθυνση Επενδυτικής Στρατηγικής, υποστηρίζει τους RMs στην προετοιμασία επενδυτικών προτάσεων για τους υποψήφιους πελάτες παρέχοντας εξειδικευμένη γνώση και επενδυτική άποψη με παρουσιάσεις προϊόντων ή ακόμη και συμμετέχοντας στη συνάντηση με τον πελάτη.

Μετά την προαιρετική συνεργασία με την Διεύθυνση επενδυτικής Στρατηγικής και την παραλαβή όλου του σχετικού υλικού από αυτήν, ο RM προετοιμάζει την επενδυτική πρόταση που θα παρουσιάσει στον πελάτη, την οποία καταχωρεί στο σύστημα [Prepare_Proposal].

Στην διεργασία [Customer_Decision] ο RM παρουσιάζει την επενδυτική πρόταση/ σεις της διεύθυνσης Private Banking στον υποψήφιο πελάτη. Εφόσον δεν έχει πραγματοποιηθεί συνάντηση (άλλα μόνο τηλεφωνική επικοινωνία) με τον υποψήφιο πελάτη, ο RM παρουσιάζει τον Φάκελο με τα ενημερωτικά έγγραφα για το Private Banking καθώς και τα επιπλέον ενημερωτικά φυλλάδια για τις υπηρεσίες και τα προϊόντα της Τράπεζας. Γενικά οι RMs οφείλουν:


1. Να είναι άρτια προετοιμασμένοι σχετικά με το ποιος είναι ο πελάτης (με όσα στοιχεία είναι δημοσίως γνωστά ή με όσα στοιχεία μπορεί να παρέχει η πηγή της σύστασης του πελάτη).
2. Να γνωρίζουν πάρα πολύ καλά το σύνολο των προϊόντων και υπηρεσιών που είναι σε θέση να προσφέρει η Τράπεζα και να είναι σε θέση να προσφέρουν τις επιλογές/ λύσεις που θα ικανοποιούν το επενδυτικό προφίλ των πελατών τους.

Σημείωση: Στη συνάντηση με τον πελάτη ανάλογα με την περίπτωση είναι παρόντες δύο RMs ή ένας RM και ο UH της μονάδας.

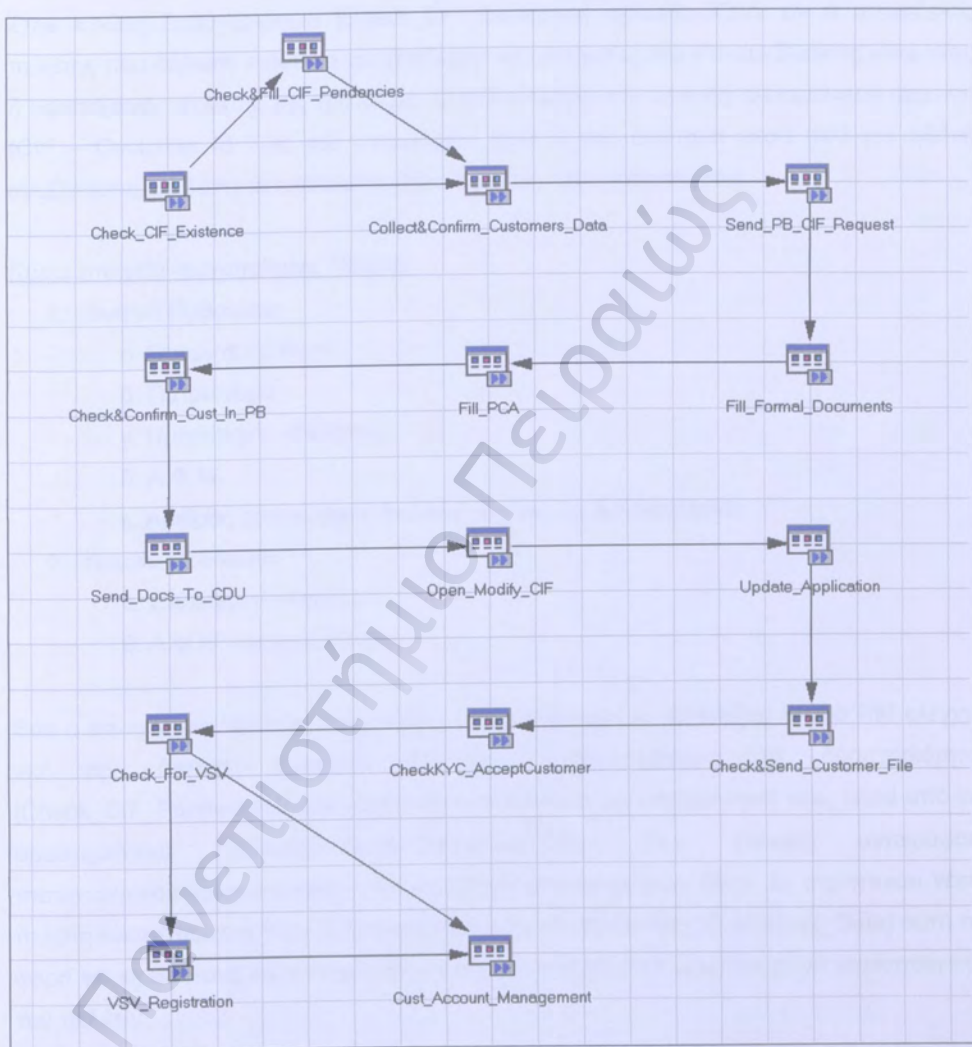
Μετά την παρουσίαση ο υποψήφιος πελάτης αποφασίζει αν θα αποδεχθεί ή όχι τις υπηρεσίες του Private Banking. Στην περίπτωση που ο πελάτης ενδιαφέρεται να αποδεχθεί τις υπηρεσίες, ο RM ενεργοποιεί την διεργασία [Customer_Interested], ενώ σε αντίθετη περίπτωση ενεργοποιεί την διεργασία [Customer_Not_Interested].

[Customer_Interested] Σε αυτήν την διεργασία ο πελάτης ενδιαφέρεται για την επενδυτική πρόταση της μονάδας Private Banking, ο RM συμπληρώνει τα αντίστοιχα συμβατικά έγγραφα και ενεργοποιεί την διαδικασία αποδοχής νέου πελάτη Private Banking.

[Customer_Not_Interested] Σε αυτήν την περίπτωση ο πελάτης δεν ενδιαφέρεται για την επενδυτική πρόταση που του έγινε. Ο RM ενημερώνει το ηλεκτρονικό αρχείο call report, ενώ ο πελάτης παραμένει στην προσωπική λίστα πιθανών πελατών του RM για το χρονικό διάστημα ενός τριμήνου με στόχο την μελλοντική προσέγγισή του.

Σημείωση: Το εικονίδιο  αποτελεί Program Activity και υποδηλώνει ανεξάρτητη δραστηριότητα. Ενώ με τα βέλη παρουσιάζεται η ροή εργασίας που υπάρχει μεταξύ των δραστηριοτήτων.

2.3.2. Process Model Αποδοχή Πελάτη Private Banking



Σχήμα 2: Διαδικασία Αποδοχής Πελάτη

Στο Σχήμα 2 αυτής της ενότητας παρουσιάζεται το Process Model αποδοχής πελάτη με τις επιμέρους Activities. Το παραπάνω μοντέλο αποτελείται από 15 δραστηριότητες.

Στην πρώτη δραστηριότητα [Check_CIF_Existance] προσδιορίζεται αν ο υποψήφιος πελάτης που δήλωσε επιθυμία να αποδεχθεί τις υπηρεσίες του Private Banking είναι νέος ή υφιστάμενος πελάτης της τράπεζας. Ο RM αναζητά την ύπαρξη ηλεκτρονικού αρχείου (CIF – Customer Id File) του υποψήφιου πελάτη στο σύστημα μέσα από μία οθόνη αναζήτησης, με βάση τα ακόλουθα κύρια στοιχεία ταυτοποίησής του.

Κύρια στοιχεία ταυτοποίησης Πελάτη

1. Φυσικά Πρόσωπα
 - α. Ονοματεπώνυμο
 - β. Πατρώνυμο
 - γ. Ημερομηνία γέννησης
 - δ. Α.Φ.Μ.
 - ε. Αριθμός αστυνομικού δελτίου ταυτότητας ή διαβατηρίου
2. Νομικά Πρόσωπα
 - α. Επωνυμία εταιρείας
 - β. Α.Φ.Μ. της εταιρείας

Εάν ο υποψήφιος πελάτης είναι υφιστάμενος πελάτης της Τράπεζας τότε ο RM ελέγχει για την ύπαρξη ελλিপών στοιχείων ενεργοποιώντας τη δραστηριότητα [Check_CIF_Pendencies] και προβαίνει στη συλλογή και καταχώρησή τους μέσα από τη δραστηριότητα [Collect&Confirm_Customers_Data] (π.χ. έλλειψη αντιγράφου πιστοποιητικού ταυτοποίησης, VSV, σύμβαση χαρτοφυλακίου κλπ). Σε περίπτωση νέου πελάτη ενεργοποιείται πάλι η δραστηριότητα [Collect&Confirm_Customers_Data] αυτή τη φορά για τη συλλογή και καταχώρηση των εξής προβλεπόμενων στοιχείων ταυτοποίησης του πελάτη:

Φυσικά Πρόσωπα

- Φωτοτυπία του αστυνομικού δελτίου ταυτότητας ή διαβατηρίου
- Ληξιαρχική πράξη γέννησης (μόνον για ανήλικους)
- Υπεύθυνη Δήλωση των γονέων περί γονικής μέριμνας (μόνον για ανήλικους)
- Βεβαίωση μόνιμης κατοικίας στο εξωτερικό (μόνο για κατοίκους εξωτερικού)

- Υπεύθυνη Δήλωση σχετικά με στοιχεία που αφορούν το παγκόσμιο φορολογικό εισόδημα (μόνον για κατοίκους εξωτερικού)

Νομικά Πρόσωπα

- Τα απαραίτητα νομιμοποιητικά έγγραφα για την έκδοση γνωμάτευσης της εταιρείας από το Legal Documentation Department (LDD).

Σημείωση 1: Στην περίπτωση υφιστάμενου πελάτη, ο RM παραλαμβάνει τα στοιχεία που λείπουν από τον φάκελο του πελάτη.

Σημείωση 2: Στην περίπτωση που δεν προσκομισθεί βεβαίωση μόνιμης κατοικίας, για κατοίκους εξωτερικού, τότε καταχωρείται προειδοποίηση.

Και στις δύο περιπτώσεις ο RM αναγνωρίζει τον πελάτη βάσει των στοιχείων ταυτοποίησης, συσχετίζοντας τα στοιχεία με την φυσική παρουσία του πελάτη.

Η δραστηριότητα [Send_PB_CIF_Request] είναι η επόμενη που ενεργοποιείται. Ο RM πριν αποστείλει ηλεκτρονικά την αίτηση ανοίγματος ή τροποποίησης CIF_PB (για φυσικά ή νομικά πρόσωπα), την εκτιπώνει και εξασφαλίζει την υπογραφή του πελάτη εντός του σχετικού πλαισίου, παραδίδοντάς του τη σύμβαση Γενικών Όρων Συναλλαγών (Γ.Ο.Σ.). Στην περίπτωση εταιρειών (offshore) παραλαμβάνει την Υπεύθυνη Δήλωση του εκπροσώπου (Beneficial Owner) της εταιρείας, που ελέγχει άμεσα ή έμμεσα την εταιρεία:

Φυσικά Πρόσωπα

- Έντυπο CIF Private Banking (CIF PB)_για Φυσικό Πρόσωπο

Νομικά Πρόσωπα

- Έντυπο CIF Private Banking (CIF PB)_για Νομικό Πρόσωπο
- Υπεύθυνη Δήλωση του εκπροσώπου (Beneficial Owner) της εταιρείας, που ελέγχει άμεσα ή έμμεσα την εταιρεία.

Σημείωση: Οι νόμιμοι εκπρόσωποι (Beneficial Owners) εταιρείας offshore πρέπει να έχουν CIF στην Τράπεζα. Στην περίπτωση που ο νόμιμος εκπρόσωπος δεν έχει CIF στην τράπεζα, παραδίδει στον RM το στοιχείο ταυτοποίησης του και συμπληρώνει και υπογράφει το CIF PB_για Φυσικό Πρόσωπο, για να ανοιχτεί το CIF στο σύστημα.

Στη δραστηριότητα [Fill_Formal_Documents], ο RM επιλέγει ηλεκτρονικά μέσα από μία λίστα (checkbox list) την συμπλήρωση των υπόλοιπων προβλεπόμενων έγγραφων του Private Banking και των εντύπων παροχής υπηρεσιών (προαιρετικά) και αφού τα εκτυπώσει, παραλαμβάνει την υπογραφή του πελάτη, ενώ διενεργεί τον έλεγχο του όμοιου της υπογραφής (SV) σε υφιστάμενο πελάτη στα υπογεγραμμένα έγγραφα όπου απαιτείται.

Προβλεπόμενα έγγραφα Private Banking

1. *Επιστολή Αποδοχής των Υπηρεσιών του Private Banking (PBAL – Private Banking Acceptance Letter).*
2. *Σύμβαση Χαρτοφυλακίου και Πληρεξουσιότητας.*
3. *Έντυπο W-8BEN.* Η συμπλήρωση και υπογραφή του είναι υποχρεωτική για νέους πελάτες της Τράπεζας. Είναι απαραίτητο να έχουν υπογράψει όλοι οι συν-δικαιούχοι του χαρτοφυλακίου το W8BEN για να προβούν σε αγορές Αμερικάνικων χρεογράφων και όχι μόνο ο Primary δικαιούχος.
4. *Ερωτηματολόγιο Προσδιορισμού Προφίλ Επενδυτικού Κινδύνου (IRQ - Investment Risk Questionnaire).* Συμπληρώνεται στην ειδική εφαρμογή για όλους τους νέους πελάτες Private Banking (Execution Only) χωρίς υπογραφή του πελάτη. Στην περίπτωση που ο πελάτης επιθυμεί Παροχή Υπηρεσιών Διάρθρωσης Χαρτοφυλακίου (Premium Advisory Services), τότε το IRQ συμπληρώνεται και υπογράφεται από τον πελάτη και τα εξουσιοδοτημένα στελέχη του PB. Σε κάθε περίπτωση παραδίδεται αντίγραφο στον πελάτη.

Έντυπα παροχής υπηρεσιών

1. *Σύμβαση Παροχής Υπηρεσιών Διάρθρωσης Χαρτοφυλακίου (PAS - Premium Advisory Services).*
2. *Σύμβαση Hold Mail..*
3. *Αίτηση Σύμβαση PB Cash Card* (μόνο για Φυσικά Πρόσωπα).
4. *Αίτηση Υπηρεσιών E-Banking* (μόνο για Φυσικά Πρόσωπα).

Στη δραστηριότητα [Fill_PCA], ο RM συμπληρώνει το έντυπο Prospect Client Approval Form μέσα από την αντίστοιχη οθόνη του συστήματος (PCA_New Client ή PCA_Existing Client) με όλους τους συν-δικαιούχους του client group (πρώτος δικαιούχος και συν-δικαιούχος/ οι), καταγράφει τα σχετικά σχόλια, εκτυπώνει και υπογράφει το έντυπο. Εάν ο υποψήφιος πελάτης είναι υφιστάμενος πελάτης άλλης μονάδας Τράπεζας, αποστέλλει με

fax το εσωτερικής χρήσης έντυπο PB acceptance letter στον Διευθυντή της αρμόδιας Μονάδας (Καταστήματος ή Διεύθυνσης) για την επικύρωση της ένταξης του πελάτη στο PB. Ο Διευθυντής αφού καταχωρήσει συστάσεις ή/ και σχόλια σχετικά με τη συνεργασία του υποψήφιου πελάτη με το κατάστημα/ Μονάδα ευθύνης του, το υπογράφει και το στέλνει πίσω.

Στη δραστηριότητα [Check&Confirm_Cust_In_PB], ο UH αφού παραλάβει τα έγγραφα του φυσικού αρχείου του υποψήφιου πελάτη ελέγχει τις καταχωρήσεις του RM ως προ την πληρότητα και ορθότητα, επίσης ελέγχει την ύπαρξη SV στις υπογραφές του πελάτη (ονοματεπώνυμο και υπογραφή του RM). Στη συνέχεια επιβεβαιώνει την τήρηση των κανόνων KYC και της πρόληψης έναντι του ξεπλύματος χρήματος. Στην περίπτωση που δεν υπάρχουν εκκρεμότητες ως προς την πληρότητα και εγκυρότητα ο UH εγκρίνει το PCA συμπληρώνοντας τυχόν σχόλια και παραδίδει στον Account Officer τον ελεγμένο φάκελο του υποψήφιου πελάτη για αποστολή στο τμήμα CDU

Σημείωση: Σε περίπτωση ύπαρξης εκκρεμοτήτων σημειώνει τη σχετική εκκρεμότητα καθώς και η προβλεπόμενη ημερομηνία τακτοποίησής της, στο σχετικό πεδίο που υπάρχει στο έντυπο PCA. Ο UH μεριμνά για την εξάλειψη των εκκρεμοτήτων στην περίπτωση έλλειψης στοιχείων στο φυσικό και ηλεκτρονικό αρχείο των πελατών με βάση την κατάσταση ελλείψεων – εκκρεμοτήτων που αποστέλλει περιοδικά το τμήμα CDU.

Στη διαδικασία [Send_Docs_To_CDU] ο ΑΟ στέλνει ηλεκτρονικά στο τμήμα CDU τις αιτήσεις δημιουργίας ή τροποποίησης CIF, εκείνων των υποψήφιων πελατών, των οποίων τα PCAs έχουν εγκριθεί από τον επικεφαλής της μονάδας UH, ενώ με εσωτερική αλληλογραφία αποστέλλονται οι φυσικοί τους φάκελοι που περιέχει τα εξής:

1. Τα πρωτότυπα συμβατικά έγγραφα/ έντυπα του φυσικού αρχείου του πελάτη στο
2. Τα αντίγραφα:
 - a. Υπογεγραμμένο έντυπο W-8BEN, αποστέλλεται στη Διεύθυνση Θεματοφυλακής
 - b. Δήλωση Μόνιμης Κατοικίας, αποστέλλεται στη Διεύθυνση Θεματοφυλακής
 - c. Υπεύθυνη Δήλωση των στοιχείων που αφορούν το παγκόσμιο φορολογικό σύστημα, αν υπάρχει, αποστέλλεται στη Διεύθυνση Θεματοφυλακής
3. Τις πρωτότυπες Αιτήσεις/ Συμβάσεις παροχής υπηρεσιών προς τις αρμόδιες Διευθύνσεις/ Θυγατρικές της Τράπεζας.

Στη δραστηριότητα [Open_Modify_CIF] ο υπάλληλος της μονάδας CDU ελέγχει τα έγγραφα που έχουν αποσταλεί ως προς την πληρότητα, ορθότητα της συμπλήρωσης των σχετικών πεδίων και την ύπαρξη των κατάλληλων υπογραφών και δημιουργεί/ τροποποιεί το CIF στο σύστημα καταχωρώντας τα στοιχεία του πελάτη.

Στην επόμενη δραστηριότητα [Update_Application], το αρμόδιο στέλεχος στο τμήμα CDU, μεριμνά για την μετατροπή του χαρτοφυλακίου σε χαρτοφυλάκιο (τύπου 55) από τη Διεύθυνση Τεχνολογίας και Συστημάτων, ενώ όταν πρόκειται για νέο πελάτη ανοίγει το χαρτοφυλάκιο (τύπου 55), εάν χρειαστεί διευκρινίσεις επικοινωνεί με τις μονάδες PB για τυχόν εκκρεμότητες. Επίσης, ενημερώνει την ειδική εφαρμογή παρακολούθησης εγγράφων/ αποδοχή πελάτη του PB και καταχωρεί ενδεχόμενες εκκρεμότητες. Τέλος προωθεί τον φάκελο στον υπεύθυνο του CDU για έλεγχο.

Ο επικεφαλής του CDU εφόσον παραλάβει τον φάκελο του πελάτη [Check&Send_Monthly_Report], ελέγχει την ορθότητα των καταχωρήσεων του υπαλλήλου του αντιπαραβάλλοντας τα σχετικά έγγραφα. Στη συνέχεια, αφού ελέγξει την τήρηση των κανόνων KYC με βάση τα προσκομισθέντα δικαιολογητικά και τα τυχόν δυσμενή στοιχεία του νέου πελάτη (ΤΕΙΠΕΣΙΑΣ) συμπληρώνει το σχετικό πεδίο στο PCA και το υπογράφει ηλεκτρονικά συμπληρώνοντας τυχόν σχόλια. Κατόπιν, προωθεί τα έγγραφα του φυσικού αρχείου του πελάτη στον επικεφαλής του τμήματος Business Control & Compliance και ενημερώνει την ειδική εφαρμογή παρακολούθησης εγγράφων/ αποδοχή πελάτη του PB ενημερώνοντας ταυτόχρονα και το status του πελάτη.

Σημείωση: Οι επικεφαλείς των μονάδων και του τμήματος Control & Compliance ελέγχουν σε μηνιαία βάση τις ενημερωτικές καταστάσεις αφορούν:

1. Εκκρεμότητες (**Pending List**)
2. Αποδοχή νέων πελατών (**Approval List**)
3. Εκκρεμότητες εγκεκριμένων πελατών (**Missing Docs List**).

Στην περίπτωση που υπάρχουν βασικές εκκρεμότητες ο φάκελος του πελάτη επιστρέφεται στον αρμόδιο Unit Head.

Ο επικεφαλής του τμήματος Business Control & Compliance αφού ολοκληρώσει τον έλεγχο τήρησης των κανόνων KYC & πρόληψης και καταστολής έναντι Ξεπλύματος

Χρήματος εγκρίνει την αποδοχή του πελάτη ενεργοποιώντας τη δραστηριότητα [CheckKYC_AcceptCustomer]. Συγκεκριμένα, ελέγχεται προληπτικά το όνομα του πελάτη σε λίστες/ καταστάσεις έναντι Ξεπλύματος Χρήματος που υπάρχουν στη Μονάδα. Στην περίπτωση που υπάρχει υποψία ότι ο πελάτης μπορεί να εμπλέκεται σε παράνομες δραστηριότητες ή το όνομά του βρίσκεται σε λίστες ξεπλύματος χρήματος ενημερώνει άμεσα τον Compliance Officer της Τράπεζας ή και το Compliance Committee του PB για περαιτέρω διερεύνηση (κατά περίπτωση). Τέλος ελέγχει την πληρότητα των κριτηρίων αποδοχής πελάτη του Private Banking (suitable criteria) με βάση την εκάστοτε πολιτική αποδοχής του πελάτη PB. Εάν το αποτέλεσμα των ελέγχων είναι θετικό, ο επικεφαλής του τμήματος Business Control & Compliance εγκρίνει το PCA καταχωρώντας την ένδειξη «Εγκεκριμένη» ή «Εγκεκριμένη με Εκκρεμότητα». και επιστρέφει το φάκελο του πελάτη στο CDU για ολοκλήρωση της διαδικασίας.

Σημείωση: Σε περίπτωση που οι πελάτες αξιολογούνται ως υψηλού κινδύνου (higher risk), προωθούνται για έγκριση στο Senior Management ή/ και στο Compliance Committee του Private Banking (κατά περίπτωση).

Αφού ολοκληρωθεί η αποδοχή του πελάτη, ο υπάλληλος του τμήματος CDU διερευνά εάν απαιτείται καταχώρηση VSV [Check_For_VSV]. Στην περίπτωση που δεν υπάρχει VSV στο σύστημα, αποστέλλει ηλεκτρονικά αίτημα προς το ΤΜΗΜΑ ΤΗΡΗΣΗΣ & ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΛΑΤΩΝ και με συστημένη εσωτερική αλληλογραφία το πιστοποιητικό ταυτοποίησης, για την καταχώριση της υπογραφής του πελάτη στο σύστημα (VSV).

Στη δραστηριότητα [VSV_Registration] το τμήμα Τήρησης και Διαχείρισης Μητρώου Πελατών πραγματοποιεί καταχώρηση στην εφαρμογή VSV των στοιχείων ταυτοποίησης του πελάτη και επιστρέφει τα έγγραφα με συστημένη αλληλογραφία στον υπάλληλο του CDU. Μετά την καταχώρηση VSV, το αρμόδιο στέλεχος του CDU αφού παραλάβει τα έγγραφα από το ΤΜΗΜΑ ΤΗΡΗΣΗΣ & ΔΙΑΧΕΙΡΙΣΗΣ ΜΗΤΡΩΟΥ ΠΕΛΑΤΩΝ αρχειοθετεί όλα τα έγγραφα του πελάτη στο σχετικό φάκελο, σύμφωνα με τις οδηγίες του PB και ενημερώνει ηλεκτρονικά τον RM για την τακτοποίηση του φακέλου πελάτη σε περίπτωση αποδοχής πελάτη με την ένδειξη «Εγκεκριμένη με εκκρεμότητα».

Στην τελευταία δραστηριότητα [Cust_Account_Management] ο RM και ο ΑΟ μπορούν να διαχειριστούν τους λογαριασμούς του πελάτη ως εξής:

1. Σε περίπτωση αρχικής κατάθεσης νέου πελάτη.

- a. Επιβεβαιώνει τα στοιχεία της αρχικής κατάθεσης (ποσό, Τράπεζα, χώρα προέλευσης) σύμφωνα με τη δήλωση του στο CIP.
- b. Αποστέλλει αντίγραφο της τραπεζικής/ προσωπικής επιταγής, ή εισερχόμενο εμβάσματος (κατά περίπτωση) στο CDU για ενημέρωση φακέλου.

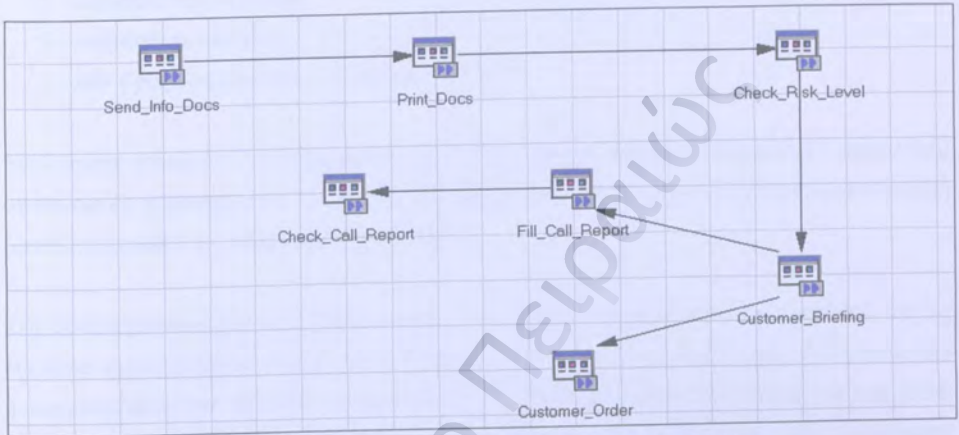
2. Μεριμνά για την ενημέρωση των πελατών σχετικά με:

- c. Τα προσφερόμενα προϊόντα της Τράπεζας
- d. Το είδος των καταθετικών λογαριασμών που μπορεί να τηρεί και τα επιτόκια.
- e. Τον τρόπο επικοινωνίας.
- f. Το είδος και τη συχνότητα της ενημέρωσης του.
- g. Τις προμήθειες με βάση τη πολιτική τιμολόγησης.

3. Φροντίζει για το άνοιγμα λογαριασμών του πελάτη.

2.3.3. Process Model Ενημέρωσης & Εξυπηρέτησης Πελατείας P.B.

Στο Σχήμα 3 αυτής της ενότητα παρουσιάζεται το Process Model Ενημέρωσης & Εξυπηρέτησης Πελατείας Private Banking μαζί με τις επιμέρους Activities. Το τρέχον μοντέλο αποτελείται από 7 δραστηριότητες.



Σχήμα 3: Διαδικασία Ενημέρωσης & Εξυπηρέτησης Πελατείας Private Banking

Στην πρώτη δραστηριότητα [Send_Info_Docs] οι διευθύνσεις Επενδυτικής Στρατηγικής, Διεθνών Κεφαλαιαγορών (Global Markets), ΑΕΔΑΚ και Securities, αποστέλλουν ενημερωτικά δελτία (εκθέσεις και αναφορές), οι οποίες αφορούν στην επιλεγμένη στρατηγική της Τράπεζας αναφορικά με αγορές, προϊόντα, αναμενόμενες εξελίξεις και επιθυμητές τοποθετήσεις.

Σημείωση: Η ΑΕΔΑΚ ενημερώνει τακτικά για τη σύνθεση/ αναδιάρθρωση των υπό διαχείριση αμοιβαίων κεφαλαίων και δίνει γενική πληροφόρηση στο σύνολο της Τράπεζας μέσω του Intranet site της.

Στη δεύτερη δραστηριότητα [Print_Docs] ο UH μαζί με τον RM παραλαμβάνουν τις ενημερωτικές αναφορές από τις αρμόδιες Διευθύνσεις της Τράπεζας, ελέγχουν τα ημερήσια ενημερωτικά δελτία στο Intranet της Τράπεζας και συμμετέχουν σε τηλεφωνικές συναντήσεις (Call Conference) προκειμένου να ενημερωθούν πλήρως για τις εξελίξεις στην Ελληνική και Διεθνή αγορά. Στη συνέχεια ο RM εκτυπώνει από το MIS τις καταστάσεις που παράγονται από το σύστημα, ώστε να ενημερωθεί για όλες τις κινήσεις που έχουν πραγματοποιηθεί στο χαρτοφυλάκιο του πελάτη. Ενδεικτικά εκτυπώνει:

1. Assets Under Management
2. Position Change Previous Day
3. Position Change from the End of Previous Year

Επίσης, ο ΑΟ καθημερινά εκτυπώνει στο σύστημα Portfolio Management

1. Ληξιάρια προθεσμιών
2. Ληξιάρια ομολόγων
3. Α/Κ βραχυπρόθεσμων τοποθετήσεων κ.τ.λ.

Ενώ όποτε απαιτείται, παραλαμβάνει τις λήξεις των προϊόντων εγγυημένου κεφαλαίου όπως αυτές έρχονται από το Treasury/ Capital Guarantee Desk σε ηλεκτρονική μορφή (email), έτσι ώστε να ενημερώσει τους RMs.

Στη δραστηριότητα [Check_Risk_Level], ο RM αφού ενημερώνεται από τον ΑΟ για τα ληξιάρια πραγματοποιεί έλεγχο στην διάρθρωση των χαρτοφυλακίων για να διαπιστωθεί η συμβατότητα του επιπέδου ρίσκου τους με το Risk Profile του πελάτη όπως αυτό εμφανίζεται από το IRQ. Κατά τον έλεγχο των χαρτοφυλακίων εξετάζεται και το ποσοστό έκθεσης του πελάτη σε συγκεκριμένο προϊόν (όπως αυτό ορίζεται κάθε φορά από τη Δ/ση του PB).

Σημείωση 1: Ο UH, επιβλέπει την πορεία των σχέσεων που διαχειρίζονται οι RMs καθώς και την πορεία επιλεγμένων χαρτοφυλακίων σημαντικών πελατών και αν απαιτείται τότε πραγματοποιείται επικοινωνία με τη Δ/ση Επενδυτικής Στρατηγικής για την προετοιμασία ενδεικτικών επενδυτικών στρατηγικών, οι οποίες αποστέλλονται με μήνυμα ηλεκτρονικού ταχυδρομείου (e-mail).

Στη δραστηριότητα [Customer_Briefing], ο RM επικοινωνεί με τον πελάτη με σκοπό να τον ενημερώσει:

1. Για τις τακτές λήξεις προϊόντων που τον αφορούν
2. Για την αγορά και την πορεία του χαρτοφυλακίου του
3. Για να επιβεβαιώσει ότι οι αντικειμενικοί στόχοι του πελάτη ικανοποιούνται και για να λάβει οδηγίες από τον πελάτη για τυχόν διορθωτικές ενέργειες.
4. Για τυχόν νέα προϊόντα
5. Για ενδεικτικές επενδυτικές στρατηγικές για το χαρτοφυλάκιο του (όπως αυτές διαμορφώθηκαν σε συνεργασία με τον UH και τη Δ/ση Επενδυτικής Στρατηγικής)

6. Να κανονίσει συνάντηση αν απαιτηθεί.

Σημείωση 1: Ο πελάτης παραλαμβάνει επίσημη αποτίμηση των χαρτοφυλακίων του από την Τράπεζα σε 3μηνιαία βάση. Σε ημερήσια βάση μπορεί να ελέγχει τη θέση του μέσω του site e-Banking.

Σημείωση 2: Σε περίπτωση απουσίας του RM, την ενημέρωση για το χαρτοφυλάκιο την κάνει ο ΑΟ.

Στην περίπτωση που ο πελάτης επιθυμεί να δώσει κάποια εντολή για την υλοποίηση/τροποποίηση της επενδυτικής του στρατηγικής τότε ενεργοποιείται η δραστηριότητα [Customer_Order] όπου καταχωρείται η εντολή του πελάτη στο σύστημα με σκοπό στη συνέχεια να κληθεί η διαδικασία Λήψης & Διαβίβασης Εντολών, σε αντίθετη περίπτωση ενεργοποιείται η δραστηριότητα [Fill_Call_Report].

Στην δραστηριότητα [Fill_Call_Report], ο RM συμπληρώνει στο Call Report του πελάτη, τα στοιχεία της τηλεφωνικής επαφής ή της συνάντησης με τον πελάτη καθώς και μία περίληψη των συναλλακτικών δραστηριοτήτων του πελάτη.

Τέλος, στην δραστηριότητα [Check_Call_Report], ο UH ελέγχει τακτικά τα Call Reports που συμπληρώνουν οι RMs της μονάδας που επιβλέπει και ενημερώνεται για την συχνότητα επαφής με τους πελάτες και γενικά τη σχέση που διατηρείται μαζί.

2.3.4. Staff – Προσωπικό

Στο Builtime του Workflow ορίζονται τα άτομα τα οποία είναι υπεύθυνα να διαχειρίζονται (εκκίνηση - τερματισμός) τις δραστηριότητες. Στη συγκεκριμένη περίπτωση ορίστηκε το ακόλουθο προσωπικό:

- E100 <ΜΑΡΙΑ ΚΑΡΑΠΑΝΑΓΙΩΤΟΥ>
- E200 <ΔΗΜΗΤΡΑ ΔΗΜΗΤΡΙΑΔΟΥ>
- E300 <ΠΑΥΛΟΣ ΠΑΥΛΙΔΗΣ>
- E400 <ΧΡΙΣΤΙΝΑ ΚΑΡΑΠΕΤΡΟΥ>
- E500 <ΙΩΑΝΝΑ ΙΩΑΝΝΙΔΟΥ>
- E600 <ΑΧΙΛΛΕΑΣ ΠΑΠΠΑΣ>
- E700 <ΑΝΤΩΝΗΣ ΑΝΤΩΝΙΟΥ>
- E800 <ΧΑΡΗΣ ΘΕΟΧΑΡΗΣ>
- Εξ' ορισμού υπάρχει ο χρήστης ADMIN <> που είναι ο διαχειριστής του συστήματος.

Για κάθε χρήστη εκτός από τα βασικά στοιχεία (User ID , Password) μπορούν να συμπληρωθούν και τα προσωπικά του στοιχεία, όπως First Name, Last Name, αλλά και το System και το Group που θα δημιουργηθεί. Στην συγκεκριμένη περίπτωση είναι τα (εξορισμού) FMCSYS – FMCGRP

2.3.5 Roles – Ρόλοι

Κάθε ρόλος αντιστοιχεί σε πολλά άτομα τα οποία έχουν ορισθεί στην προηγούμενη ενότητα. Όμοια με τα άτομα, με δεξί κλικ στο Roles επιλέγεται το New Role όπου και δημιουργείται. Στην οθόνη που εμφανίζεται, αναγράφεται το όνομα του ρόλου και μπορεί να πληκτρολογηθεί και μια μικρή περιγραφή του ρόλου που δημιουργείται (). Όπως και πριν έτσι και σε αυτό το σημείο πρέπει να ορισθούν τα άτομα που κατέχουν το συγκεκριμένο ρόλο. Από την φαίνεται ότι οι E100 και E200 είναι στελέχη σχέσεων πελατών (RM). Αν κάποιο άτομο έχει ορισθεί σαν Coordinator τότε θα εκτελεί τις δραστηριότητες των ατόμων που είναι απασχολημένοι. Οι ρόλοι που έχουν ορισθεί για το συγκεκριμένο μοντέλο είναι οι εξής:

- RELATIONSHIP MANAGER (RM)
- ACCOUNT OFFICER (AO)

- UNIT HEAD (UH)
- CDU EXECUTIVE (CDU_EXEC)
- IS EXECUTIVE (IS_EXEC)
- MURC EXECUTIVE (MURC_EXEC)
- BCC HEAD (BDDH)
- System Administrator (εξορισμού ρόλος για τον Διαχειριστή του συστήματος)

Σε κάθε ρόλο μπορούν να ορισθούν τα άτομα που θα έχουν τις ιδιότητες αυτού του ρόλου, καθώς και ο coordinator του ρόλου, ο οποίος στην περίπτωση που είναι απασχολημένο κάποιο άτομο του ίδιο ρόλου εκτελεί αυτός τις δραστηριότητες του.

2.3.6. Organizations – Οργανισμοί

Τέλος, δημιουργούνται οι οργανισμοί που συμμετέχουν σε κάθε διαδικασία στην προκειμένη περίπτωση, οι οργανισμοί θα αντιστοιχούν στα διάφορα τμήματα, και διευθύνσεις που εμπλέκονται στις τρεις διαδικασίες της Τράπεζας. Οι οργανισμοί των συγκεκριμένων process models είναι οι ακόλουθοι:

- Τμήμα Private Banking (PB)
- Investment Strategy (IS)
- Τμήμα Client Documentation Unit (CDU)
- Τμήμα Business Control & Compliance (BCC)
- Τμήμα Τήρησης & Διαχείρισης Μητρώου Πελατών (MURC)

Για κάθε οργανισμό ορίζεται ένα άτομο που θα ο manager του οργανισμού

2.3.7. Programs – Προγράμματα

Σε κάθε δραστηριότητα εκτελείται ένα πρόγραμμα το οποίο είναι κατασκευασμένο σε γλώσσα προγραμματισμού java. Θα πρέπει να σημειωθεί πως κάθε πρόγραμμα μπορεί να υλοποιηθεί σε διαφορετική γλώσσα προγραμματισμού, μιας και είναι ανεξάρτητα μεταξύ τους. Η ανεξαρτησία των προγραμμάτων οφείλεται κατά κύριο μέρος στο γεγονός της μεταφοράς δεδομένων με Containers, δηλαδή με τη βοήθεια κώδικα.

Για λόγους ευκολίας Στην συγκεκριμένη εφαρμογή τα ονόματα των προγραμμάτων είναι ίδια με αυτά των δραστηριοτήτων στις οποίες ανήκουν. π.χ. στην δραστηριότητα

[Check_For_New_Customers] είναι το Check_For_New_Customers.cmd. Στη συγκεκριμένη περίπτωση τα προγράμματα που δημιουργήθηκαν είναι τα εξής:

- Check_For_New_Customers
- Create_Personal_Customer_List
- Create_Week_Schedule
- Renew_Schedule
- First_Contact_Customers
- Renew_Personal_Customers_List
- Inform_Invest_Strategy
- Support_From_Invest_Strategy
- Prepare_Proposal
- Customer_Decision
- Customer_Not_Interested
- Customer_Interested
- Check_CIF_Existence
- Check&Fill_CIF_Pendencies
- Collect&Confirm_Customers_Data
- Send_PB_CIF_Request
- Fill_Formal_Documents
- Fill_PCA
- Check&Confirm_Cust_In_PB
- Send_Docs_To_CDU
- Open_Modify_CIF
- Update_Application
- Check&Send_Customer_File
- CheckKYC_AcceptCustomer
- Check_For_VSV
- VSV_Registration
- Cust_Account_Management
- Send_Info_Docs
- Print_Docs
- Check_Risk_Level
- Customer_Briefing
- Fill_Call_Report
- Check_Call_Report

- Customer_Order

Σε κάθε πρόγραμμα συμπληρώνεται η διαδρομή και το όνομα του εκτελέσιμου αρχείου που καλείται από αυτό, στην προκειμένη περίπτωση το αρχείο είναι κατάληξης **cmd** (Command file). Επίσης αναγράφεται και το μονοπάτι που βρίσχετε εγκαταστημένη η εφαρμογή, αποτελώντας το Working Directory, που για την περίπτωση αυτή είναι το **D:\DIPLOMATIKI\PROGRAMS_EXE**

Για τη σωστή λειτουργία του προγράμματος επιβάλλεται τα δεδομένα εισόδου και εξόδου (Input – Output Data) ενός προγράμματος να συμφωνούν με τα δεδομένα εισόδου και εξόδου της αντίστοιχης δραστηριότητας από την οποία αυτό καλείται, δηλαδή να έχουν ορισθεί οι ίδιοι containers (Data Structure).

2.3.8. Data Structure – Δομές Δεδομένων

Για κάθε δραστηριότητα ορίζονται οι δομές δεδομένων (Data Structure) που θα χρησιμοποιεί. Κάθε δραστηριότητα μπορεί να έχει διαφορετική Data Structure εισόδου και εξόδου, αυτό εξαρτάται από τα δεδομένα που δέχεται και τα αποτελέσματα που πρέπει να «περάσει» στην επόμενη δραστηριότητα. Πάντα όμως θα πρέπει να συμφωνούν οι Data Structures των δραστηριοτήτων με αυτές των προγραμμάτων που εκτελούνται κατά την κλήση τους. Στην προκειμένη περίπτωση ορίσθηκαν οι εξής Data Structures:

- EmployeeDS
- PR_CustomerDS

Θα πρέπει να σημειωθεί πως υποστηρίζονται οι «φωλιασμένες» δομές δεδομένων, κατά αυτόν τον τρόπο η PR_CustomerDS εκτός από ανεξάρτητη δομή δεδομένων, αποτελεί και μέλος της δομής EmployeeDS.

2.3.9. Export

Αφού ολοκληρωθεί ο σχεδιασμός του μοντέλου και όλων των ρυθμίσεων του, γίνεται εξαγωγή (Export) όλων αυτών των στοιχείων σε ένα αρχείο με κατάληξη **.fdl**, έτσι ώστε στη συνέχεια να εισαχθεί στον Client. Η εξαγωγή των στοιχείων μπορεί να γίνει για κάθε μία διαδικασία ξεχωριστά ή και για τις τρεις μαζί. Ακόμη μπορεί να γίνει εξαγωγή του αρχείου με μορφή **.html**.

Μετά την εξαγωγή το FDL αρχείο που δημιουργήθηκε είναι το εξής:

```

/*
 *
 * FDL definitions generated by MQ Workflow at 2009-07-16, 18:12:35.
 *
 */
CODEPAGE 1253
FM_RELEASE V3R4 0

/*
 * STRUCTURES
 */
UPDATE STRUCTURE 'Default Data Structure'
END 'Default Data Structure'

STRUCTURE 'PR_CustomersDs'
  DESCRIPTION "Ο υπάλληλος επιλέγει ελέγχει τους υποψήφιους πελάτες"
  'CID': LONG(15)
  DESCRIPTION "Κωδικός υποψήφιου πελάτη";
  'PR_CUSTOMER_NAME': STRING(50)
  DESCRIPTION "Όνομα υποψήφιου πελάτη";
  'PR_CUST_SNAME': STRING(50)
  DESCRIPTION "Επώνυμο υποψήφιου πελάτη / Επωνυμία εταιρείας";
  'PR_CUST_FNAME': STRING(50)
  DESCRIPTION "Πατρώνυμο";
  'PR_CUST_ADT': STRING(10)
  DESCRIPTION "Αριθμός Δελτίου Ταυτότητας";
  'PR_CUST_TAXCODE': STRING(15)
  DESCRIPTION "Αριθμός Φορολογικού Μητρώου";
  'PR_FLG_KIND': STRING(1)
  DESCRIPTION "FLAG Κατηγορίας Πελάτη 0 = Φυσικό πρόσωπο, 1 = Νομικό πρόσωπο";
  'PR_FLG_STATUS': STRING(1)
  DESCRIPTION "FLAG Κατάστασης Πελάτη 0 = Μη καταχωρημένος, 1 = Καταχωρημένος";
END 'PR_CustomersDs'

STRUCTURE 'EmployeeDS'
  'EMPL_ID': STRING(10)
  DESCRIPTION "Κωδικός Υπαλλήλου";
  'EMPL_NAME': STRING(50)
  DESCRIPTION "Όνομα Υπαλλήλου";
  'EMPL_SNAME': STRING(50);
  'PR_CustomerDS': STRING(10)
  DESCRIPTION "";
END 'EmployeeDS'

```



```

/*
 * PROCESS_CATEGORYs
 */
UPDATE PROCESS_CATEGORY 'Private_Banking'
  DOCUMENTATION "Private Banking"
END 'Private_Banking'

/*
 * PROGRAMs
 */
UPDATE PROGRAM 'Check_For_New_Customers' ( 'Default Data Structure',
'EmployeeDS' )
  DESCRIPTION ""
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WINNT EXE PATH AND FILENAME
"D:\DIPLOMATIKI\PROGRAMS_EXE\Check_For_New_Customers.cmd"
  WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'Check_For_New_Customers'

PROGRAM 'Create_Personal_Customer_List' ( 'EmployeeDS', 'EmployeeDS' )
  DESCRIPTION "Ο RM καταρτίζει την προσωπική του λίστα υποψήφιων πελάτων"
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WINNT EXE PATH AND FILENAME
"D:\DIPLOMATIKI\PROGRAMS_EXE\Create_Personal_Customer_List.cmd"
  WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'Create_Personal_Customer_List'

PROGRAM 'Create_Week_Schedule' ( 'Default Data Structure', 'Default Data
Structure' )
  DESCRIPTION "Ο RM καταρτίζει εβδομαδιαίο χρονοπρόγραμμα ενεργειών
το οποίο περιλαμβάνει τις επισκέψεις στο Δίκτιο
και τις λοιπές ενέργειες που σχεδιάζει να πραγματοποιήσει προκειμένου να
επισχύσει το πελατολόγιό του."
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WINNT EXE PATH AND FILENAME
"D:\DIPLOMATIKI\PROGRAMS_EXE\Create_Week_Schedule.cmd"
  WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'Create_Week_Schedule'

PROGRAM 'Renew_Schedule' ( 'EmployeeDS', 'EmployeeDS' )
  DESCRIPTION "Ο UH συζητά το πρόγραμμα με τον RM με στόχο την αναθεώρηση/
προτεραιοποίηση ενεργειών"
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WINNT EXE PATH AND FILENAME "D:\DIPLOMATIKI\PROGRAMS_EXE\Renew_Schedule.cmd"
  WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'Renew_Schedule'

```

```

PROGRAM 'First Contact Customers' ( 'EmployeeDS', 'EmployeeDS' )
  DESCRIPTION "Ο RM προετοιμάζει την πρώτη επικοινωνία με τον υποψήφιο πελάτη.
  Την δηλώνει στο σύστημα CRM και αφού διενεργήσει
  την πρώτη επικοινωνία συμπληρώνει το call report."
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WINNT EXE_PATH_AND_FILENAME
  "D:\DIPLOMATIKI\PROGRAMS_EXE\First_Contact_Customers.cmd"
  WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'First_Contact_Customers'

PROGRAM 'Renew Personal Customers List' ( 'EmployeeDS', 'EmployeeDS' )
  DESCRIPTION "Ο RM προτεραιοποιεί τη λίστα υποψήφίων πελατών του καταλήγοντας
  στους περισσότερο πιθανούς πελάτες."
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WINNT EXE_PATH_AND_FILENAME
  "D:\DIPLOMATIKI\PROGRAMS_EXE\Create_Possible_Customers_List.cmd"
  WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'Renew_Personal_Customers_List'

PROGRAM 'Inform Invest Strategy Section' ( 'Default Data Structure', 'Default
Data Structure' )
  DESCRIPTION "Ενημέρωση της Δ/σης Επενδυτικής Στρατηγικής και συνεργασία
  για τη διαμόρφωση της επενδυτικής πρότασης"
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WINNT EXE_PATH_AND_FILENAME
  "D:\DIPLOMATIKI\PROGRAMS_EXE\Inform_Invest_Strategy_Section.cmd"
  WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'Inform_Invest_Strategy_Section'

UPDATE PROGRAM 'Support From Invest Strategy' ( 'Default Data Structure',
'Default Data Structure' )
  DESCRIPTION "Η Διεύθυνση Επενδυτικής Στρατηγικής υποστηρίζει τους RMs στην
  προετοιμασία τους για πιθανούς πελάτες παρέχοντας εξειδικευμένη γνώση και
  επενδυτική άποψη με παρουσιάσεις προϊόντων & επενδυτικών προτάσεων
  ή και συμμετέχοντας στη συνάντηση με τον πελάτη."
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WINNT EXE_PATH_AND_FILENAME
  "D:\DIPLOMATIKI\PROGRAMS_EXE\Support_From_Invest_Strategy.cmd"
  WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'Support_From_Invest_Strategy'

PROGRAM 'Prepare Proposal' ( 'Default Data Structure', 'Default Data Structure'
)
  DESCRIPTION "Παρουσίαση Επενδυτικής Πρότασης. Ο RM παρουσιάζει την επενδυτική
  πρόταση
  της Διεύθυνσης Private Banking στους πιθανούς πελάτες."
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED

```

```

WINNT EXE PATH_AND_FILENAME
"D:\DIPLOMATIKI\PROGRAMS_EXE\Prepare_Proposal.cmd"
  WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'Prepare_Proposal'

PROGRAM 'Customer_Decision' ( 'PR_CustomersDs', 'PR_CustomersDs' )
  DESCRIPTION "Αποφασίζει ο πελάτης εάν ενδιαφέρεται για τις υπηρεσίες του
Private Banking"
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WINNT EXE PATH_AND_FILENAME
"D:\DIPLOMATIKI\PROGRAMS_EXE\Customer_Decision.cmd"
  WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'Customer_Decision'

PROGRAM 'Customer_Not_Interested' ( 'Default Data Structure', 'Default Data
Structure' )
  DESCRIPTION "Ο πελάτης δεν ενδιαφέρεται να αποδεχτεί τις υπηρεσίες του
Private Banking"
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WINNT EXE PATH_AND_FILENAME
"D:\DIPLOMATIKI\PROGRAMS_EXE\Customer_Not_Interested.cmd"
  WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'Customer_Not_Interested'

PROGRAM 'Customer_Interested' ( 'Default Data Structure', 'Default Data
Structure' )
  DESCRIPTION "Ο πελάτης ενδιαφέρεται να αποδεχθεί τις υπηρεσίες του Private
Banking.
Στην περίπτωση αυτή ο RM συμπληρώνει τα συμβατικά
έγγραφα και έντυπα της Τράπεζας"
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WINNT EXE PATH_AND_FILENAME
"D:\DIPLOMATIKI\PROGRAMS_EXE\Customer_Interested.cmd"
  WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'Customer_Interested'

UPDATE PROGRAM 'Check_CIF_Existence' ( 'Default Data Structure', 'Default Data
Structure' )
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
END 'Check_CIF_Existence'

UPDATE PROGRAM 'Check&Fill_CIF_Pendencies' ( 'Default Data Structure', 'Default
Data Structure' )
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
END 'Check&Fill_CIF_Pendencies'

```



```
UPDATE PROGRAM 'Collect&Confirm_Customers_Data' ( 'Default Data Structure',
'Default Data Structure' )
  DESCRIPTION ""
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
END 'Collect&Confirm_Customers_Data'

UPDATE PROGRAM 'Open_Modify_CIF' ( 'Default Data Structure', 'Default Data
Structure' )
  DESCRIPTION ""
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
END 'Open_Modify_CIF'

UPDATE PROGRAM 'Send_Docs_To_CDU' ( 'Default Data Structure', 'Default Data
Structure' )
  DESCRIPTION ""
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
END 'Send_Docs_To_CDU'

UPDATE PROGRAM 'Update_Application' ( 'Default Data Structure', 'Default Data
Structure' )
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
END 'Update_Application'

UPDATE PROGRAM 'Check&Confirm_Cust_In_PB' ( 'Default Data Structure', 'Default
Data Structure' )
  DESCRIPTION ""
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
END 'Check&Confirm_Cust_In_PB'

UPDATE PROGRAM 'CheckKYC_AcceptCustomer' ( 'Default Data Structure', 'Default
Data Structure' )
  DESCRIPTION ""
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
END 'CheckKYC_AcceptCustomer'

UPDATE PROGRAM 'VSV_Registration' ( 'Default Data Structure', 'Default Data
Structure' )
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
END 'VSV_Registration'

UPDATE PROGRAM 'Fill_PCA' ( 'Default Data Structure', 'Default Data Structure'
)
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
END 'Fill_PCA'

UPDATE PROGRAM 'Fill_Formal_Documents' ( 'Default Data Structure', 'Default
Data Structure' )
  DESCRIPTION ""
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
END 'Fill_Formal_Documents'

UPDATE PROGRAM 'Check_For_VSV' ( 'Default Data Structure', 'Default Data
Structure' )
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
```

```
END 'Check_For_VSV'

UPDATE PROGRAM 'Check&Send_Customer_File' ( 'Default Data Structure', 'Default
Data Structure' )
    NO STRUCTURES_FROM_ACTIVITY
    NOT UNATTENDED
END 'Check&Send_Customer_File'

UPDATE PROGRAM 'Send_PB_CIF_Request' ( 'Default Data Structure', 'Default Data
Structure' )
    DESCRIPTION ""
    NO STRUCTURES_FROM_ACTIVITY
    NOT UNATTENDED
END 'Send_PB_CIF_Request'

UPDATE PROGRAM 'Cust_Account_Management' ( 'Default Data Structure', 'Default
Data Structure' )
    NO STRUCTURES_FROM_ACTIVITY
    NOT UNATTENDED
END 'Cust_Account_Management'

UPDATE PROGRAM 'Send_Info_Docs' ( 'Default Data Structure', 'Default Data
Structure' )
    DESCRIPTION ""
    NO STRUCTURES_FROM_ACTIVITY
    NOT UNATTENDED
    WINNT EXE_PATH_AND_FILENAME "D:\DIPLOMATIKI\PROGRAMS_EXE\Send_Info_Docs.cmd"
    WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
    INHERIT_ENVIRONMENT
    STYLE_VISIBLE
    START_FOREGROUND
END 'Send_Info_Docs'

PROGRAM 'Print_Docs' ( 'Default Data Structure', 'Default Data Structure' )
    DESCRIPTION ""
    NO STRUCTURES_FROM_ACTIVITY
    NOT UNATTENDED
    WINNT EXE_PATH_AND_FILENAME "D:\DIPLOMATIKI\PROGRAMS_EXE\Print_Docs.cmd"
    WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
    INHERIT_ENVIRONMENT
    STYLE_VISIBLE
    START_FOREGROUND
END 'Print_Docs'

UPDATE PROGRAM 'Check_Risk_Level' ( 'Default Data Structure', 'Default Data
Structure' )
    DESCRIPTION ""
    NO STRUCTURES_FROM_ACTIVITY
    NOT UNATTENDED
    WINNT EXE_PATH_AND_FILENAME
"D:\DIPLOMATIKI\PROGRAMS_EXE\Check_Risk_Level.cmd"
    WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
    INHERIT_ENVIRONMENT
    STYLE_VISIBLE
    START_FOREGROUND
END 'Check_Risk_Level'

PROGRAM 'Customer_Briefing' ( 'Default Data Structure', 'Default Data
Structure' )
    DESCRIPTION ""
    NO STRUCTURES_FROM_ACTIVITY
    NOT UNATTENDED
    WINNT EXE_PATH_AND_FILENAME
"D:\DIPLOMATIKI\PROGRAMS_EXE\Customer_Briefing.cmd"
```

```

WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
INHERIT ENVIRONMENT
STYLE VISIBLE
START FOREGROUND
END 'Customer_Briefing'

PROGRAM 'Customer_Order' ( 'Default Data Structure', 'Default Data Structure' )
  DESCRIPTION ""
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WINNT_EXE_PATH_AND_FILENAME "D:\DIPLOMATIKI\PROGRAMS_EXE\Customer_Order.cmd"
  WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'Customer_Order'

PROGRAM 'Fill_Call_Report' ( 'Default Data Structure', 'Default Data Structure'
)
  DESCRIPTION ""
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WINNT_EXE_PATH_AND_FILENAME
  "D:\DIPLOMATIKI\PROGRAMS_EXE\Fill_Call_Report.cmd"
  WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'Fill_Call_Report'

UPDATE PROGRAM 'Check_Call_Report' ( 'Default Data Structure', 'Default Data
Structure' )
  DESCRIPTION ""
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WIN95_EXE_PATH_AND_FILENAME
  "D:\DIPLOMATIKI\PROGRAMS_EXE\Check_Call_Report.cmd"
  WORKING_DIRECTORY "D:\DIPLOMATIKI\PROGRAMS_EXE"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'Check_Call_Report'

/*
* PERSONS
*/
UPDATE PERSON 'ADMIN'
  AUTHORIZED_FOR STAFF
  AUTHORIZED_FOR AUTHORIZATION
  AUTHORIZED_FOR OPERATION
  AUTHORIZED_FOR TOPOLOGY
  AUTHORIZED_FOR PROCESS_MODELING
  AUTHORIZED_FOR PROCESS_CATEGORY ALL AS ADMINISTRATOR
  AUTHORIZED_FOR WORKITEMS OF ALL
  PASSWORD %F5E2FF2FA2F06A33133BEE48DB1ED773
  IS_NOT_ABSENT
  DO NOT RESET ABSENT
  GROUP 'FMCGRP'
  SYSTEM 'FMCSYS'
END 'ADMIN'

PERSON 'E100'
  DESCRIPTION ""
  PERSON_ID "E100"

```



```
LAST_NAME "ΚΑΡΑΠΑΝΑΓΙΩΤΟΥ"  
FIRST_NAME "ΜΑΡΙΑ"  
PASSWORD %CE511648FAE8643D9B2C37FAA1178544  
IS_NOT_ABSENT  
DO NOT RESET ABSENT  
GROUP 'FMCGRP'  
SYSTEM 'FMCSYS'  
END 'E100'
```

```
PERSON 'E200'  
DESCRIPTION ""  
PERSON_ID "E200"  
LAST_NAME "ΔΗΜΗΤΡΙΑΔΟΥ"  
FIRST_NAME "ΔΗΜΗΤΡΑ"  
PASSWORD %A2BB2BABF2595DB40178DA8A85AA05F9  
IS_NOT_ABSENT  
DO NOT RESET ABSENT  
GROUP 'FMCGRP'  
SYSTEM 'FMCSYS'  
END 'E200'
```

```
PERSON 'E300'  
DESCRIPTION ""  
PERSON_ID "E300"  
LAST_NAME "ΠΑΥΛΙΔΗΣ"  
FIRST_NAME "ΠΑΥΛΟΣ"  
PASSWORD %F8F6B982FF8C2AE17E3DC3652CCFC91F  
IS_NOT_ABSENT  
DO NOT RESET ABSENT  
GROUP 'FMCGRP'  
SYSTEM 'FMCSYS'  
END 'E300'
```

```
PERSON 'E400'  
DESCRIPTION ""  
PERSON_ID "E400"  
LAST_NAME "ΚΑΡΑΠΕΤΡΟΥ"  
FIRST_NAME "ΧΡΙΣΤΙΝΑ"  
PASSWORD %2B4E9CD5B655D77B9788AB2FFA29D2D9  
IS_NOT_ABSENT  
DO NOT RESET ABSENT  
GROUP 'FMCGRP'  
SYSTEM 'FMCSYS'  
END 'E400'
```

```
PERSON 'E500'  
DESCRIPTION ""  
PERSON_ID "E500"  
LAST_NAME "ΙΩΑΝΝΙΔΟΥ"  
FIRST_NAME "ΙΩΑΝΝΑ"  
PASSWORD %489ED076F183EE681FEF4FECC52C5739  
IS_NOT_ABSENT  
DO NOT RESET ABSENT  
GROUP 'FMCGRP'  
SYSTEM 'FMCSYS'  
END 'E500'
```

```
PERSON 'E600'  
DESCRIPTION ""  
PERSON_ID "E600"  
LAST_NAME "ΠΑΠΠΑΣ"  
FIRST_NAME "ΑΧΙΛΛΕΑΣ"  
PASSWORD %B2B646CFEC0B3C0F77B5B062A8232898  
IS_NOT_ABSENT
```

```

DO NOT RESET ABSENT
GROUP 'FMGRP'
SYSTEM 'FMCSYS'
END 'E600'

PERSON 'E700'
DESCRIPTION ""
PERSON_ID "E700"
LAST_NAME "ΑΝΤΩΝΙΟΥ"
FIRST_NAME "ΑΝΤΩΝΗΣ"
PASSWORD %EC37013C204F86A2371C410E92A867BC
IS_NOT_ABSENT
DO NOT RESET ABSENT
GROUP 'FMGRP'
SYSTEM 'FMCSYS'
END 'E700'

PERSON 'E800'
DESCRIPTION ""
PERSON_ID "E800"
LAST_NAME "ΘΕΟΧΑΡΗΣ"
FIRST_NAME "ΧΑΡΗΣ"
PASSWORD %BE1D988729740AE2EF4B2A00C9B464DD
IS_NOT_ABSENT
DO NOT RESET ABSENT
GROUP 'FMGRP'
SYSTEM 'FMCSYS'
END 'E800'

/*
 * ROLES
 */
UPDATE ROLE 'System administrator'
RELATED_PERSON 'ADMIN'
END 'System administrator'

ROLE 'RELATIONSHIP MANAGER'
DESCRIPTION "Ανατίθεται σε έναν επικεφαλής της ίδιας μονάδας (Unit Head),
έχει πρόσβαση στους υποψήφιους πελάτες"
RELATED_PERSON 'E100' 'E200'
END 'RELATIONSHIP MANAGER'

ROLE 'UNIT HEAD'
DESCRIPTION "Είναι ο επικεφαλής μιας μονάδας, διαχειρίζεται έναν ή και
περισσότερους
Relationship Managers της ίδιας μονάδας. Έχει πρόσβαση στους πελάτες
του και στους πελάτες των υφισταμένων του"
RELATED_PERSON 'E300'
END 'UNIT HEAD'

ROLE 'ACCOUNT OFFICER'
DESCRIPTION ""
RELATED_PERSON 'E400'
END 'ACCOUNT OFFICER'

ROLE 'CDU EXECUTIVE'
DESCRIPTION "Ο αρμόδιος υπάλληλος του τμήματος Client Documentation Unit"
RELATED_PERSON 'E500'
END 'CDU EXECUTIVE'

ROLE 'BCC HEAD'
DESCRIPTION "Ο επικεφαλής του τμήματος Business Control & Compliance"
RELATED_PERSON 'E800'
END 'BCC HEAD'

```

```

ROLE 'MURC EXECUTIVE'
  DESCRIPTION "Ο αρμόδιος υπαλληλος του τμήματος Τήρησης & Διαχείρισης Μητρώου
  Πελατών"
  RELATED_PERSON 'E600'
END 'MURC EXECUTIVE'

ROLE 'IS EXECUTIVE'
  DESCRIPTION ""
  RELATED_PERSON 'E700'
END 'IS EXECUTIVE'

/*
 * ORGANIZATIONS
 */
ORGANIZATION 'PRIVATE BANKING'
  DESCRIPTION ""
  MANAGER 'ADMIN'
  RELATED_PERSON 'E100' 'E200' 'E300' 'E400'
END 'PRIVATE BANKING'

ORGANIZATION 'CLIENT DOCUMENTATION UNIT'
  DESCRIPTION ""
  MANAGER 'E500'
  RELATED_PERSON 'E500'
END 'CLIENT DOCUMENTATION UNIT'

ORGANIZATION 'MURC'
  DESCRIPTION "Τμήμα Τήρησης και Διαχείρισης Μητρώου Πελατών"
  MANAGER 'E600'
  RELATED_PERSON 'E600'
END 'MURC'

ORGANIZATION 'INFORMATION STRATEGY'
  DESCRIPTION "Τμήμα Επενδυτικής Στρατηγικής"
  MANAGER 'E700'
  RELATED_PERSON 'E700'
END 'INFORMATION STRATEGY'

ORGANIZATION 'BUSINESS CONTROL & COMPLIANCE'
  DESCRIPTION ""
  MANAGER 'E800'
  RELATED_PERSON 'E800'
END 'BUSINESS CONTROL & COMPLIANCE'

/*
 * PROCESSES
 */
/*
 * PROCESS Attract_Customers_Process
 */
UPDATE PROCESS 'Attract_Customers_Process' ( 'Default Data Structure', 'Default
Data Structure' )
  DESCRIPTION "Η διαδικασία προσέλκυσης πελατείας περιγράφει τις ενέργειες
προσέλκυσης πελατών από τους Relationship Managers των μονάδων Private Banking"
  DOCUMENTATION "Η διαδικασία προσέλκυσης πελατείας αφορά την Διεύθυνση Private
Banking"
  CATEGORY 'Private_Banking'
  DO NOT PROMPT AT_PROCESS_START
  WINDOW VIEW REFERENCE_POINT XPOS 0 YPOS 489
  WINDOW SHOW ALL CONNECTORS
  WINDOW SHOW NO CONDITIONS
  PROGRAM_ACTIVITY 'Check_For_New_Customers' ( 'Default Data Structure',
'EmployeeDS' )

```



```

DESCRIPTION "Έλεγχος νέων Υποψήφιων Πελατών"
START MANUAL WHEN AT LEAST ONE CONNECTOR TRUE
EXIT AUTOMATIC WHEN " _RC=1" XPOS -800 YPOS 1059
LAYOUT XPOS -800 YPOS 1200
NAME_POSITION XPOS -800 YPOS 1125
PRIORITY DEFINED IN INPUT_CONTAINER
DONE BY PERSON 'ADMIN' 'E100' 'E200'
PROGRAM 'Check_For_New_Customers'
SYNCHRONIZATION NESTED
END 'Check_For_New_Customers'
PROGRAM_ACTIVITY 'Create_Personal_Customer_List' ( 'EmployeeDS', 'EmployeeDS'
)

```

```

DESCRIPTION "Δημιουργία Προσωπικής Λίστας Υποψήφιων Πελατών"
START MANUAL WHEN AT LEAST ONE CONNECTOR TRUE
EXIT AUTOMATIC WHEN " _RC=1" XPOS -300 YPOS 1059
LAYOUT XPOS -300 YPOS 1200
NAME_POSITION XPOS -300 YPOS 1125
PRIORITY DEFINED IN INPUT_CONTAINER
DONE BY PERSON 'ADMIN' 'E100' 'E200' 'E300'
PROGRAM 'Create_Personal_Customer_List'
SYNCHRONIZATION NESTED
END 'Create_Personal_Customer_List'
PROGRAM_ACTIVITY 'Create_Week_Schedule' ( 'Default Data Structure', 'Default
Data Structure' )

```

```

DESCRIPTION "Καθορισμός Χρονοπρογράμματος Ενεργειών"
START MANUAL WHEN AT LEAST ONE CONNECTOR TRUE
EXIT AUTOMATIC WHEN " _RC=1" XPOS 200 YPOS 1059
LAYOUT XPOS 200 YPOS 1200
NAME_POSITION XPOS 200 YPOS 1125
PRIORITY DEFINED IN INPUT_CONTAINER
DONE BY PERSON 'E100' 'E200' 'E300'
PROGRAM 'Create_Week_Schedule'
SYNCHRONIZATION NESTED
END 'Create_Week_Schedule'
PROGRAM_ACTIVITY 'Customer_Decision' ( 'PR_CustomersDs', 'PR_CustomersDs' )

```

```

DESCRIPTION "Παρουσίαση Επενδυτικής Πρότασης - Απόφαση του πελάτη εάν
ενδιαφέρεται για τις υπηρεσίες του Private Banking "
START MANUAL WHEN AT LEAST ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS 0 YPOS 100
NAME_POSITION XPOS 0 YPOS 25
PRIORITY DEFINED IN INPUT_CONTAINER
DONE BY PERSON 'E100' 'E200' 'E300' 'E400'
PROGRAM 'Customer_Decision'
SYNCHRONIZATION NESTED
END 'Customer_Decision'
PROGRAM_ACTIVITY 'Customer_Interested' ( 'Default Data Structure', 'Default
Data Structure' )
DESCRIPTION "Ο πελάτης ενδιαφέρεται να αποδεχθεί τις υπηρεσίες τους Private
Banking"
START MANUAL WHEN AT LEAST ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS 700 YPOS -100
NAME_POSITION XPOS 700 YPOS -175
PRIORITY DEFINED IN INPUT_CONTAINER
DONE BY PERSON 'E100' 'E200' 'E300' 'E400'
PROGRAM 'Customer_Interested'
SYNCHRONIZATION NESTED
END 'Customer_Interested'
PROGRAM_ACTIVITY 'Customer_Not_Interested' ( 'Default Data Structure',
'Default Data Structure' )
DESCRIPTION "Ο πελάτης δεν ενδιαφέρεται να αποδεχθεί τις υπηρεσίες του
Private Banking"
START MANUAL WHEN AT LEAST ONE CONNECTOR TRUE

```

```

EXIT AUTOMATIC
LAYOUT XPOS 700 YPOS 400
NAME POSITION XPOS 700 YPOS 325
PRIORITY DEFINED IN INPUT_CONTAINER
DONE BY PERSON 'E100' 'E200' 'E300' 'E400'
PROGRAM 'Customer_Not_Interested'
SYNCHRONIZATION NESTED
END 'Customer_Not_Interested'
PROGRAM ACTIVITY 'First_Contact_Customers' ( 'EmployeeDS', 'EmployeeDS' )
DESCRIPTION "Πρώτη επικοινωνία/ Επαφή με υποψήφιο πελάτη"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC WHEN " RC=1" XPOS 700 YPOS 559
LAYOUT XPOS 700 YPOS 700
NAME POSITION XPOS 700 YPOS 625
PRIORITY DEFINED IN INPUT_CONTAINER
DONE BY PERSON 'E100' 'E200' 'E300' 'E400'
PROGRAM 'First_Contact_Customers'
SYNCHRONIZATION NESTED
END 'First_Contact_Customers'
PROGRAM ACTIVITY 'Inform_Invest_Strategy' ( 'Default Data Structure',
'Default Data Structure' )
DESCRIPTION "Αίτηση συνεργασίας με τη Δ/ση Επενδυτικής Στρατηγικής"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS -800 YPOS 700
NAME POSITION XPOS -800 YPOS 625
PRIORITY DEFINED IN INPUT_CONTAINER
DONE BY PERSON 'E700'
PROGRAM 'Inform_Invest_Strategy_Section'
SYNCHRONIZATION NESTED
END 'Inform_Invest_Strategy'
PROGRAM ACTIVITY 'Prepare_Proposal' ( 'Default Data Structure', 'Default Data
Structure' )
DESCRIPTION "Προετοιμασία της Επενδυτικής Πρότασης"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS -800 YPOS 100
NAME POSITION XPOS -800 YPOS 25
PRIORITY DEFINED IN INPUT_CONTAINER
DONE BY PERSON 'E100' 'E200' 'E300' 'E400'
PROGRAM 'Prepare_Proposal'
SYNCHRONIZATION NESTED
END 'Prepare_Proposal'
PROGRAM ACTIVITY 'Renew_Personal_Customers_List' ( 'EmployeeDS', 'EmployeeDS'
)
DESCRIPTION "Αναθεώρηση Προσωπικής Λίστας Υποψήφιων Πελατών"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS -100 YPOS 700
NAME POSITION XPOS -100 YPOS 625
PRIORITY DEFINED IN INPUT_CONTAINER
DONE BY PERSON 'E100' 'E200' 'E300'
PROGRAM 'Renew_Personal_Customers_List'
SYNCHRONIZATION NESTED
END 'Renew_Personal_Customers_List'
PROGRAM ACTIVITY 'Renew_Schedule' ( 'EmployeeDS', 'EmployeeDS' )
DESCRIPTION "Έλεγχος - Αναθεώρηση προγράμματος"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS 700 YPOS 1200
NAME POSITION XPOS 700 YPOS 1125
PRIORITY DEFINED IN INPUT_CONTAINER
DONE BY PERSON 'E100' 'E200' 'E300' 'E400'
PROGRAM 'Renew_Schedule'

```

```

    SYNCHRONIZATION NESTED
END 'Renew_Schedule'
PROGRAM ACTIVITY 'Support_From_Invest_Strategy' ( 'Default Data Structure',
'Default Data Structure' )
    DESCRIPTION "Υποστήριξη από τη Δ/ση Επενδυτικής Στρατηγικής"
    START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
    EXIT AUTOMATIC
    LAYOUT XPOS -800 YPOS 400
    NAME_POSITION XPOS -800 YPOS 325
    PRIORITY DEFINED IN INPUT_CONTAINER
    DONE_BY PERSON 'E700'
    PROGRAM 'Support_From_Invest_Strategy'
    SYNCHRONIZATION NESTED
END 'Support_From_Invest_Strategy'
CONTROL
    FROM 'Support_From_Invest_Strategy' TO 'Prepare_Proposal'
CONTROL
    FROM 'Check_For_New_Customers' TO 'Create_Personal_Customer_List'
CONTROL
    FROM 'Renew_Personal_Customers_List' TO 'Inform_Invest_Strategy'
    LAYOUT
        XPOS -400 YPOS 700
CONTROL
    FROM 'Renew_Personal_Customers_List' TO 'Prepare_Proposal'
CONTROL
    FROM 'Create_Personal_Customer_List' TO 'Create_Week_Schedule'
CONTROL
    FROM 'Create_Week_Schedule' TO 'First_Contact_Customers'
CONTROL
    FROM 'Create_Week_Schedule' TO 'Renew_Schedule'
CONTROL
    FROM 'Customer_Decision' TO 'Customer_Interested'
CONTROL
    FROM 'Customer_Decision' TO 'Customer_Not_Interested'
CONTROL
    FROM 'First_Contact_Customers' TO 'Renew_Personal_Customers_List'
    LAYOUT
        XPOS 300 YPOS 700
CONTROL
    FROM 'Inform_Invest_Strategy' TO 'Support_From_Invest_Strategy'
CONTROL
    FROM 'Prepare_Proposal' TO 'Customer_Decision'
    LAYOUT
        XPOS -400 YPOS 100
CONTROL
    FROM 'Renew_Schedule' TO 'First_Contact_Customers'
END 'Attract_Customers_Process'

/*
* PROCESS Customer_Acceptance_Process
*/
UPDATE PROCESS 'Customer_Acceptance_Process' ( 'Default Data Structure',
'Default Data Structure' )
    DESCRIPTION "Διαδικασία Αποδοχής Πελάτη"
    DOCUMENTATION "Η διαδικασία αποδοχής πελάτη Private Banking περιγράφει τις
απαραίτητες ενέργειες που απαιτούνται για την αποδοχή του πελάτη σύμφωνα με τα
κριτήρια του Private Banking, των διαδικασιών της Τράπεζας και της Πολιτικής
Know Your Customer και Anti - Money Laundering του ομίλου."
    CATEGORY 'Private Banking'
    DO NOT PROMPT AT PROCESS_START
    WINDOW VIEW_REFERENCE_POINT XPOS 0 YPOS -146
    WINDOW SHOW_ALL CONNECTORS
    WINDOW SHOW_NO CONDITIONS

```



```

PROGRAM_ACTIVITY 'Check_CIF_Existence' ( 'Default Data Structure', 'Default
Data Structure' )
  DESCRIPTION "Ο ΑΟ λαμβάνει το αίτημα από τον RM και προβαίνει στον έλεγχο
ύπαρξης CIF"
  START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
  EXIT AUTOMATIC
  LAYOUT XPOS -700 YPOS 1100
  NAME_POSITION XPOS -700 YPOS 1025
  PRIORITY DEFINED_IN INPUT_CONTAINER
  PROGRAM 'Check_CIF_Existence'
  SYNCHRONIZATION NESTED
END 'Check_CIF_Existence'
PROGRAM_ACTIVITY 'Check_For_VSV' ( 'Default Data Structure', 'Default Data
Structure' )
  DESCRIPTION "Διερεύνηση εάν απαιτείται καταχώριση VSV"
  START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
  EXIT AUTOMATIC
  LAYOUT XPOS -700 YPOS -100
  NAME_POSITION XPOS -700 YPOS -175
  PRIORITY DEFINED_IN INPUT_CONTAINER
  PROGRAM 'Check_For_VSV'
  SYNCHRONIZATION NESTED
END 'Check_For_VSV'
PROGRAM_ACTIVITY 'Check&Confirm_Cust_In_PB' ( 'Default Data Structure',
'Default Data Structure' )
  START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
  EXIT AUTOMATIC
  LAYOUT XPOS -700 YPOS 700
  NAME_POSITION XPOS -700 YPOS 625
  PRIORITY DEFINED_IN INPUT_CONTAINER
  PROGRAM 'Check&Confirm_Cust_In_PB'
  SYNCHRONIZATION NESTED
END 'Check&Confirm_Cust_In_PB'
PROGRAM_ACTIVITY 'Check&Fill_CIF_Pendencies' ( 'Default Data Structure',
'Default Data Structure' )
  DESCRIPTION "Έλεγχος Εκκρεμοτήτων"
  START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
  EXIT AUTOMATIC
  LAYOUT XPOS -400 YPOS 1400
  NAME_POSITION XPOS -400 YPOS 1325
  PRIORITY DEFINED_IN INPUT_CONTAINER
  PROGRAM 'Check&Fill_CIF_Pendencies'
  SYNCHRONIZATION NESTED
END 'Check&Fill_CIF_Pendencies'
PROGRAM_ACTIVITY 'Check&Send_Customer_File' ( 'Default Data Structure',
'Default Data Structure' )
  DESCRIPTION "Έλεγχος πληρότητας φακέλου πελάτη"
  START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
  EXIT AUTOMATIC
  LAYOUT XPOS 700 YPOS -100
  NAME_POSITION XPOS 700 YPOS -175
  PRIORITY DEFINED_IN INPUT_CONTAINER
  PROGRAM 'Check&Send_Customer_File'
  SYNCHRONIZATION NESTED
END 'Check&Send_Customer_File'
PROGRAM_ACTIVITY 'CheckKYC_AcceptCustomer' ( 'Default Data Structure',
'Default Data Structure' )
  DESCRIPTION "Έλεγχος τήρησης των κανόνων KYC & Αποδοχή πελάτη"
  START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
  EXIT AUTOMATIC
  LAYOUT XPOS 0 YPOS -100
  NAME_POSITION XPOS 0 YPOS -175
  PRIORITY DEFINED_IN INPUT_CONTAINER
  PROGRAM 'CheckKYC_AcceptCustomer'

```

```

SYNCHRONIZATION NESTED
END 'CheckKYC_AcceptCustomer'
PROGRAM ACTIVITY 'Collect&Confirm_Customers_Data' ( 'Default Data Structure',
'Default Data Structure' )
DESCRIPTION "Ευλόγη στοιχεία αυτοποίησης"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS 0 YPOS 1100
NAME_POSITION XPOS 0 YPOS 1025
PRIORITY DEFINED_IN INPUT_CONTAINER
PROGRAM 'Collect&Confirm_Customers_Data'
SYNCHRONIZATION NESTED
END 'Collect&Confirm_Customers_Data'
PROGRAM ACTIVITY 'Cust_Account_Management' ( 'Default Data Structure',
'Default Data Structure' )
DESCRIPTION "Διαχείριση Λογαριασμών Πελάτη"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS 0 YPOS -600
NAME_POSITION XPOS 0 YPOS -675
PRIORITY DEFINED_IN INPUT_CONTAINER
PROGRAM 'Cust_Account_Management'
SYNCHRONIZATION NESTED
END 'Cust_Account_Management'
PROGRAM ACTIVITY 'Fill_Formal_Documents' ( 'Default Data Structure', 'Default
Data Structure' )
DESCRIPTION "Συμπλήρωση λοιπων προβλεπόμενων συμβατικών εγγράφων καθώς και
εντύπων παροχής υπηρεσιών και παραλαβή υπογραφής πελάτη"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS 700 YPOS 700
NAME_POSITION XPOS 700 YPOS 625
PRIORITY DEFINED_IN INPUT_CONTAINER
PROGRAM 'Fill_Formal_Documents'
SYNCHRONIZATION NESTED
END 'Fill_Formal_Documents'
PROGRAM ACTIVITY 'Fill_PCA' ( 'Default Data Structure', 'Default Data
Structure' )
DESCRIPTION "Συμπλήρωση και υπογραφή εντύπου PCA"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS 0 YPOS 700
NAME_POSITION XPOS 0 YPOS 625
PRIORITY DEFINED_IN INPUT_CONTAINER
PROGRAM 'Fill_PCA'
SYNCHRONIZATION NESTED
END 'Fill_PCA'
PROGRAM ACTIVITY 'Open_Modify_CIF' ( 'Default Data Structure', 'Default Data
Structure' )
DESCRIPTION "Άνοιγμα CIF νέου πελάτη στο σύστημα "
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS 0 YPOS 300
NAME_POSITION XPOS 0 YPOS 225
PRIORITY DEFINED_IN INPUT_CONTAINER
PROGRAM 'Open_Modify_CIF'
SYNCHRONIZATION NESTED
END 'Open_Modify_CIF'
PROGRAM ACTIVITY 'Send_Docs_To_CDU' ( 'Default Data Structure', 'Default Data
Structure' )
DESCRIPTION "Αποστολή στοιχείων νέου πελάτη στο CDU"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS -700 YPOS 300

```

```

NAME_POSITION XPOS -700 YPOS 225
PRIORITY DEFINED_IN INPUT_CONTAINER
PROGRAM 'Send_Docs_To_CDU'
SYNCHRONIZATION NESTED
END 'Send_Docs_To_CDU'
PROGRAM_ACTIVITY 'Send_PB_CIF_Request' ( 'Default Data Structure', 'Default
Data Structure' )
DESCRIPTION "Συμπλήρωση και υπογραφή αίτησης ανοίγματος CIF πελάτη Private
Banking"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS 700 YPOS 1100
NAME_POSITION XPOS 700 YPOS 1025
PRIORITY DEFINED_IN INPUT_CONTAINER
PROGRAM 'Send_PB_CIF_Request'
SYNCHRONIZATION NESTED
END 'Send_PB_CIF_Request'
PROGRAM_ACTIVITY 'Update_Application' ( 'Default Data Structure', 'Default
Data Structure' )
DESCRIPTION "Ενημέρωση Εφαρμογής"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS 700 YPOS 300
NAME_POSITION XPOS 700 YPOS 225
PRIORITY DEFINED_IN INPUT_CONTAINER
PROGRAM 'Update_Application'
SYNCHRONIZATION NESTED
END 'Update_Application'
PROGRAM_ACTIVITY 'VSV_Registration' ( 'Default Data Structure', 'Default Data
Structure' )
DESCRIPTION "Καταχώρηση VSV"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS -700 YPOS -600
NAME_POSITION XPOS -700 YPOS -675
PRIORITY DEFINED_IN INPUT_CONTAINER
PROGRAM 'VSV_Registration'
SYNCHRONIZATION NESTED
END 'VSV_Registration'
CONTROL
FROM 'Check_CIF_Existence' TO 'Check&Fill_CIF_Pendencies'
CONTROL
FROM 'Check_CIF_Existence' TO 'Collect&Confirm_Customers_Data'
LAYOUT
XPOS -100 YPOS 1100
CONTROL
FROM 'Check&Confirm_Cust_In_PB' TO 'Send_Docs_To_CDU'
LAYOUT
XPOS -700 YPOS 400
CONTROL
FROM 'Check_For_VSV' TO 'Cust_Account_Management'
CONTROL
FROM 'Check_For_VSV' TO 'VSV_Registration'
LAYOUT
XPOS -700 YPOS -300
CONTROL
FROM 'CheckKYC_AcceptCustomer' TO 'Check_For_VSV'
LAYOUT
XPOS -400 YPOS -100
CONTROL
FROM 'Check&Fill_CIF_Pendencies' TO 'Collect&Confirm_Customers_Data'
CONTROL
FROM 'Check&Send_Customer_File' TO 'CheckKYC_AcceptCustomer'
LAYOUT

```



```

XPOS 400 YPOS -100
CONTROL
FROM 'Collect&Confirm_Customers_Data' TO 'Send_PB_CIF_Request'
LAYOUT
XPOS 600 YPOS 1100
CONTROL
FROM 'Fill_Formal_Documents' TO 'Fill_PCA'
LAYOUT
XPOS 300 YPOS 700
CONTROL
FROM 'Send_PB_CIF_Request' TO 'Fill_Formal_Documents'
LAYOUT
XPOS 700 YPOS 800
CONTROL
FROM 'Fill_PCA' TO 'Check&Confirm_Cust_In_PB'
LAYOUT
XPOS -400 YPOS 700
CONTROL
FROM 'Open_Modify_CIF' TO 'Update_Application'
LAYOUT
XPOS 300 YPOS 300
CONTROL
FROM 'Send_Docs_To_CDU' TO 'Open_Modify_CIF'
LAYOUT
XPOS -300 YPOS 300
CONTROL
FROM 'Update_Application' TO 'Check&Send_Customer_File'
LAYOUT
XPOS 700 YPOS 100
CONTROL
FROM 'VSV_Registration' TO 'Cust_Account_Management'
END 'Customer_Acceptance_Process'

/*
* PROCESS Customer_Inform&Service_Process
*/
UPDATE PROCESS 'Customer_Inform&Service_Process' ( 'Default Data Structure',
'Default Data Structure' )
DESCRIPTION "Διαδικασία Ενημέρωσης και Εξυπηρέτησης Πελατείας"
DOCUMENTATION "Η διαδικασία Ενημέρωσης και Εξυπηρέτησης Πελατείας περιγράφει
τις ενέργειες των Relationship Managers των μονάδων Private Banking προς τους
πελάτες τους"
CATEGORY 'Private Banking'
DO NOT PROMPT AT PROCESS_START
WINDOW VIEW_REFERENCE_POINT XPOS 0 YPOS 671
WINDOW SHOW_ALL_CONNECTORS
WINDOW SHOW_NO_CONDITIONS
PROGRAM ACTIVITY 'Check_Call_Report' ( 'Default Data Structure', 'Default
Data Structure' )
DESCRIPTION "Τακτικός έλεγχος Call Report"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS -300 YPOS 700
NAME_POSITION XPOS -300 YPOS 625
PRIORITY DEFINED_IN_INPUT_CONTAINER
PROGRAM 'Check_Call_Report'
SYNCHRONIZATION_NESTED
END 'Check_Call_Report'
PROGRAM ACTIVITY 'Check_Risk_Level' ( 'Default Data Structure', 'Default Data
Structure' )
DESCRIPTION "Έλεγχος χαρτοφυλακίων και εξέταση επιθυμητού επιπέδου ρίσκου"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS 700 YPOS 1000

```

```

NAME POSITION XPOS 700 YPOS 925
PRIORITY DEFINED_IN INPUT_CONTAINER
PROGRAM 'Check_Risk_Level'
SYNCHRONIZATION NESTED
END 'Check_Risk_Level'
PROGRAM_ACTIVITY 'Customer_Briefing' ( 'Default Data Structure', 'Default
Data Structure' )
DESCRIPTION "Επικοινωνία για ενημέρωση πελάτη"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS 700 YPOS 500
NAME_POSITION XPOS 700 YPOS 425
PRIORITY DEFINED_IN INPUT_CONTAINER
PROGRAM 'Customer_Briefing'
SYNCHRONIZATION NESTED
END 'Customer_Briefing'
PROGRAM_ACTIVITY 'Customer_Order' ( 'Default Data Structure', 'Default Data
Structure' )
DESCRIPTION "Λήψη Εντολής Πελάτη"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS 200 YPOS 300
NAME_POSITION XPOS 200 YPOS 225
PRIORITY DEFINED_IN INPUT_CONTAINER
PROGRAM 'Customer_Briefing'
SYNCHRONIZATION NESTED
END 'Customer_Order'
PROGRAM_ACTIVITY 'Fill_Call_Report' ( 'Default Data Structure', 'Default Data
Structure' )
DESCRIPTION "Ο Πελάτης δεν επιθυμεί να δώσει κάποια εντολή - Ο RM
συμπληρώνει το Call Report"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS 200 YPOS 700
NAME_POSITION XPOS 200 YPOS 625
PRIORITY DEFINED_IN INPUT_CONTAINER
PROGRAM 'Fill_Call_Report'
SYNCHRONIZATION NESTED
END 'Fill_Call_Report'
PROGRAM_ACTIVITY 'Print_Docs' ( 'Default Data Structure', 'Default Data
Structure' )
DESCRIPTION "Εκτύπωση Ενημερωτικών Καταστάσεων"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS 0 YPOS 1000
NAME_POSITION XPOS 0 YPOS 925
PRIORITY DEFINED_IN INPUT_CONTAINER
PROGRAM 'Print_Docs'
SYNCHRONIZATION NESTED
END 'Print_Docs'
PROGRAM_ACTIVITY 'Send_Info_Docs' ( 'Default Data Structure', 'Default Data
Structure' )
DESCRIPTION "Αποστολή Ενημερωτικών Δελτίων"
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC
LAYOUT XPOS -700 YPOS 1000
NAME_POSITION XPOS -700 YPOS 925
PRIORITY DEFINED_IN INPUT_CONTAINER
PROGRAM 'Send_Info_Docs'
SYNCHRONIZATION NESTED
END 'Send_Info_Docs'
CONTROL
FROM 'Check_Risk_Level' TO 'Customer_Briefing'
CONTROL

```

```
FROM 'Customer_Briefing' TO 'Customer_Order'  
CONTROL  
FROM 'Customer_Briefing' TO 'Fill_Call_Report'  
CONTROL  
FROM 'Fill_Call_Report' TO 'Check_Call_Report'  
CONTROL  
FROM 'Print_Docs' TO 'Check_Risk_Level'  
CONTROL  
FROM 'Send_Info_Docs' TO 'Print_Docs'  
END 'Customer_Inform&Service_Process'
```

Πανεπιστήμιο Πειραιώς

2.4 WebSphere MQ Workflow Client

Μετά την εξαγωγή του fdl αρχείου θα πρέπει να αποθηκευθεί στον εξής φάκελο:

C:\Program Files\IBM WebSphere MQ Workflow\FDL

για να μπορέσει στη συνέχεια να γίνει import στον client. Ο Client αποτελεί το μέσο με το οποίο ο χρήστης θα μπορεί να τρέχει την εφαρμογή.

Η διαδικασία εισαγωγής (import) στον client γίνεται από το MSDOS Command Prompt, αφού μεταφερθούμε στον φάκελο που αποθηκεύθηκε το αρχείο fdl, τρέχουμε την ακόλουθη εντολή:

fmcibie -u ADMIN -p password -i PBANK.fdl -o -t

Στην περίπτωση που δεν υπάρχει κάποιο λάθος τότε στην οθόνη (Εικόνα 1) θα εμφανιστεί η πρόταση **fmcibie finished and found 0 errors 0 warnings. RC=0**

```

C:\WINDOWS\system32\cmd.exe
FMC25100I REPLACE ROLE 'UNIT HEAD' finished.
FMC25100I CREATE ROLE 'ACCOUNT OFFICER' finished.
FMC25100I CREATE ROLE 'CDU EXECUTIVE' finished.
FMC25100I CREATE ROLE 'BCC HEAD' finished.
FMC25100I CREATE ROLE 'MURC EXECUTIVE' finished.
FMC25100I CREATE ROLE 'IS EXECUTIVE' finished.
FMC25100I REPLACE ORGANIZATION 'PRIVATE BANKING' finished.
FMC25100I CREATE ORGANIZATION 'CLIENT DOCUMENTATION UNIT' finished.
FMC25100I CREATE ORGANIZATION 'MURC' finished.
FMC25100I CREATE ORGANIZATION 'INFORMATION STRATEGY' finished.
FMC25100I CREATE ORGANIZATION 'BUSINESS CONTROL & COMPLIANCE' finished.
FMC25100I REPLACE PROCESS 'Attract_Customers_Process' finished.
FMC20510I Finished parsing FDL_01.fdl.
FMC21500I Begin verification of process 'Attract_Customers_Process'.
FMC21510I End verification of process 'Attract_Customers_Process' (0 errors, 0 w
arnings).
FMC26500I Begin translation of process 'Attract_Customers_Process'.
FMC26510I End translation of process 'Attract_Customers_Process' (0 errors, 0 wa
arnings).
FMC24560I fmcibie finished and found 0 errors 0 warnings. RC = 0
C:\Program Files\IBM WebSphere MQ Workflow\FDL>
  
```

Εικόνα 3: Αποτέλεσμα εισαγωγής fdl αρχείου στον client

Τώρα πλέον ο χρήστης μπορεί να θέσει σε λειτουργία τον Client και να τρέξει την εφαρμογή. Αυτό επιτυγχάνεται μέσα από την Έναρξη → Προγράμματα → IBM WebSphere MQ Workflow → WebSphere MQ Workflow Client – FMC.

Την πρώτη φορά που θα τρέξει ο χρήστης τον Client θα πρέπει να δημιουργήσει τα δικά του Worklist, Templates και Instances. Το ίδιο μπορεί να κάνει και ο Administrator για να παρακολουθεί τη ροή των διαδικασιών.

The screenshot displays three windows from the IBM Worklist application:

- Work Items [FMCGRP ADMIN ADMIN]**: A table listing tasks.

Name	Owner	State	Activity Type
Check_For_New_Customers	ADMIN	Ready	Program
Check_For_New_Customers	E100	Ready	Program
Check_For_New_Customers	E200	Ready	Program
- Process Templates [FMCGRP ADMIN ADMIN]**: A table listing templates.

Name	Description	Category	Creation Time	Last Modification
Attract_Customers...	Η διαδικασία ηρο...	Private_Banking	16/7/2009 16:10:16	16/7/2009 16:10:16
- Process Instances [FMCGRP ADMIN ADMIN]**: A table listing instances.

Name	Description	Category	Parent
Attract_Customers_Process	Η διαδικασία ηρο...	Private_Banking	
- Tree View**: A hierarchical view of the system structure, showing folders for 'Process Template Lists', 'Process Instance Lists', 'Worklists', and 'ADMIN' under the 'FMCGRP - ADMIN' node.

Εικόνα 4: Δημιουργία Worklist, Template και Instances

Ενδεικτικά ο χρήστης πατώντας επάνω στη διαδικασία [Check_For_New_Customers] εμφανίζεται η παρακάτω οθόνη:

Οθόνη 1: Έλεγχος Διαθέσιμων Υποψήφιων Πελατών

Επιλογή Βοήθεια

Ημερομηνία: 16/07/2009 Ώρα: 11:11:44

ΤΜΗΜΑ: Προσέλευσης Πελατείας

Διαθέσιμοι Υποψήφιοι Πελάτες

Κωδικός Πελάτη	Επώνυμο Πελάτη	Όνομα Πελάτη	Πατρώνυμο Πελάτη	Α.Φ.Μ.	Επιλογή Πελάτη
100	ΠΑΠΑΔΑΚΗΣ	ΓΩΡΓΟΣ	ΜΑΝΩΛΗΣ	1001001001	0
200	ΠΑΠΑΔΟΠΟΥΛΟΣ	ΠΑΝΗΣ	ΠΕΤΡΟΣ	1001001002	0
300	ΓΕΡΑΣΙΜΙΔΟΥ	ΕΛΕΝΗ	ΑΝΔΡΕΑΣ	1001001003	0

Φυσικά Πρόσωπα
 Νομικά Πρόσωπα

Εικόνα 5: Οθόνη εφαρμογής που αντιστοιχεί στη διαδικασία Προσέλευσης Πελατείας

3.1. Εισαγωγή

Το βιβλίο αυτό είναι η έκδοση της δεύτερης έκδοσης του βιβλίου της Microsoft Office → VBA 2003. Η 2η έκδοση του βιβλίου αφορά την Office 2003 και για την 3η έκδοση του βιβλίου επηρεάζονται το βιβλίο της Τεχνολογίας Τεχνολογίας 47.5.11.

Με την ολοκλήρωση των εργασιών του βιβλίου αυτού, η βάση δεδομένων είναι έτοιμη να χρησιμοποιηθεί στην πράξη. Οι πληροφορίες που είναι γνωστές ή άγνωστες για τους αναγνώστες είναι, με την ολοκλήρωση της έκδοσης, οι πληροφορίες που είναι γνωστές ή άγνωστες για τους αναγνώστες.

ΚΕΦΑΛΑΙΟ 3° ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Πανεπιστήμιο Πειραιώς

3.1. Εισαγωγή

Τα διαγράμματα αυτής της ενότητας είναι κατασκευασμένα με το εργαλείο της Microsoft Office → Visio 2003. Η βάση δεδομένων που χρησιμοποιείται είναι η Oracle v9.2.0.1 ενώ για την δημιουργία των πινάκων χρησιμοποιήθηκε το εργαλείο της Quest Software Toad v7.6.0.11

Με την ολοκλήρωση των βημάτων του προηγούμενου κεφαλαίου, η Βάση Δεδομένων είναι έτοιμη να χρησιμοποιηθεί στην εφαρμογή. Πριν όμως αναλυθεί με ποιο τρόπο γίνεται η σύνδεση της από πλευράς κώδικα, παρουσιάζεται και αναλύεται το μοντέλο Οντοτήτων Σχέσεων (ER) για τους ρόλους και τμηματικά για κάθε μία από τις τρεις διαδικασίες.

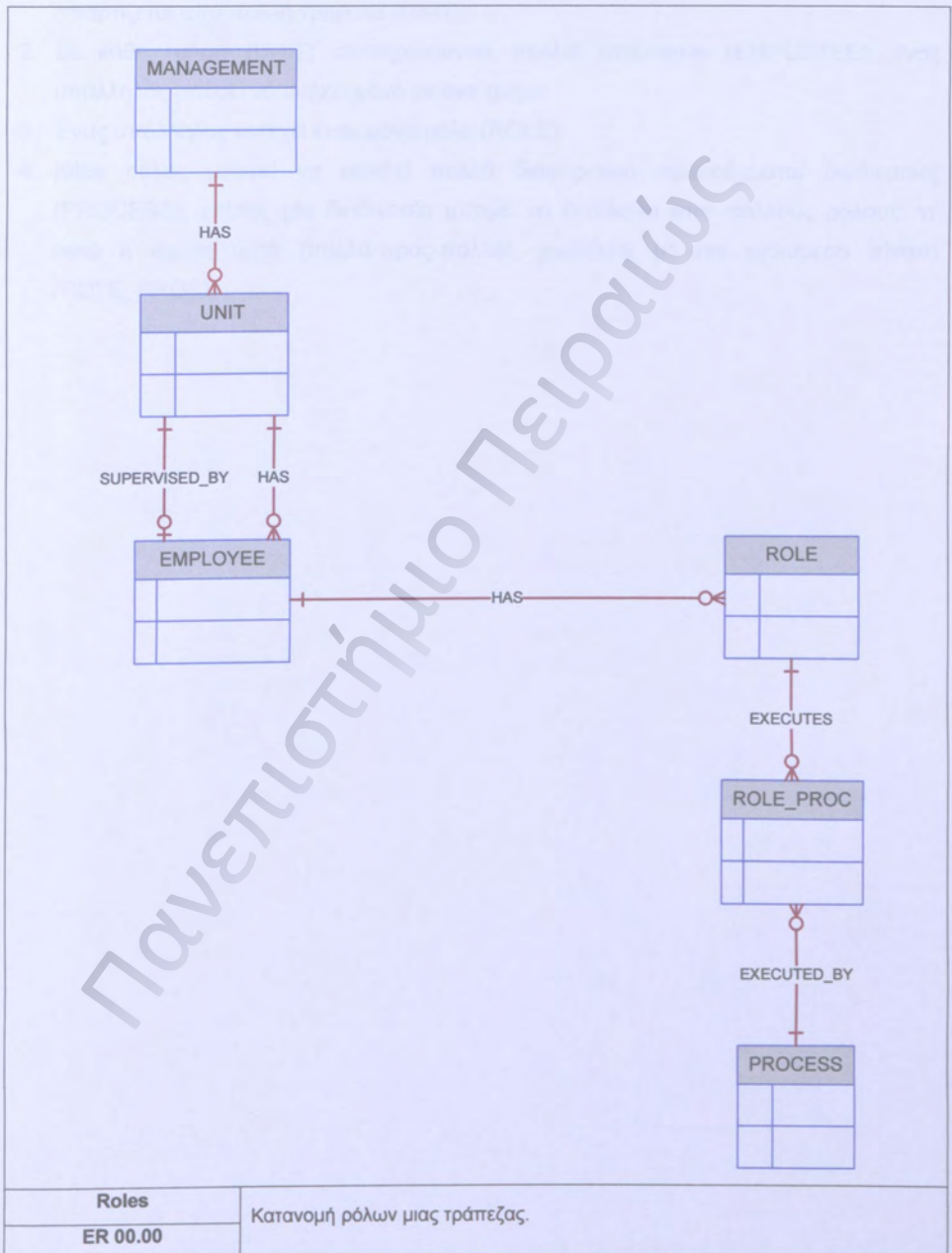
Πανεπιστήμιο Πειραιώς

3.2. Διαχείριση Ρόλων

Μία τράπεζα αποτελείται από πολλές διευθύνσεις, μία από αυτές είναι και η διεύθυνση Private Banking. Κάθε διεύθυνση απαρτίζεται από τμήματα ή μονάδες, σε κάθε τμήμα/μονάδα απασχολούνται διαφορετικές κατηγορίες υπάλληλων. Κάθε υπάλληλος που ανήκει σε ένα τμήμα κατέχει έναν συγκεκριμένο ρόλο. Ένας ρόλος έχει πρόσβαση σε συγκεκριμένα προγράμματα.

Πανεπιστήμιο Πειραιώς

3.2.1. Μοντέλο Οντοτήτων Συσχετίσεων

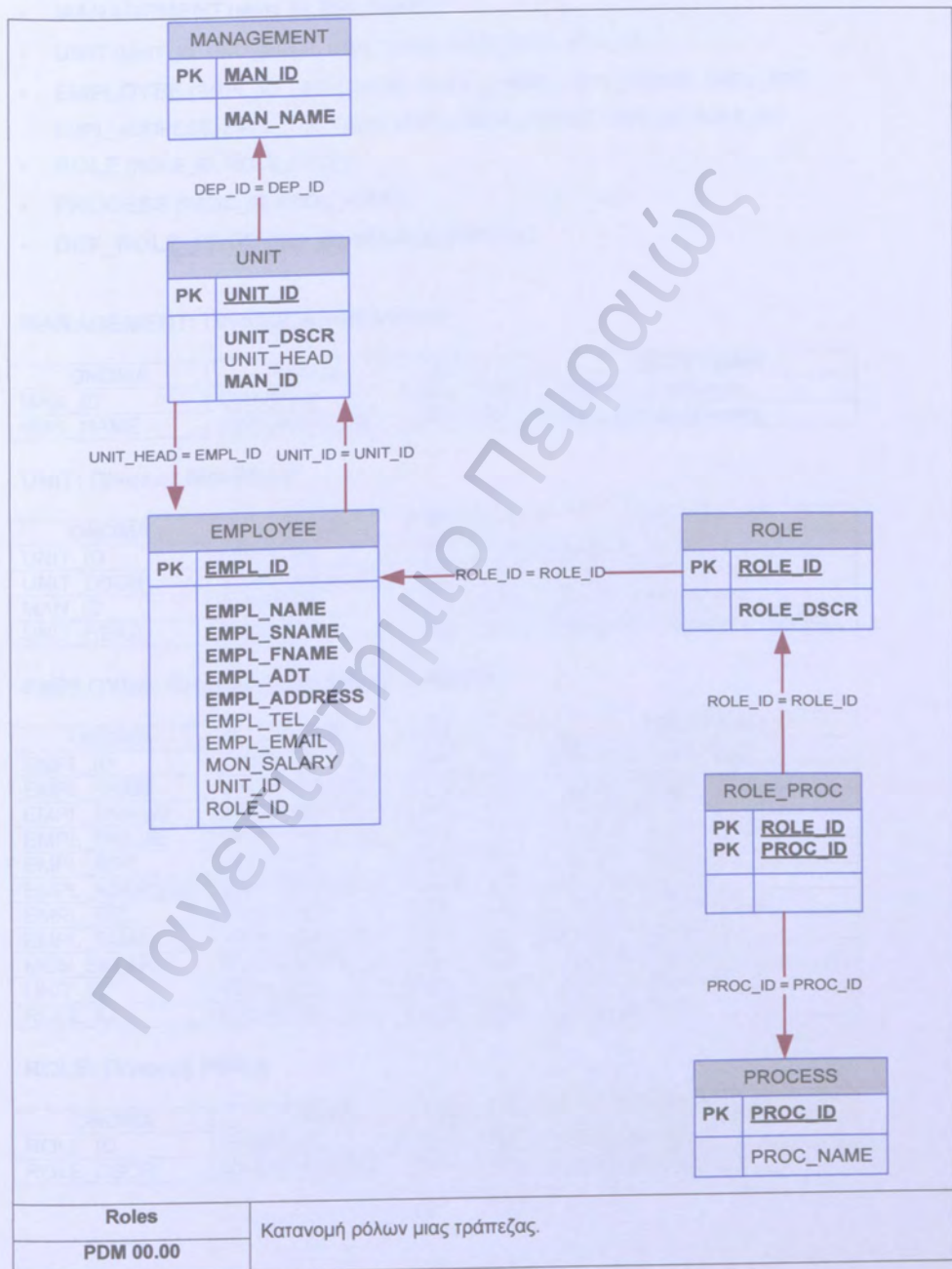


Σχήμα 4: Μοντέλο Οντοτήτων – Συσχετίσεων των ρόλων μιας τράπεζας.

Συνοπτικά:

1. Μία τράπεζα αποτελείται από πολλές διευθύνσεις (MANAGEMENT) κάθε διεύθυνση απαρτίζεται από πολλά τμήματα (UNIT).
2. Σε κάθε τμήμα (UNIT) απασχολούνται πολλοί υπάλληλοι (EMPLOYEE), ένας υπάλληλος μπορεί να ανήκει μόνο σε ένα τμήμα.
3. Ένας υπάλληλος κατέχει έναν μόνο ρόλο (ROLE).
4. Κάθε ρόλος μπορεί να εκτελεί πολλά διαφορετικά προγράμματα/ διαδικασίες (PROCESS), επίσης μία διαδικασία μπορεί να εκτελείται από πολλούς ρόλους, γι' αυτό η σχέση αυτή (πολλά-προς-πολλά), χωρίζεται με τον ενδιάμεσο πίνακα (ROLE_PROC).

3.2.2. Σχεσιακό Σχήμα



Σχήμα 5: Σχεσιακό Σχήμα ρόλων μιας τράπεζας.

Σύμφωνα με το σχεσιακό σχήμα (σχήμα 5) υπάρχουν οι ακόλουθοι σχεσιακοί πίνακες:

- **MANAGEMENT** (MAN_ID, DEP_NAME)
- **UNIT** (UNIT_ID, UNIT_DSCR, UNIT_HEAD, UNIT_HEAD, MAN_ID)
- **EMPLOYEE** (EMPL_ID, EMPL_NAME, EMPL_SNAME, EMPL_FNAME, EMPL_ADT, EMPL_ADDRESS, EMPL_TEL, EMPL_EMAIL, MON_SALARY, UNIT_ID, ROLE_ID)
- **ROLE** (ROLE_ID, ROLE_DSCR)
- **PROCESS** (PROC_ID, PROC_NAME)
- **DEP_ROLE_PROC** (DEP_ID, ROLE_ID, PROC_ID)

MANAGEMENT: Πίνακας Διευθύνσεων

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
MAN_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Διεύθυνσης
MAN_NAME	VARCHAR2(50)	ΟΧΙ	ΝΑΙ	Όνομασία Διεύθυνσης

UNIT: Πίνακας Μονάδων

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
UNIT_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Μονάδας
UNIT_DSCR	VARCHAR2(50)	ΟΧΙ	ΝΑΙ	Όνομασία Μονάδας
MAN_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός Διεύθυνσης
UNIT_HEAD	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός Υπεύθυνου Μονάδας

EMPLOYEE: Πίνακας Υπαλλήλων Τράπεζας

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
EMPL_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Υπαλλήλου
EMPL_NAME	VARCHAR2(50)	ΟΧΙ	ΝΑΙ	Όνομα Υπαλλήλου
EMPL_SNAME	VARCHAR2(50)	ΟΧΙ	ΝΑΙ	Επώνυμο Υπαλλήλου
EMPL_FNAME	VARCHAR2(50)	ΟΧΙ	ΝΑΙ	Πατρώνυμο Υπαλλήλου
EMPL_ADT	CHAR(10)	ΟΧΙ	ΝΑΙ	Α.Δ.Τ. Υπαλλήλου
EMPL_ADDRESS	VARCHAR2(50)	ΟΧΙ	ΟΧΙ	Δ/ση Υπαλλήλου
EMPL_TEL	CHAR(10)	ΟΧΙ	ΟΧΙ	Τηλέφωνο Υπαλλήλου
EMPL_EMAIL	VARCHAR2(50)	ΟΧΙ	ΟΧΙ	e-mail Υπαλλήλου
MON_SALARY	NUMEMBER(5,2)	ΟΧΙ	ΟΧΙ	Μηνιαίος Μισθός
UNIT_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός Μονάδας
ROLE_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός Ρόλου

ROLE: Πίνακας Ρόλων

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
ROLE_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Ρόλου
ROLE_DSCR	VARCHAR2(50)	ΟΧΙ	ΝΑΙ	Όνομασία Ρόλου

PROCESS: Πίνακας Διαδικασιών

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
PROC_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Διαδικασίας
PROC_DSCR	VARCHAR2(50)	ΟΧΙ	ΝΑΙ	Ονομασία Διαδικασίας

ROLE_PROC: Πίνακας Σχέσεων Ρόλων με Διαδικασίες

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
ROLE_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Ρόλου
PROC_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Διαδικασίας

3.3. Διαδικασία Προσέλκυσης Πελατείας Private Banking

Η διαδικασία προσέλκυσης πελατείας της διεύθυνσης Private Banking μιας τράπεζας ανατίθεται στους Relationship Managers - RM. Κάθε RM δημιουργεί μία προσωπική λίστα υποψήφιων πελατών (PER_CUST_LIST) με την επίβλεψη του επικεφαλής τμήματος Unit Head – UH στο οποίο ανήκει. Η προσωπική λίστα (PER_CUST_LIST) κάθε RM μπορεί να περιλαμβάνει πολλούς υποψήφιους πελάτες (PROSPECT_CUSTOMER) ένας υποψήφιος πελάτης όμως, θα πρέπει να ανήκει μόνο στη λίστα ενός RM. Στη συνέχεια, εφόσον οριστούν οι υποψήφιοι πελάτες για κάθε RM, ο ίδιος καθορίζει το χρονοπρόγραμμα των ενεργειών του (SCHEDULE) ανάλογα με το πελατολόγιό του. Το κάθε χρονοπρόγραμμα ελέγχεται από τον UH με στόχο την αναθεώρηση ή και προτεραιοποίηση των ενεργειών που περιλαμβάνει. Σύμφωνα με το αναθεωρημένο χρονοπρόγραμμα ο RM προετοιμάζει και διενεργεί την πρώτη επικοινωνία με τον υποψήφιο πελάτη και βάσει αυτής συμπληρώνει το ηλεκτρονικό αρχείο αναφοράς (CALL REPORT), που τηρείται για κάθε υποψήφιο πελάτη στο σύστημα. Το ηλεκτρονικό αυτό αρχείο συμπληρώνεται κάθε φορά που υπάρχει μία επικοινωνία με τον πελάτη.

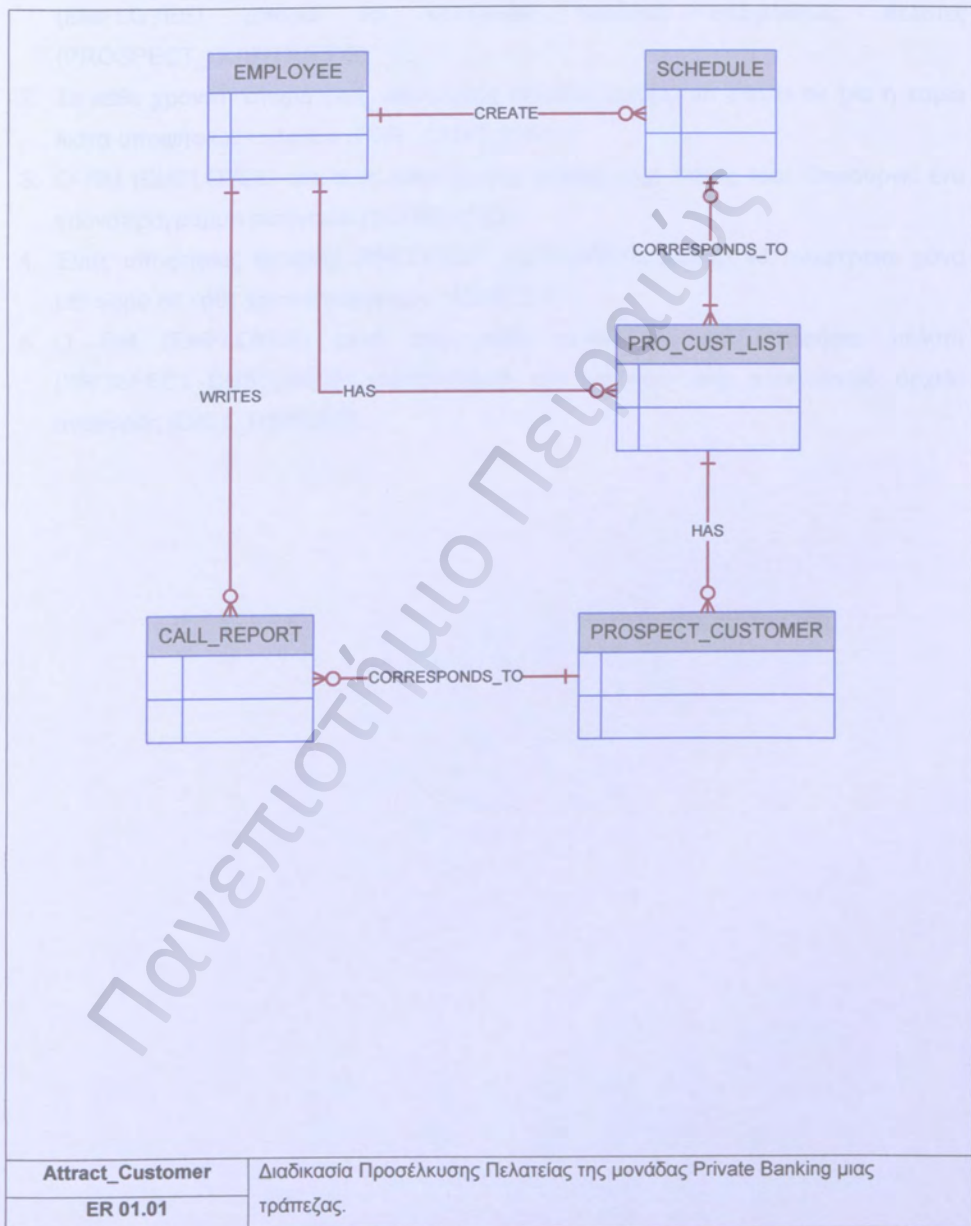
Μια από τις αρμοδιότητες του RM, αφού διενεργήσει την πρώτη επικοινωνία με τον υποψήφιο πελάτη, είναι να προτεραιοποιήσει τη λίστα των υποψήφιων πελατών του, με τους περισσότερο πιθανούς πελάτες (μέσα στο τρίμηνο) και συντονίζοντας τις επόμενες επισκέψεις μαζί τους. Στους υποψήφιους πελάτες αυτής της αναθεωρημένης λίστας ο RM θα παρουσιάσει επενδυτικές προτάσεις (PROPOSAL), ανάλογα με το επενδυτικό προφίλ του κάθε πελάτη. Για την κάλυψη αυτής της αρμοδιότητας ο RM καταφεύγει, όποτε είναι απαραίτητο, στη βοήθεια της Διεύθυνσης Επενδυτικής Στρατηγικής υποβάλλοντας αίτημα (REQUEST) σε εξειδικευμένα στελέχη της τράπεζας (Αναλυτές αγοράς, Dealers, Διαχειριστές αμοιβαίων κεφαλαίων, κ.τ.λ.). Κάθε τέτοιο αίτημα παραλαμβάνεται από ένα εξειδικευμένο στέλεχος στην αντίστοιχη κατηγορία επενδυτικών προϊόντων, ο οποίος αφού μελετήσει τα στοιχεία από το φάκελο του υποψήφιου πελάτη, συγγράφει τεκμηριωμένη αναφορά (REPORT) (παρέχοντας εξειδικευμένη γνώση και επενδυτική άποψη με παρουσιάσεις προϊόντων & επενδυτικών προτάσεων) την οποία στέλνει στον αιτούντα RM. Με βάση την αναφορά αυτή, ο RM ετοιμάζει μια επενδυτική πρόταση (PROPOSAL) την οποία και παρουσιάζει στον υποψήφιο πελάτη του.

Στην περίπτωση που ο υποψήφιος πελάτης δεχτεί την επενδυτική πρόταση (PROPOSAL), τότε ενεργοποιείται η διαδικασία αποδοχής πελάτη, σε διαφορετική

περίπτωση ο RM συμπληρώνει το αντίστοιχο αρχείο αναφοράς (CALL REPORT) του υποψήφιου πελάτη, ο οποίος παραμένει στην προσωπική λίστα (PRO_CUST_LIST) των υποψήφιων πελατών του RM, με στόχο τη μελλοντική προσέγγισή του.

Πανεπιστήμιο Πειραιώς

3.3.1. Μοντέλο Οντοτήτων Συσχετίσεων

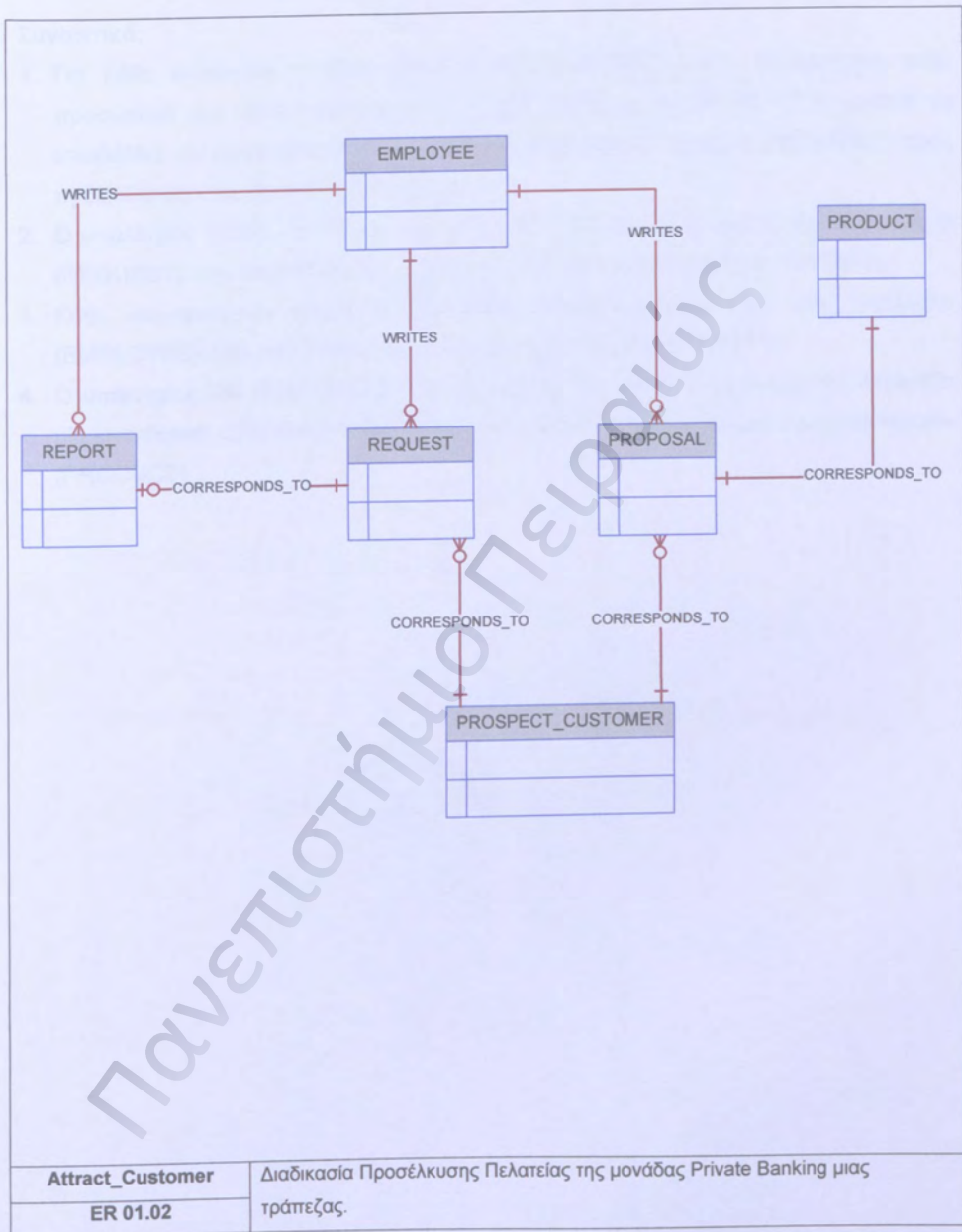


Σχήμα 6: Μοντέλο Οντοτήτων – Συσχετίσεων Διαδικασίας Προσέλκυσης Πελατείας (τμήμα 1^ο).

Συνοπτικά:

1. Σε κάθε μία προσωπική λίστα υποψήφιων πελατών (PER_CUST_LIST), ένας RM (EMPLOYEE) μπορεί να καταχωρεί πολλούς υποψήφιους πελάτες (PROSPECT_CUSTOMERS).
2. Σε κάθε χρονική στιγμή ένας υποψήφιος πελάτης μπορεί να ανήκει σε μία ή καμία λίστα υποψήφιων πελατών (PER_CUST_LIST).
3. Ο RM (EMPLOYEE) για τους υποψήφιους πελάτες της λίστας τους δημιουργεί ένα χρονοπρόγραμμα ενεργειών (SCHEDULE).
4. Ένας υποψήφιος πελάτης (PROSPECT_CUSTOMER) μπορεί να αναφέρεται μόνο μία φορά σε κάθε χρονοπρόγραμμα (SCHEDULE).
5. Ο RM (EMPLOYEE) μετά από κάθε επικοινωνία με υποψήφιο πελάτη (PROSPECT_CUSTOMER) συμπληρώνει μία εγγραφή στο ηλεκτρονικό αρχείο αναφοράς (CALL_REPORT).

Πανεπιστήμιο Πειραιώς

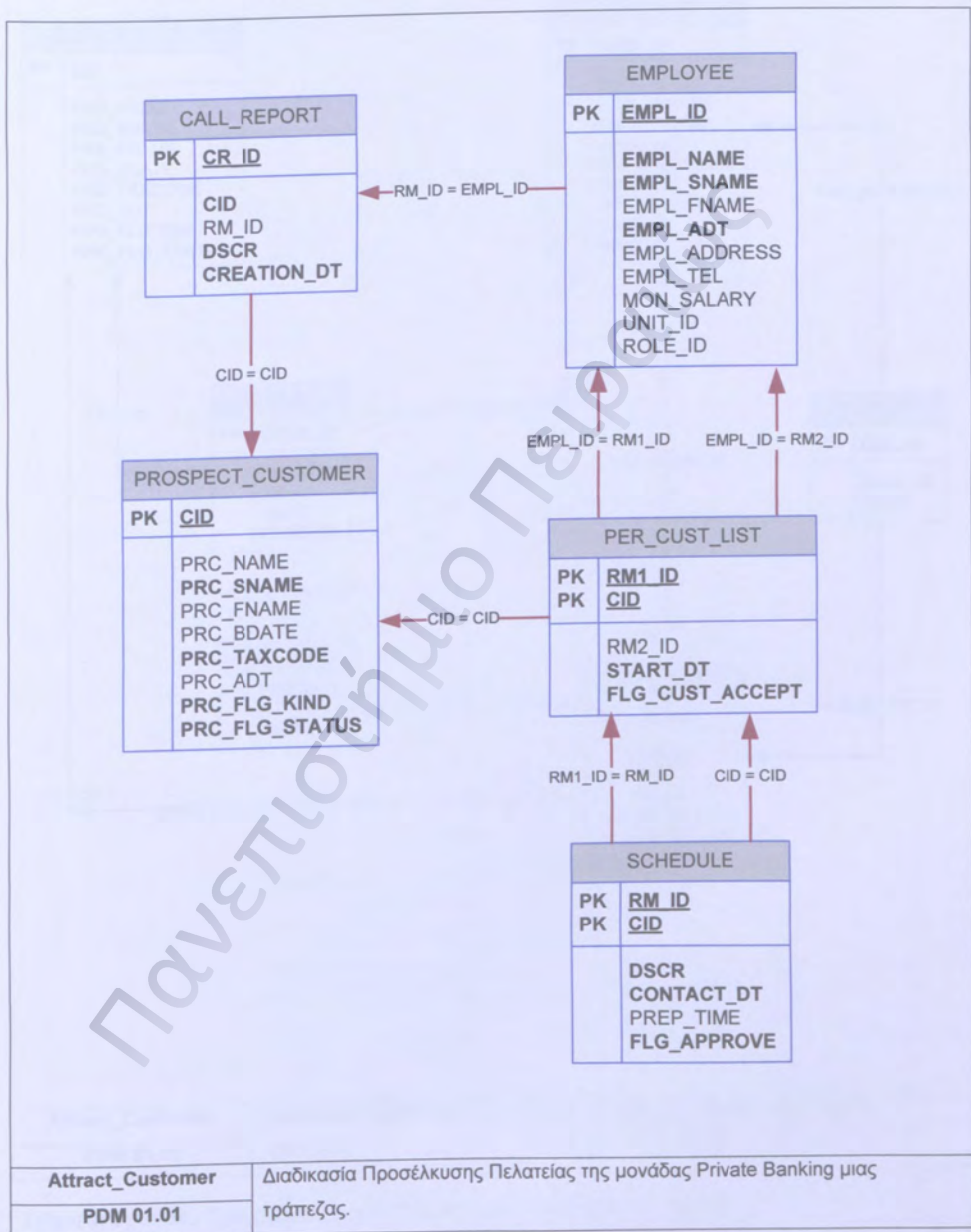


Σχήμα 7: Μοντέλο Οντοτήτων – Συσχετίσεων Διαδικασίας Προσέλκυσης Πελατείας (τμήμα 2^ο).

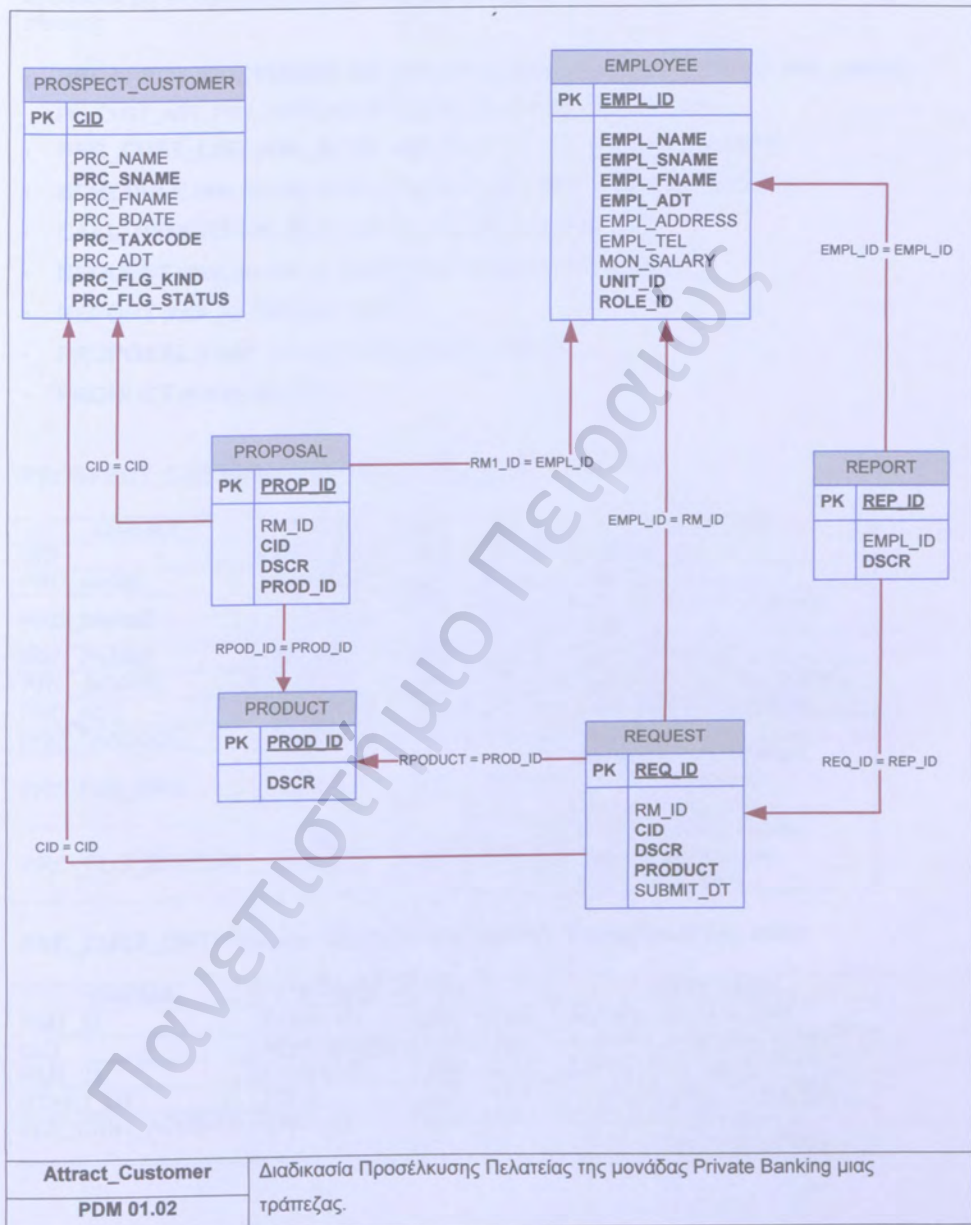
Συνοπτικά:

1. Για κάθε υποψήφιο πελάτη (PROSPECT_CUSTOMER) που εμπεριέχεται στην προσωπική του λίστα πελατών (PER_CUST_LIST), ο RM (EMPLOYEE) μπορεί να υποβάλλει, αν κρίνει απαραίτητο, ένα ή και περισσότερα αιτήματα (REQUEST) προς τη διεύθυνση Επενδυτικής Στρατηγικής.
2. Ο υπάλληλος (EMPLOYEE) του τμήματος Επενδυτικής Στρατηγικής για κάθε αίτημα (REQUEST) που παραλαμβάνει, γράφει μία τεκμηριωμένη αναφορά (REPORT).
3. Κάθε τεκμηριωμένη αναφορά (REPORT) παραλαμβάνεται από τον υπάλληλο (EMPLOYEE) που έχει υποβάλλει την αντίστοιχη αίτηση (REQUEST).
4. Ο υπάλληλος RM (EMPLOYEE) για κάθε υποψήφιο πελάτη της λίστας του ετοιμάζει μία επενδυτική πρόταση (PROPOSAL), κάθε τέτοια πρόταση αφορά ένα μόνο προϊόν (PRODUCT).

3.3.2. Σχεσιακό Σχήμα



Σχήμα 8: Σχεσιακό Σχήμα Διαδικασίας Προσέλκυσης Πελατείας (τμήμα 1^ο).



Σχήμα 9: Σχεσιακό Σχήμα Διαδικασίας Προσέλκυσης Πελατείας (τμήμα 2^ο).

Σύμφωνα με το σχεσιακό σχήμα (σχήμα 6 & σχήμα 7) υπάρχουν οι ακόλουθοι σχεσιακοί πίνακες:

- **PROSPECT_CUSTOMER** (CID, PRC_NAME, PRC_SNAME, PRC_FNAME, PRC_BDATE, PR_CUST_ADT, PRC_TAXCODE, PRC_FLG_KIND, PRC_FLG_STATUS)
- **PER_CUST_LIST** (RM1_ID, CID, RM2_ID, START_DT, FLG_CUST_ACCEPT)
- **SCHEDULE** (RM_ID, CID, DSCR, CONTACT_DT, PREP_TIME, FLG_APPROVE)
- **CALL_REPORT** (CR_ID, RM_ID, CID, DSCR, CREATION_DT)
- **REQUEST** (REQ_ID, RM_ID, CID, DSCR, PRODUCT, SUBMIT_DT)
- **REPORT** (REP_ID, EMPL_ID, DSCR)
- **PROPOSAL** (PROP_ID, RM_ID, CID, DSCR, PROD_ID)
- **PRODUCT** (PROD_ID, DSCR)

PROSPECT_CUSTOMER: Πίνακας Υποψήφιων Πελατών

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
CID	NUMBER(15)	ΝΑΙ	ΝΑΙ	Κωδικός πελάτη
PRC_NAME	CHAR(50)	ΟΧΙ	ΟΧΙ	Όνομα πελάτη
PRC_SNAME	CHAR(50)	ΟΧΙ	ΝΑΙ	Επώνυμο πελάτη/ Επωνυμία επιχείρησης
PRC_FNAME	CHAR(50)	ΟΧΙ	ΟΧΙ	Πατρώνυμο πελάτη
PRC_BDATE	DATE	ΟΧΙ	ΟΧΙ	Ημ/νία γεννήσεως/ δημιουργίας
PRC_ADT	CHAR(10)	ΟΧΙ	ΟΧΙ	Αριθμός δελτίου ταυτότητας
PRC_TAXCODE	CHAR(10)	ΟΧΙ	ΝΑΙ	Α.Φ.Μ.
PRC_FLG_KIND	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Κατηγορίας Πελάτη 0 = Φυσικό πρόσωπο, 1 = Νομικό πρόσωπο
PRC_FLG_STATUS	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Κατάστασης Πελάτη 0 = Μη καταχωρημένος, 1 = Καταχωρημένος

PER_CUST_LIST: Πίνακας Προσωπικής Λίστας Υποψήφιων Πελατών

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
RM1_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός πρωτεύον RM
CID	NUMBER(15)	ΝΑΙ	ΝΑΙ	Κωδικός υποψήφιου πελάτη
RM2_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός δευτερεύον RM
START_DT	DATE	ΟΧΙ	ΟΧΙ	Ημ/νία Έναρξης Συνεργασίας
FLG_CUST_ACCEPT	CHAR(1)	ΟΧΙ	ΟΧΙ	FLAG Συνεργασίας 0 = Απόρριψη, 1 = Αποδοχή

SCHEDULE: Πίνακας Χρονοπρογραμμάτων

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
RM_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός RM
CID	NUMBER(15)	ΝΑΙ	ΝΑΙ	Κωδικός υποψήφιου πελάτη
DSCR	VARCHAR2(250)	ΟΧΙ	ΝΑΙ	Περιγραφή
CONTACT_DT	DATE	ΟΧΙ	ΟΧΙ	Ημ/νία επικοινωνίας
PREP_TIME	NUMBER	ΟΧΙ	ΝΑΙ	Χρόνος προετοιμασίας
FLG_APPROVE	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG έγκρισης 0 = ΟΧΙ, 1 = ΝΑΙ

CALL_REPORT: Πίνακας Ηλεκτρονικού Αρχείου Πελατών

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
CR_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Αρχείου
CID	NUMBER(15)	ΟΧΙ	ΝΑΙ	Κωδικός Πελάτη
RM_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός RM
DSCR	VARCHAR2(250)	ΟΧΙ	ΝΑΙ	Περιγραφή
CREATION_DT	DATE	ΟΧΙ	ΝΑΙ	Ημ/νία δημιουργίας εγγραφής

REQUEST: Πίνακας Αιτημάτων RM

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
REQ_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός αιτήματος
RM_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός RM
CID	NUMBER(15)	ΟΧΙ	ΝΑΙ	Κωδικός πελάτη
DSCR	VARCHAR2(250)	ΟΧΙ	ΝΑΙ	Περιγραφή αιτήματος
PRODUCT	CHAR(10)	ΟΧΙ	ΝΑΙ	Κωδικός προϊόντος
SUBMIT_DT	DATE	ΟΧΙ	ΝΑΙ	Ημ/νία υποβολής αιτήματος

REPORT: Πίνακας Αναφορών Διεύθυνσης Επενδυτικής Στρατηγικής

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
REP_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός αναφοράς
EMPL_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός υπαλλήλου
DSCR	VARCHAR2(250)	ΟΧΙ	ΝΑΙ	Περιγραφή

PROPOSAL: Πίνακας Επενδυτικών Προτάσεων σε Υποψήφιους Πελάτες

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
PROP_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός επενδυτικής πρότασης
RM_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός RM
CID	NUMBER(15)	ΟΧΙ	ΝΑΙ	Κωδικός πελάτη
DSCR	VARCHAR2(250)	ΟΧΙ	ΝΑΙ	Περιγραφή
PROD_ID	CHAR(10)	ΟΧΙ	ΝΑΙ	Κωδικός Προϊόντος

PRODUCT: Πίνακας προϊόντων Private Banking

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
PROD_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός προϊόντος
DSCR	VARCHAR2(250)	ΟΧΙ	ΝΑΙ	Περιγραφή

3.4. Αποδοχή Πελάτη *Private Banking*

Η διαδικασία αποδοχής υποψήφιου πελάτη είναι δεύτερη κατά σειρά και τίθεται σε λειτουργία εφόσον ο πελάτης επιθυμεί συνεργασία με τη διεύθυνση *Private Banking* της. Για κάθε τέτοιο υποψήφιο πελάτη που περιλαμβάνεται στην προσωπική λίστα υποψήφιων πελατών ο RM συμπληρώνει αίτηση ανοίγματος ή τροποποίησης του ηλεκτρονικού αρχείου πελάτη (PB_CIF_REQUEST). Εάν ο υποψήφιος πελάτης δεν έχει προσκομίσει κάποιο στοιχείο ταυτοποίησης ή λείπει κάποιο από τον φάκελό του, τότε καταχωρείται στο σύστημα προειδοποίηση (WARNING), στην οποία αναγράφεται η ημερομηνία πιθανής διεκπεραίωσης της. Επίσης, σε κάθε αίτηση (PB_CIF_REQUEST) αντιστοιχεί και μία λίστα συμβατικών εγγράφων, σύμφωνα με τις υπηρεσίες που επιθυμεί να αποκτήσει ο υποψήφιος πελάτης. Κάθε τέτοια αίτηση ανοίγματος ή τροποποίησης CIF (PB_CIF_REQUEST) εγκρίνεται από τον UH, όταν δεν εκκρεμούν για αυτήν καταχωρημένες προειδοποιήσεις (WARNING).

Για κάθε υποψήφιο πελάτη ο RM συμπληρώνει μία αίτηση PCA (Prospect Client Approval). Η φόρμα αυτή αποστέλλεται ηλεκτρονικά στον επικεφαλής της μονάδας UH, ο οποίος ελέγχει για κάθε πελάτη τα στοιχεία της φόρμας (PCA) ως προς την πληρότητα των κριτηρίων αποδοχής πελάτη (suitability criteria) με βάση τις εκάστοτε πολιτικές του *Private Banking* και αφού συμπληρώσει παρατηρήσεις ή σχόλια, εγκρίνει ή απορρίπτει την αίτηση PCA. Η αίτηση PCA εγκρίνεται επίσης και από τον επικεφαλής του τμήματος CDU άλλα και από τον

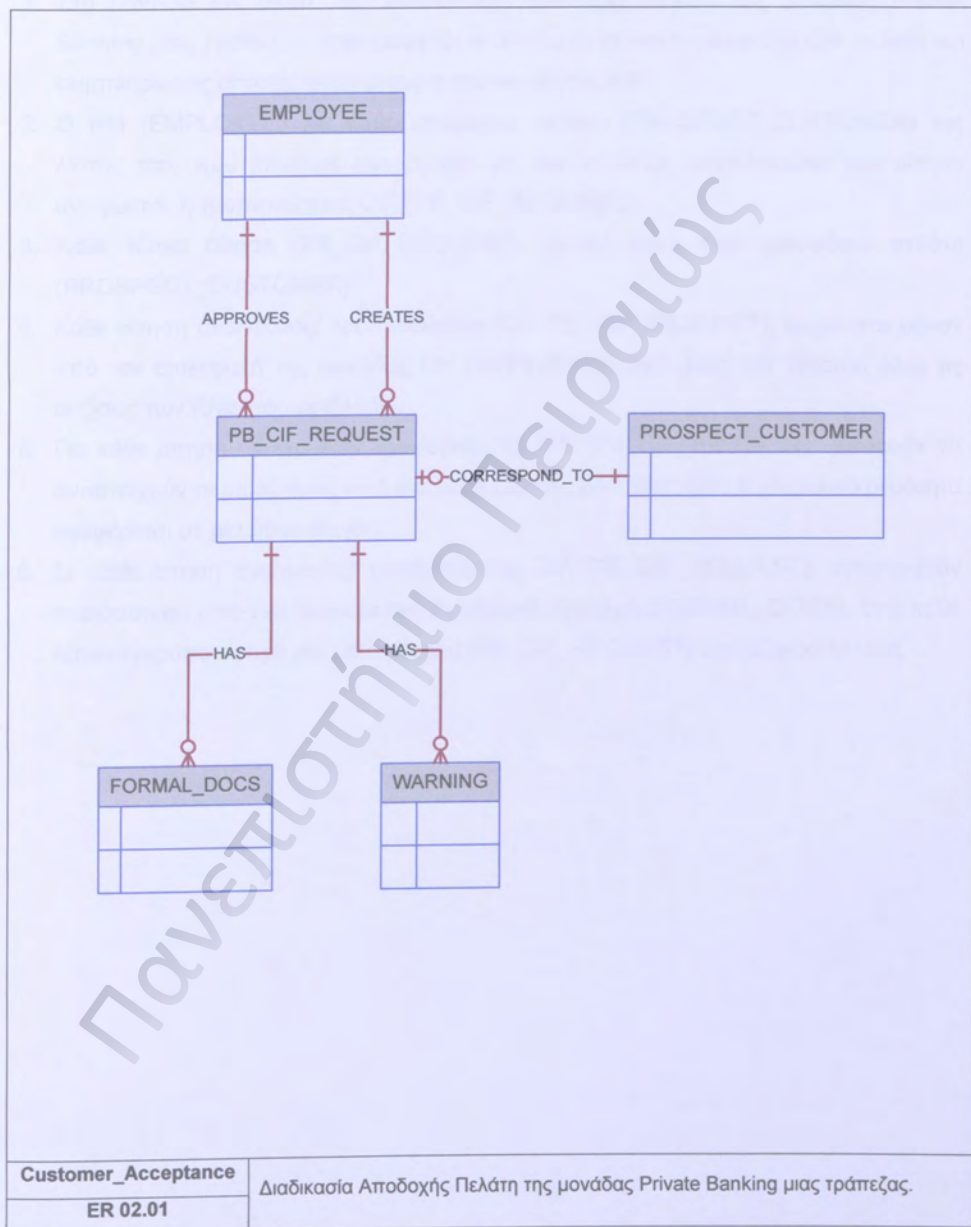
καταγράφοντας τυχόν σχόλια στο αρχείο (PCA) και το προωθεί στον BCCH, στην περίπτωση που υπάρχουν ελλείψεις στον φάκελο του πελάτη αυτές καταγράφονται στις λίστες εκκρεμοτήτων και εκκρεμοτήτων εγκεκριμένων πελατών. Με τον ίδιο τρόπο ο BCCH παρακολουθεί σε οθόνη του συστήματος τα εγκεκριμένα (PCAs) των πελατών και αφού παραλάβει τον φάκελο του πελάτη ελέγχει τα ακόλουθα:

Στο τελικό στάδιο αυτής της διαδικασίας το αρμόδιο στέλεχος του τμήματος CDU CDU_EXECUTIVE κρίνει για κάθε πελάτη εάν απαιτείται ή όχι καταχώριση (VSV) και αντίστοιχα αποστέλλει, σε ηλεκτρονική μορφή, αίτημα δημιουργίας VSV (CREATE_VSV_REQ) προς το τμήμα τήρησης και διαχείρισης μητρώου πελατών MURC, στο ίδιο τμήμα αποστέλλει με συστημένη εσωτερική αλληλογραφία το πιστοποιητικό ταυτοποίησης του πελάτη. Στο τμήμα MURC ο υπεύθυνος υπάλληλος

MURC_EMPLOYEE αφού παραλάβει τα έγγραφα του πελάτη πραγματοποιεί καταχώρηση στον πίνακα (VSV) και στη συνέχεια αποστέλλει πίσω στο τμήμα CDU με εσωτερική αλληλογραφία, το πιστοποιητικό ταυτοποίησης του πελάτη.

Πανεπιστήμιο Πειραιώς

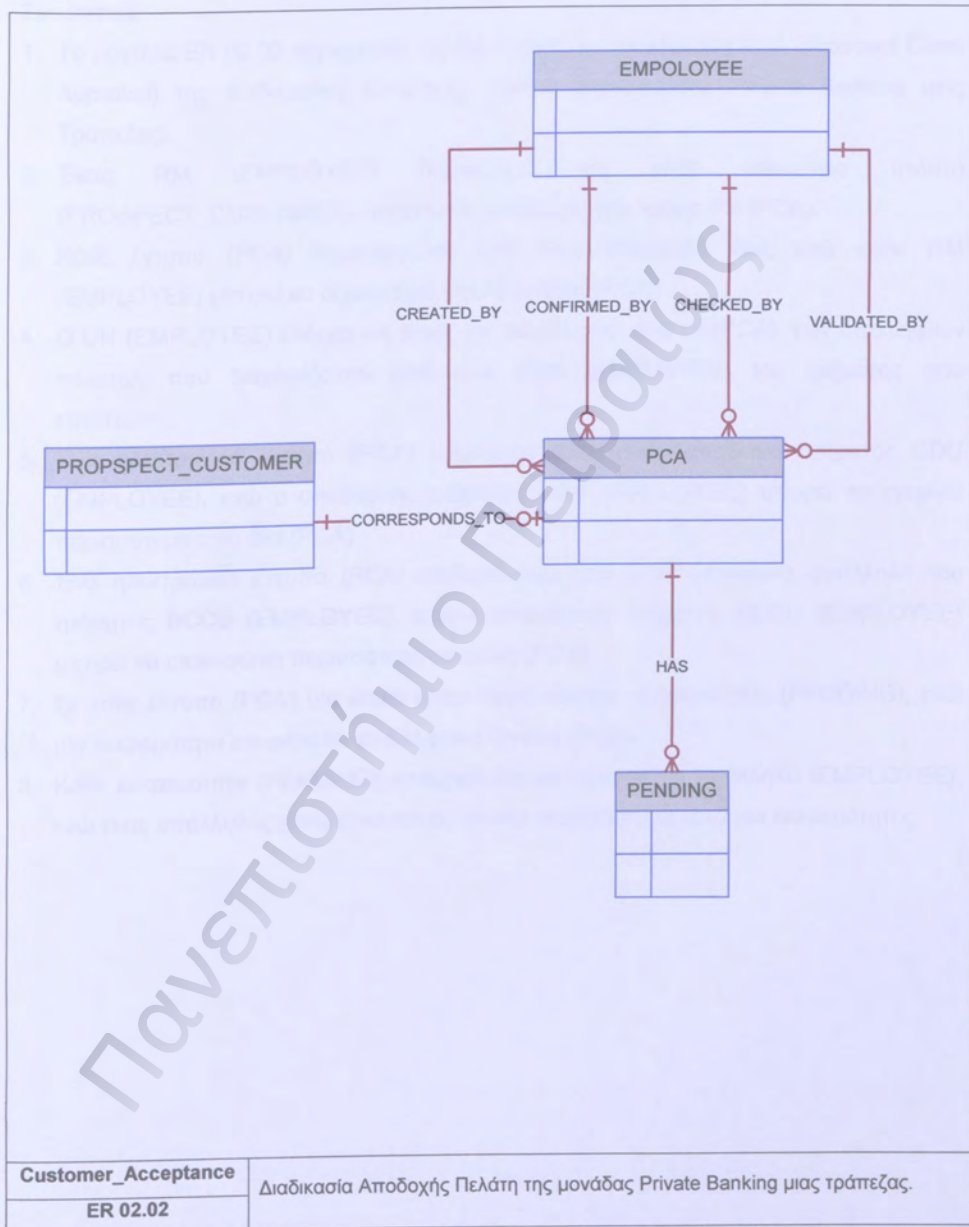
3.4.1. Μοντέλο Οντοτήτων Συσχετίσεων



Σχήμα 10: Μοντέλο Οντοτήτων – Συσχετίσεων Διαδικασίας Αποδοχής Πελάτη (τμήμα 1^ο).

Συνοπτικά:

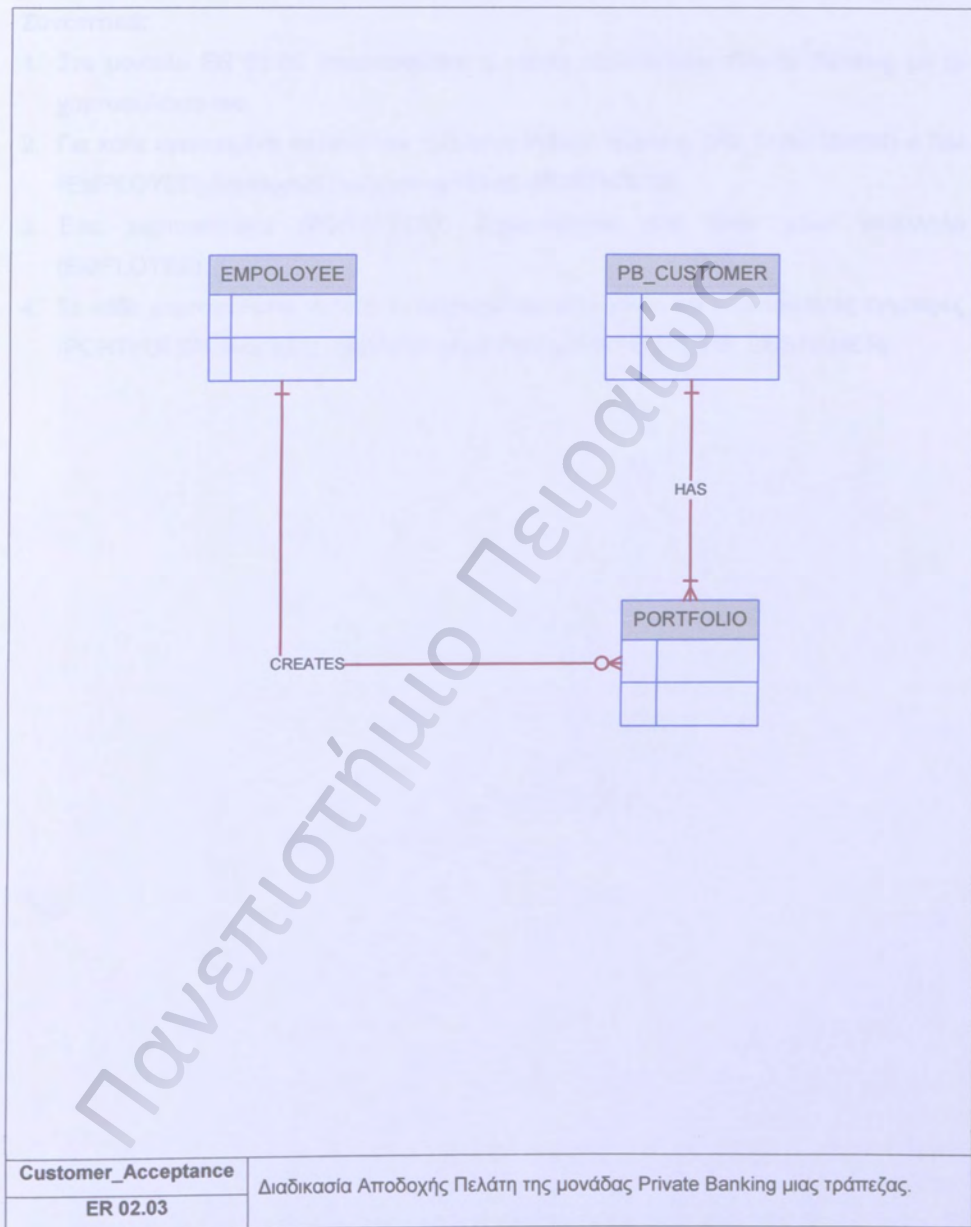
1. Στο μοντέλο ER 02.01 της διαδικασίας Αποδοχής Πελάτη της μονάδας Private Banking μιας Τράπεζας, περιγράφεται το στάδιο διερεύνησης ύπαρξης CIF πελάτη και συμπλήρωσης αίτησης ανοίγματος ή τροποποίησης CIF.
2. Ο RM (EMPLOYEE) για κάθε υποψήφιο πελάτη (PROSPECT_CUSTOMER) της λίστας του, που επιθυμεί συνεργασία με την τράπεζα, συμπληρώνει μία αίτηση ανοίγματος ή τροποποίησης CIF (PB_CIF_REQUEST).
3. Κάθε τέτοια αίτηση (PB_CIF_REQUEST), αφορά μόνο έναν υποψήφιο πελάτη (PROSPECT_CUSTOMER).
4. Κάθε αίτηση ανοίγματος/ τροποποίησης CIF (PB_CIF_REQUEST), εγκρίνεται μόνον από τον επικεφαλής της μονάδας UH (EMPLOYEE), ενώ ένας UH εγκρίνει όλες τις αιτήσεις των RMs που επιβλέπει.
5. Για κάθε αίτηση ανοίγματος/ τροποποίησης CIF (PB_CIF_REQUEST), μπορούν να αντιστοιχούν περισσότερες από μία εκκρεμότητες (WARNINGS). Ενώ μία εκκρεμότητα αναφέρεται σε μία μόνο αίτηση.
6. Σε κάθε αίτηση ανοίγματος/ τροποποίησης CIF (PB_CIF_REQUEST), αντιστοιχούν περισσότερα από ένα διαφορετικά συμβατικά έγγραφα (FORMAL_DOCS), ενώ κάθε τέτοιο έγγραφο αφορά μία μόνο αίτηση (PB_CIF_REQUEST) υποψήφιου πελάτη.



Σχήμα 11: Μοντέλο Οντοτήτων – Συσχετίσεων Διαδικασίας Αποδοχής Πελάτη (τμήμα 2^ο).

Συνοπτικά:

1. Το μοντέλο ER 02.02 περιγράφει την δημιουργία και επικύρωση PCA (Prospect Client Approval) της διαδικασίας Αποδοχής Πελάτη της μονάδας Private Banking μιας Τράπεζας.
2. Ένας RM (EMPLOYEE) δημιουργεί, για κάθε υποψήφιο πελάτη (PROSPECT_CUSTOMER) ένα έντυπο αποδοχής στο τμήμα PB (PCA).
3. Κάθε έντυπο (PCA) δημιουργείται από έναν υπάλληλο RM, ενώ κάθε RM (EMPLOYEE) μπορεί να δημιουργεί πολλά έντυπα (PCA).
4. Ο UH (EMPLOYEE) ελέγχει ως προς την πληρότητα, όλα τα (PCA), των υποψήφιων πελατών που διαχειρίζονται από τους RMs (EMPLOYEE) του τμήματος που εμποττεύει.
5. Ένα ηλεκτρονικό έντυπο (PCA) εγκρίνεται από έναν υπεύθυνο τμήματος CDU (EMPLOYEE), ενώ ο υπεύθυνος τμήματος CDU (EMPLOYEE) μπορεί να εγκρίνει περισσότερα από ένα (PCA)
6. Ένα ηλεκτρονικό έντυπο (PCA) επικυρώνεται από έναν υπεύθυνο υπάλληλο του τμήματος BCCU (EMPLOYEE), ενώ ο υπεύθυνος τμήματος BCCU (EMPLOYEE) μπορεί να επικυρώσει περισσότερα από ένα (PCA)
7. Σε κάθε έντυπο (PCA) μία καμία ή και περισσότερες εκκρεμότητες (PENDING), ενώ μία εκκρεμότητα αναφέρεται σε ένα μόνο έντυπο (PCA).
8. Κάθε εκκρεμότητα (PENDING) καταχωρείται από ένα μόνο υπάλληλο (EMPLOYEE), ενώ ένας υπάλληλος μπορεί να καταχωρήσει περισσότερες από μία εκκρεμότητες.

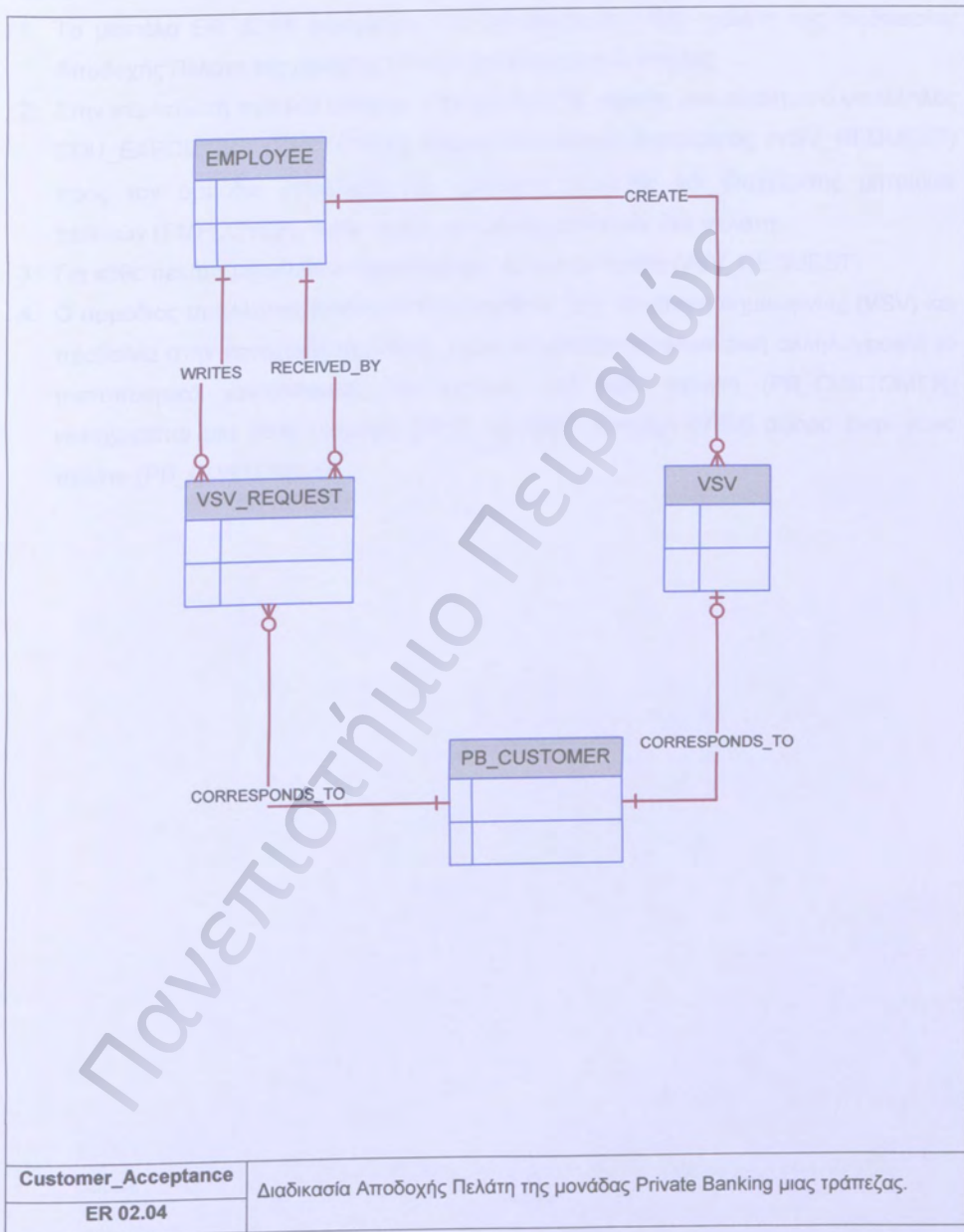


Σχήμα 12: Μοντέλο Οντοτήτων – Συσχετίσεων Διαδικασίας Αποδοχής Πελάτη (τμήμα 3^ο).

Συνοπτικά:

1. Στο μοντέλο ER 02.03 παρουσιάζεται η σχέση κάθε πελάτη Private Banking με το χαρτοφυλάκιο του.
2. Για κάθε εγκεκριμένο πελάτη του τμήματος Private Banking (PB_CUSTOMER) ο RM (EMPLOYEE) δημιουργεί ένα χαρτοφυλάκιο (PORTFOLIO).
3. Ένα χαρτοφυλάκιο (PORTFOLIO) δημιουργείται από έναν μόνο υπάλληλο (EMPLOYEE).
4. Σε κάθε χαρτοφυλάκιο πελάτη καταχωρούνται από μία ή και περισσότερες εγγραφές (PORTFOLIO), ενώ κάθε εγγραφή αφορά έναν μόνο πελάτη (PB_CUSTOMER).

Πανεπιστήμιο Πειραιώς

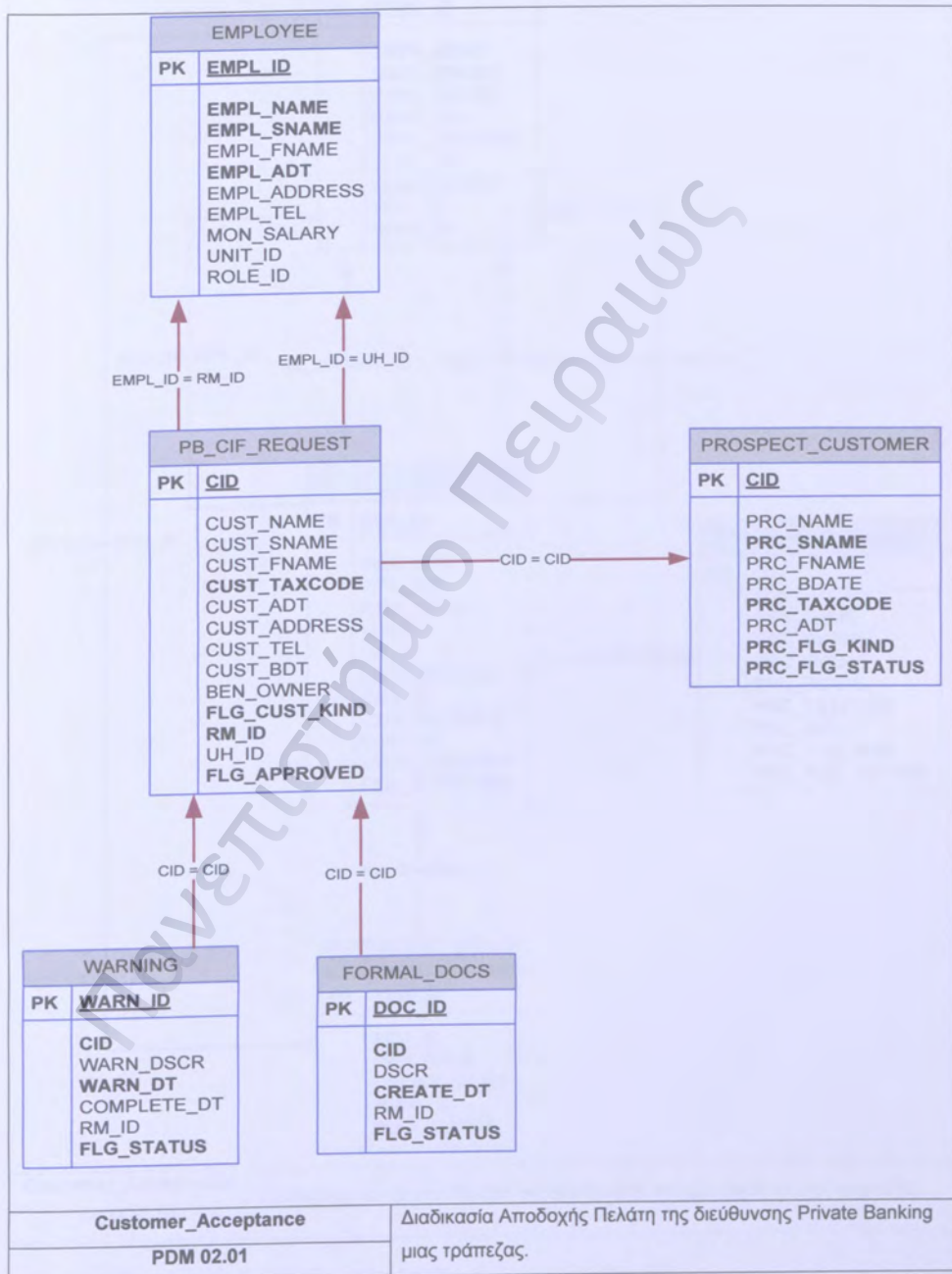


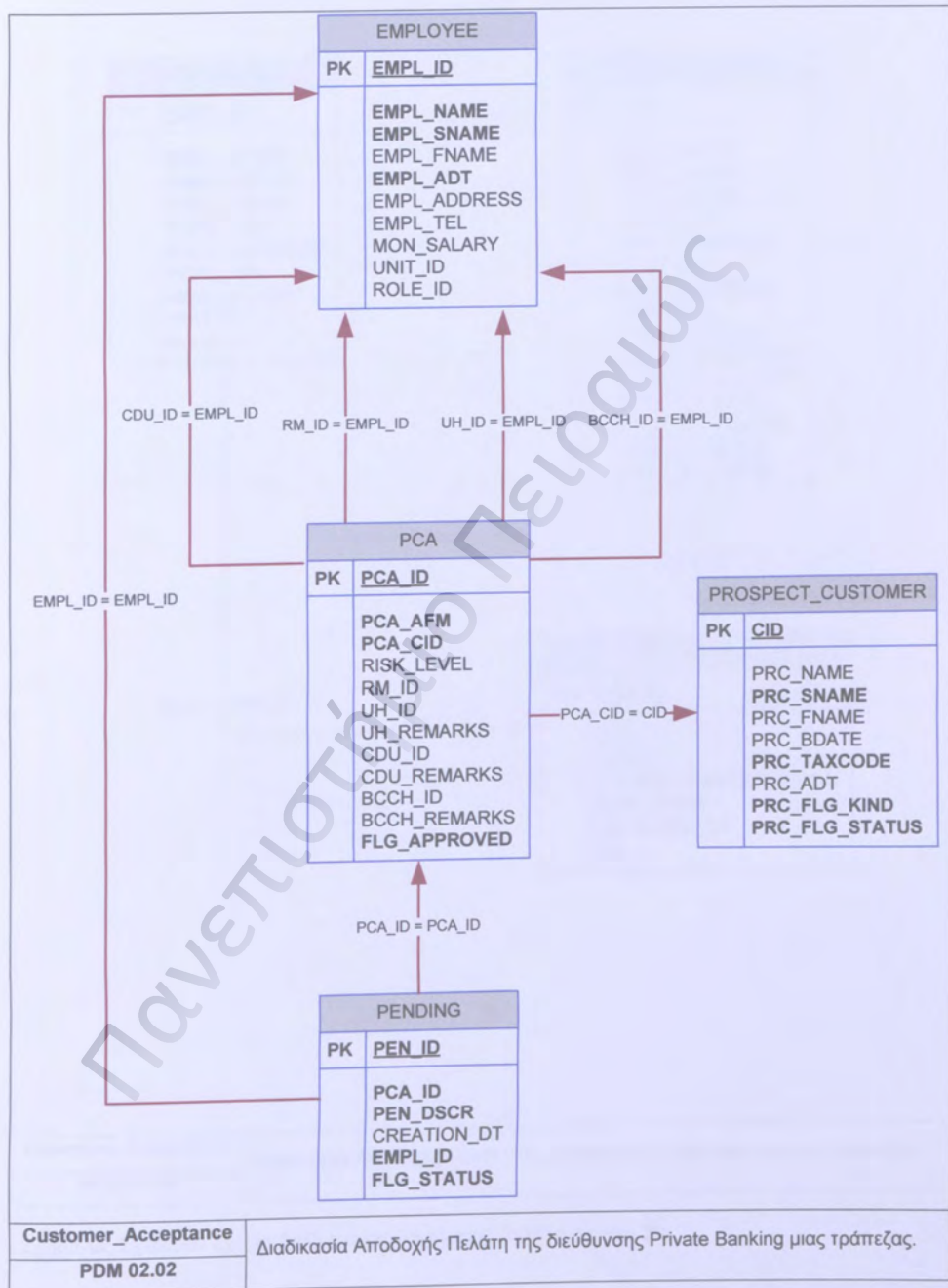
Σχήμα 13: Μοντέλο Οντοτήτων – Συσχετίσεων Διαδικασίας Αποδοχής Πελάτη (τμήμα 4^ο).

Συνοπτικά:

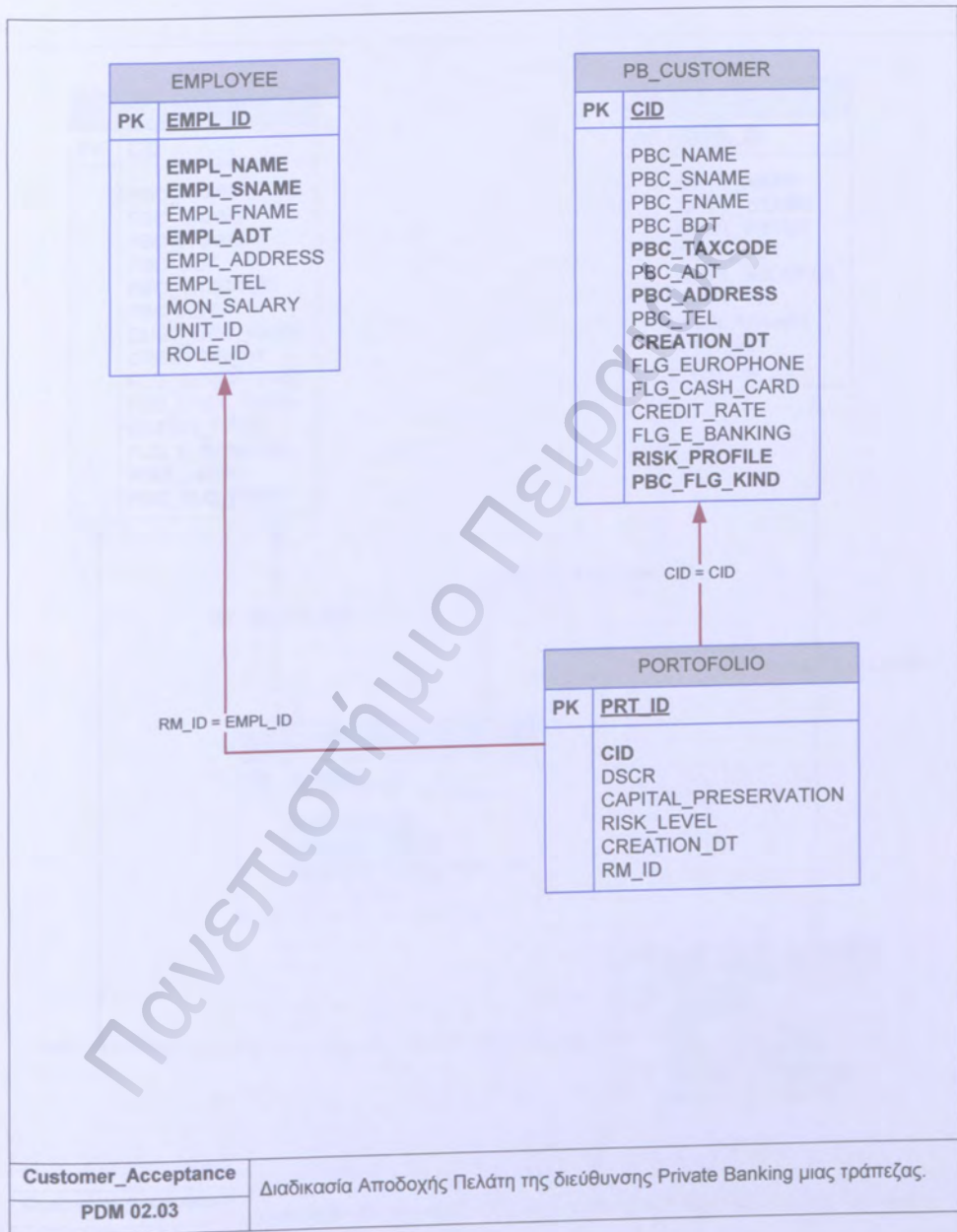
1. Το μοντέλο ER 02.04 περιγράφει την καταχώρηση (VSV) πελάτη της διαδικασίας Αποδοχής Πελάτη της μονάδας Private Banking μιας Τράπεζας.
2. Στην περίπτωση που δεν υπάρχει VSV για τον PB_πελάτη στο σύστημα ο υπάλληλος CDU_EXECUTIVE (EMPLOYEE) αποστέλλει αίτημα δημιουργίας (VSV_REQUEST) προς τον αρμόδιο υπάλληλο του τμήματος τήρησης και διαχείρισης μητρώων πελατών (EMPLOYEE). Κάθε τέτοιο αίτημα αφορά μόνον ένα πελάτη.
3. Για κάθε πελάτη μπορούν να αποσταλούν πολλά αιτήματα (VSV_REQUEST)
4. Ο αρμόδιος υπάλληλος (EMPLOYEE) λαμβάνει όλα τα αίτημα δημιουργίας (VSV) και προβαίνει στην καταχώρηση (VSV), αφού παραλάβει με εσωτερική αλληλογραφία το πιστοποιητικό ταυτοποίησης του πελάτη. Για κάθε πελάτη (PB_CUSTOMER) καταχωρείται μία μόνο εγγραφή (VSV) και κάθε εγγραφή (VSV) αφορά έναν μόνο πελάτη (PB_CUSTOMER).

3.4.2. Σχεσιακό Σχήμα

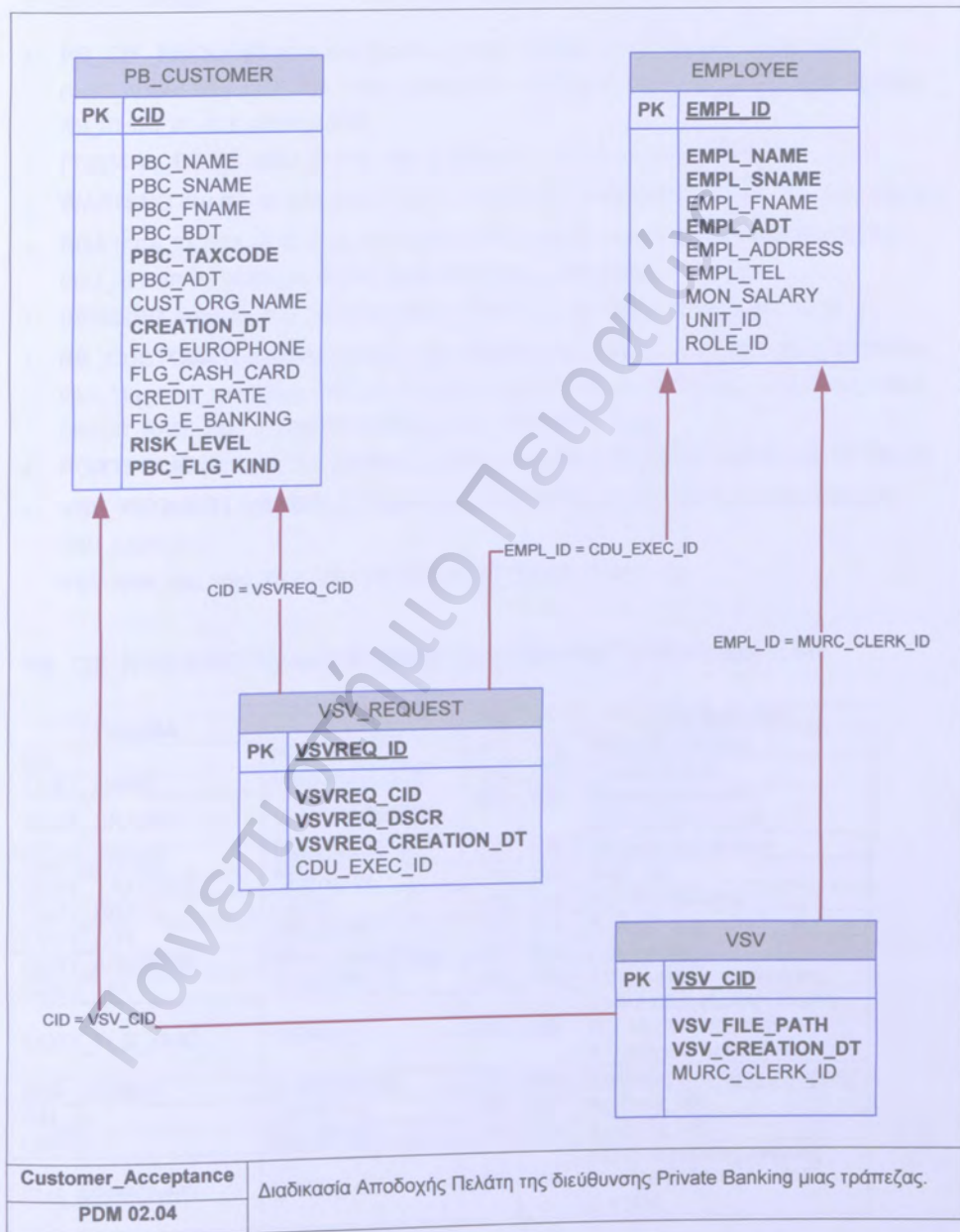
Σχήμα 14: Σχεσιακό Σχήμα Διαδικασίας Αποδοχής Πελάτη (τμήμα 1^ο).



Σχήμα 15: Σχεσιακό Σχήμα Διαδικασίας Αποδοχής Πελάτη (τμήμα 2^ο).



Σχήμα 16: Σχεσιακό Σχήμα Διαδικασίας Αποδοχής Πελάτη (τμήμα 3^ο).



Σχήμα 17: Σχεσιακό Σχήμα Διαδικασίας Αποδοχής Πελάτη (τμήμα 4^ο).

Σύμφωνα με το σχεσιακό σχήμα (σχήμα 14, 15, 16, & 17) υπάρχουν οι ακόλουθοι σχεσιακοί πίνακες:

- **PB_CIF_REQUEST** (CID, CUST_NAME, CUST_SNAME, CUST_FNAME, CUST_BDT, CUST_ADDRESS, CUST_TEL, CUST_TAXCODE, CUST_ADT, CUST_FLG_KIND, BEN_OWNER, RM_ID, UH_ID, FLG_APPROVED)
- **FORMAL_DOCS** (DOC_ID, CID, DSCR, CREATE_DT, RM_ID, FLG_STATUS)
- **WARNING** (WARN_ID, CID, WARN_DSCR, WARN_DT, COMPLETE_DT, FLG_STATUS, RM_ID)
- **PCA** (PCA_ID, PCA_AFM, PCA_CID, RISK_LEVEL, RM_ID, UH_ID, UH_REMARKS, CDU_ID, CDU_REMARKS, BCCH_ID, BCCH_REMARKS, FLG_APPROVED)
- **PENDING** (PEN_ID, PCA_ID, PEN_DSCR, CREATION_DT, EMPL_ID, FLG_STATUS)
- **PB_CUSTOMER** (CID, PBC_NAME, PBC_SNAME, PBC_FNAME, PBC_BDT, PBC_ADDRESS, PBC_TEL, PBC_TAXCODE, PBC_ADT, CREATION_DT, FLG_EUROPHONE, FLG_CASH_CARD, CREDIT_RATE, FLG_E_BANKING, RISK_LEVEL, PBC_FLG_KIND)
- **PORTFOLIO** (PRT_ID, CID, CAPITAL_PRESERVATION, RISK_LEVEL, CREATION_DT, RM_ID)
- **VSV_REQUEST** (VSVREQ_ID, VSVREQ_CID, VSVREQ_DSCR, VSVREQ_CREATION_DT, CDU_EXEC_ID)
- **VSV** (VSV_CID, VSV_FILE, VSV_CREATION_DT, MURC_CLERK_ID)

PB_CIF_REQUEST: Πίνακας Αιτήσεων Δημιουργίας/ Τροποποίησης CIF

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
CID	NUMBER(15)	ΝΑΙ	ΝΑΙ	Κωδικός Αίτησης
CUST_NAME	VARCHAR2(50)	ΟΧΙ	ΟΧΙ	Όνομα πελάτη
CUST_SNAME	VARCHAR2(50)	ΟΧΙ	ΝΑΙ	Επώνυμο πελάτη / Επωνυμία Εταιρίας
CUST_FNAME	VARCHAR2(50)	ΟΧΙ	ΟΧΙ	Πατρώνυμο πελάτη
CUST_TAXCODE	CHAR(15)	ΟΧΙ	ΝΑΙ	Α. Φ. Μ.
CUST_BDATE	DATE	ΟΧΙ	ΟΧΙ	Ημ/νία Γέννησης
CUST_ADT	CHAR(10)	ΟΧΙ	ΟΧΙ	Α. Δ. Τ.
CUST_ADDRESS	VARCHAR2(150)	ΟΧΙ	ΝΑΙ	Διεύθυνση
CUST_TEL	NUMBER(15)	ΟΧΙ	ΝΑΙ	Τηλέφωνο Επικοινωνίας
CUST_FLG_KIND	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Κατηγορίας Πελάτη 0 = Φυσικό πρόσωπο, 1 = Νομικό πρόσωπο
BEN_OWNER	NUMBER(15)	ΟΧΙ	ΝΑΙ	Κωδικός Ιδιοκτήτη (εταιρεία)
RM_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός RM
UH_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός UH
FLG_APPROVED	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Έγκρισης Αίτησης 0 = ΑΝΑΜΟΝΗ, 1 = ΟΧΙ, 2 = ΝΑΙ

FORMAL_DOCS: Πίνακας Συμβατικών Εγγράφων

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
DOC_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Εγγράφου
CID	CHAR(10)	ΟΧΙ	ΝΑΙ	Κωδικός Υποψήφιου Πελάτη
DSCR	VARCHAR2(150)	ΟΧΙ	ΝΑΙ	Περιγραφή
CREATE_DT	DATE	ΟΧΙ	ΝΑΙ	Ημ/νία Καταχώρησης
RM_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός RM
FLG_STATUS	CHAR(1)	ΟΧΙ	ΝΑΙ	Κατάσταση εγγράφου 0 = ενεργό, 1 = ανενεργό

WARNING: Πίνακας Προειδοποιήσεων

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
WARN_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Προειδοποίησης
CID	CHAR(10)	ΟΧΙ	ΝΑΙ	Κωδικός Υποψήφιου Πελάτη
WARN_DSCR	VARCHAR2(150)	ΟΧΙ	ΝΑΙ	Περιγραφή
WARN_DT	DATE	ΟΧΙ	ΝΑΙ	Ημ/νία Προειδοποίησης
COMPLETE_DT	DATE	ΟΧΙ	ΝΑΙ	Ημ/νία πιθανής ολοκλήρωσης
FLG_STATUS	CHAR(1)	ΟΧΙ	ΝΑΙ	Κατάσταση προειδοποίησης 0 = ενεργή, 1 = ανενεργή
RM_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός RM

PCA: Πίνακας PCA

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
PCA_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Εντύπου PCA
PCA_AFM	CHAR(15)	ΟΧΙ	ΝΑΙ	A. Φ. Μ.
PCA_CID	NUMBER(15)	ΟΧΙ	ΝΑΙ	Κωδικός Πελάτη
RISK_LEVEL	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Επενδυτικού Κινδύνου 0 = Χαμηλό, 1 = Μεσαίο, 2 = Υψηλό
RM_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός RM
UH_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός UH
UH_REMARKS	VARCHAR2(250)	ΟΧΙ	ΟΧΙ	Σχόλια UH
CDU_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός CDU
CDU_REMARKS	VARCHAR2(250)	ΟΧΙ	ΟΧΙ	Σχόλια CDU
BCCH_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός BCCH
BCCH_REMARKS	VARCHAR2(250)	ΟΧΙ	ΟΧΙ	Σχόλια BCCH
FLG_APPROVED	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Έγκρισης Πελάτη 0 = Εκκρεμότητα, 1 = Αποδοχή, 2 = Έγκριση με εκκρεμότητες 3 = Απόρριψη

PENDING: Πίνακας Εκκρεμοτήτων Εντύπου PCA

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
PEN_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Εκκρεμότητας
PCA_ID	CHAR(10)	ΟΧΙ	ΝΑΙ	Κωδικός PCA
PEN_DSCR	VARCHAR2(150)	ΟΧΙ	ΝΑΙ	Περιγραφή
CREATION_DT	DATE	ΟΧΙ	ΝΑΙ	Ημ/νία Καταχώρησης
EMPL_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός υπαλλήλου
FLG_STATUS	CHAR(1)	ΟΧΙ	ΝΑΙ	Κατάσταση Εκκρεμότητας 0 = ενεργή, 1 = ανενεργή

PB_CUSTOMER: Πίνακας Πελατών Private Banking

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
CID	NUMBER(15)	ΝΑΙ	ΝΑΙ	Κωδικός Πελάτη
PBC_NAME	VARCHAR2(50)	ΟΧΙ	ΟΧΙ	Όνομα Πελάτη
PBC_SNAME	VARCHAR2(50)	ΟΧΙ	ΝΑΙ	Επώνυμο/ Πελάτη / Επωνυμία εταιρίας
PBC_FNAME	VARCHAR2(50)	ΟΧΙ	ΟΧΙ	Πατρώνυμο Πελάτη
PBC_BDATE	DATE	ΟΧΙ	ΟΧΙ	Ημέρια Γέννησης
PBC_TAXCODE	CHAR(15)	ΟΧΙ	ΝΑΙ	Α. Φ. Μ.
PBC_ADT	VARCHAR2(50)	ΟΧΙ	ΟΧΙ	Α. Δ. Τ.
PBC_ADDRESS	VARCHAR2(250)	ΟΧΙ	ΝΑΙ	Διεύθυνση
PBC_TEL	NUMBER(15)	ΟΧΙ	ΝΑΙ	Τηλέφωνο Επικοινωνίας
CREATION_DT	CHAR(10)	ΟΧΙ	ΝΑΙ	Ημέρια Δημιουργίας
FLG_EUROPHONE	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Υπηρεσίας Europhone 0 = ΟΧΙ, 1 = ΝΑΙ
FLG_CASH_CARD	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Υπηρεσίας Cash Card 0 = ΟΧΙ, 1 = ΝΑΙ
CREDIT_RATE	NUMBER	ΟΧΙ	ΝΑΙ	Ποσό Πίστωσης
FLG_E_BANKING	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Υπηρεσίας e Banking 0 = ΟΧΙ, 1 = ΝΑΙ
RISK_PROFILE	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Προφίλ Επενδυτικού Κινδύνου 0 = Χαμηλό, 1 = Μεσαίο 2 = Υψηλό
PBC_FLG_KIND	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Κατηγορίας Πελάτη 0 = Φυσικό πρόσωπο, 1 = Νομικό πρόσωπο

PORTFOLIO: Πίνακας Χαρτοφυλακίων

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
PRT_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Χαρτοφυλακίου
CID	NUMBER(15)	ΟΧΙ	ΝΑΙ	Κωδικός πελάτη
CAPITAL	NUMBER	ΟΧΙ	ΝΑΙ	Κεφάλαιο
RISK_LEVEL	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Επενδυτικού Κινδύνου 0 = Χαμηλό, 1 = Μεσαίο 2 = Υψηλό
CREATION_DT	DATE	ΟΧΙ	ΝΑΙ	Ημέρια Δημιουργίας
RM_ID	CHAR(10)	ΟΧΙ	ΝΑΙ	Κωδικός RM

VSV_REQUEST: Πίνακας Αιτημάτων δημιουργίας VSV

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
VSVREQ_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός
VSVREQ_CID	NUMBER(15)	ΟΧΙ	ΝΑΙ	Κωδικός Πελάτη
VSVREQ_DSCR	VARCHAR2(250)	ΟΧΙ	ΝΑΙ	Περιγραφή
VSVREQ_CREATION_DT	DATE	ΟΧΙ	ΝΑΙ	Ημέρια καταχώρησης
CDU_EXEC_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός CDU_EXEC

VSV: Πίνακας VSV

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
VSV_CID	NUMBER(15)	ΝΑΙ	ΝΑΙ	Κωδικός Πελάτη
VSV_FILE_PATH	VARCHAR2(250)	ΟΧΙ	ΝΑΙ	Path Αρχείου
VSV_CREATION_DT	DATE	ΟΧΙ	ΝΑΙ	Ημ/νία Καταχώρησης
MURC_CLERK_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός MURK_CLERK

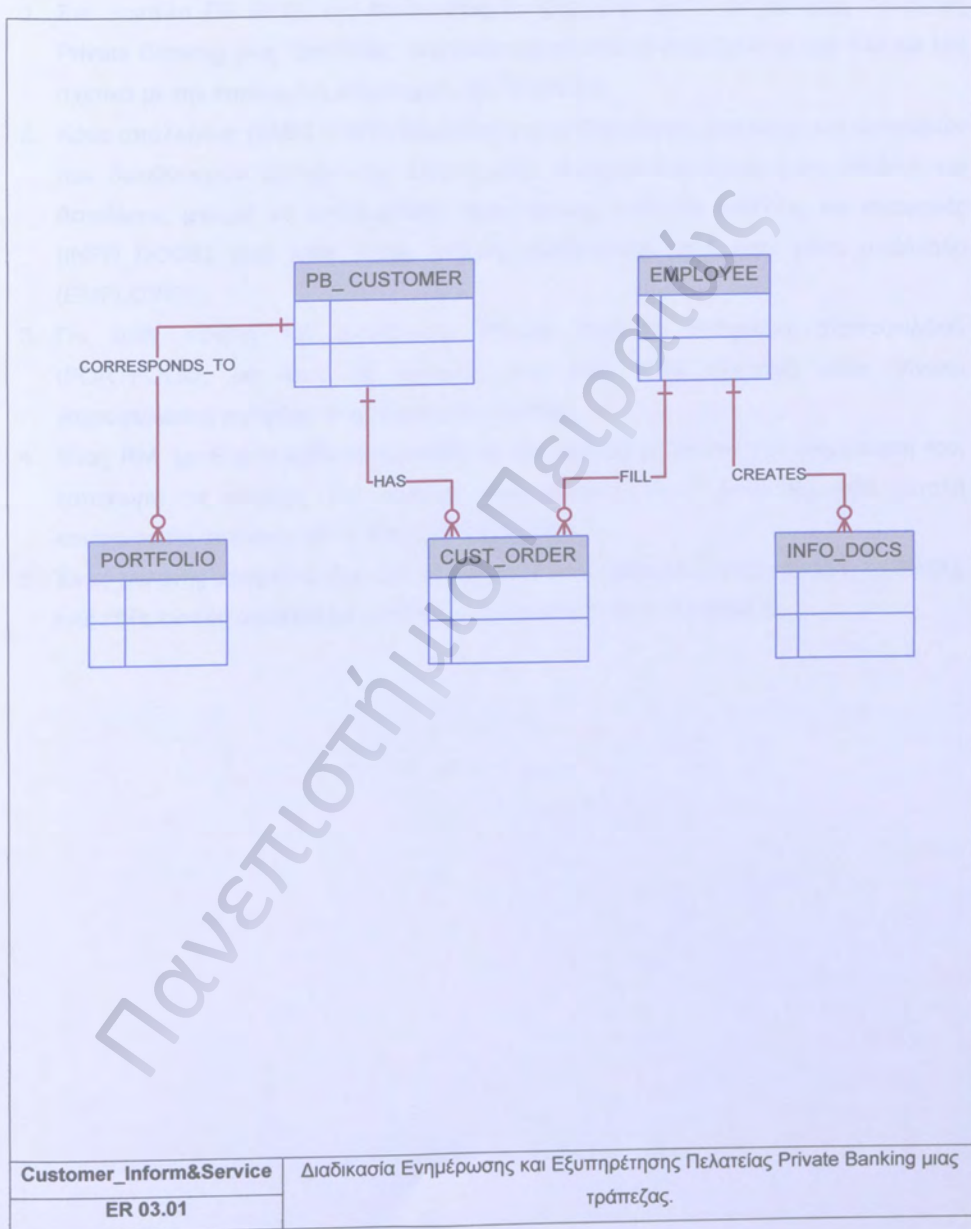
CIF: Πίνακας Ηλεκτρονικών Αρχείων Πελατών της Τράπεζας

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
CID	NUMBER(15)	ΝΑΙ	ΝΑΙ	Κωδικός Πελάτη
NAME	VARCHAR2(50)	ΟΧΙ	ΟΧΙ	Όνομα Πελάτη
SNAME	VARCHAR2(50)	ΟΧΙ	ΟΧΙ	Επώνυμο Πελάτη
FNAME	VARCHAR2(50)	ΟΧΙ	ΟΧΙ	Πατρώνυμο
TAXCODE	CHAR(15)	ΟΧΙ	ΝΑΙ	Α. Φ. Μ.
ADT	CHAR(10)	ΟΧΙ	ΟΧΙ	ΑΔΤ
ADDRESS	VARCHAR2(250)	ΟΧΙ	ΝΑΙ	Διεύθυνση
TEL	NUMBER(15)	ΟΧΙ	ΝΑΙ	Τηλέφωνο Επικοινωνίας
CREATION_DT	DATE	ΟΧΙ	ΝΑΙ	Ημ/νία Δημιουργίας
LAST_DEP	CHAR(10)	ΟΧΙ	ΝΑΙ	Τμήμα στο οποίο ανήκει
RISK_PROFILE	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Επενδυτικό προφίλ πελάτη 0 = Χαμηλό, 1 = Μεσαίο, 2 = Υψηλό
FLG_KIND	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Κατηγορίας Πελάτη 0 = Φυσικό πρόσωπο, 1 = Νομικό πρόσωπο

3.5. Ενημέρωση & Εξυπηρέτηση Πελατείας Private Banking

Στη διαδικασία ενημέρωσης και εξυπηρέτησης πελατείας, οι αρμόδιοι υπάλληλοι από τις Διευθύνσεις Επενδυτικής Στρατηγικής, Διεθνών Κεφαλαιαγορών (Global Markets), ΑΕΔΑΚ και Securities, καταχωρούν ενημερωτικά δελτία (εκθέσεις και αναφορές), τα οποία αφορούν την επιλεγμένη στρατηγική της Τράπεζας. Ο RM αφού παραλάβει και εκτυπώσει τα ενημερωτικά δελτία, ελέγχει τα χαρτοφυλάκια των πελατών του για να διαπιστωθεί η συμβατότητα του επιπέδου ρίσκου τους με το Risk Profile του πελάτη. Στη συνέχεια επικοινωνεί με τον πελάτη για ενημέρωση και συμπληρώνει το Call Report, εάν ο πελάτης δώσει κάποια εντολή, τότε αυτή καταχωρείται στον πίνακα CUST_ORDER, για να χρησιμοποιηθεί μετέπειτα από τη διαδικασία Λήψης και Διαβίβασης Εντολών.

3.5.1. Μοντέλο Οντοτήτων Συσχετίσεων

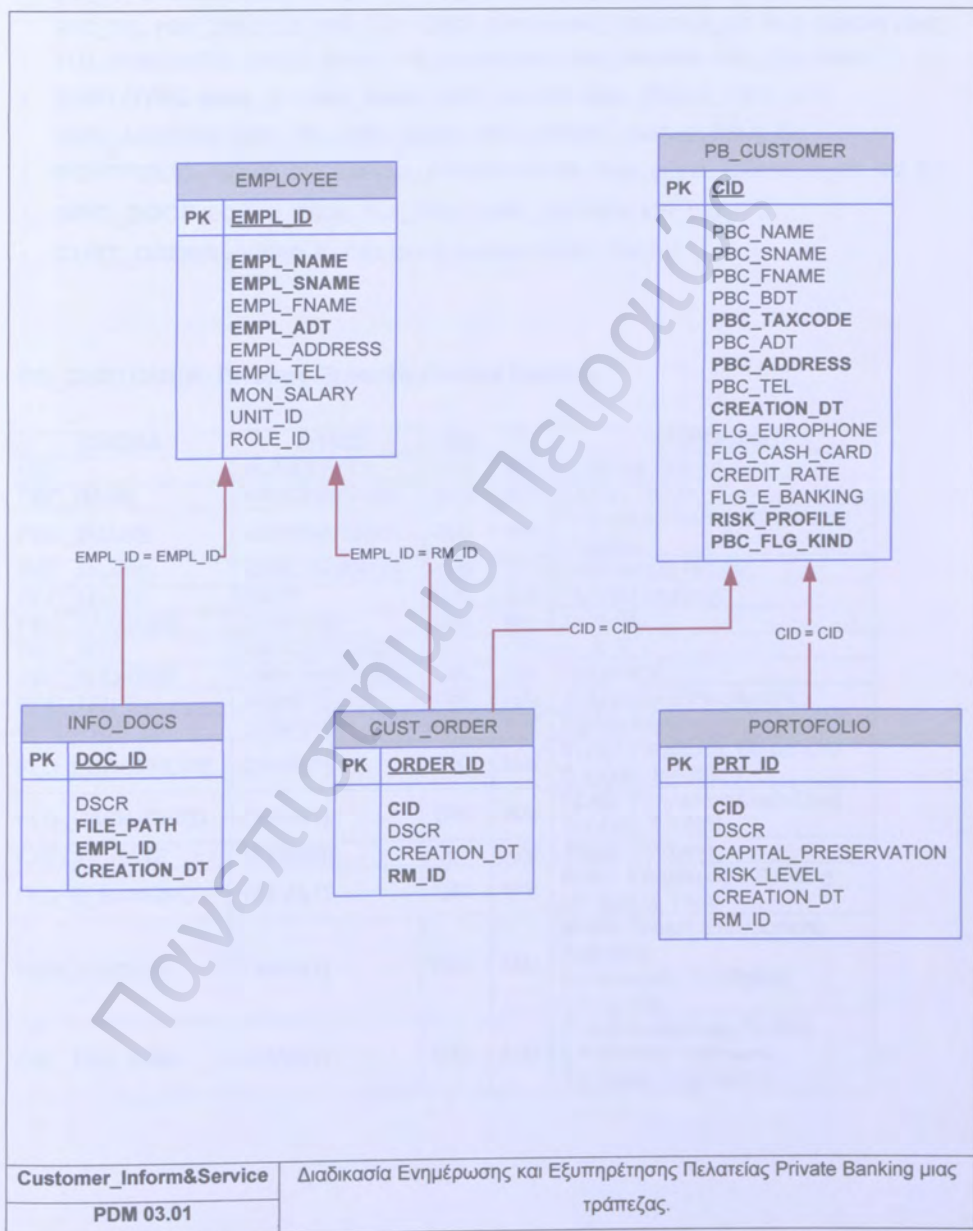


Σχήμα 18: Μοντέλο Οντοτήτων – Συσχετίσεων Διαδικασίας Ενημέρωσης & Εξυπηρέτησης Πελατείας.

Συνοπτικά:

1. Στο μοντέλο ER 03.01 της διαδικασίας Ενημέρωσης και Εξυπηρέτησης Πελατείας Private Banking μιας Τράπεζας, περιγράφεται το στάδιο ενημέρωσης των RM και UH σχετικά με την επιλεγμένη στρατηγική της Τράπεζας
2. Κάθε υπάλληλος (EMPLOYEE) αρμόδιος για τη δημιουργία εκθέσεων και αναφορών των διευθύνσεων Επενδυτικής Στρατηγικής, Διεθνών Κεφαλαιαγορών, ΑΕΔΑΚ και Ασφάλειας μπορεί να καταχωρήσει περισσότερες από μία εκθέσεις και αναφορές (INFO_DOCS), ενώ κάθε τέτοια έκθεση καταχωρείται από έναν μόνο υπάλληλο (EMPLOYEE).
3. Για κάθε πελάτη της Διεύθυνσης Private Banking διατηρείται χαρτοφυλάκιο (PORTFOLIO) με όλες τις κινήσεις του. Ενώ κάθε εγγραφή στον πίνακα χαρτοφυλακίου αναφέρεται σε έναν μόνο πελάτη.
4. Ένας RM μετά από κάθε επικοινωνία με τον πελάτη με σκοπό την ενημέρωση του, καταχωρεί τις εντολές του πελάτη στον πίνακα (CUST_ORDER). Μία εντολή καταχωρείται από έναν μόνο RM (EMPLOYEE).
5. Ένας πελάτης μπορεί να έχει δώσει περισσότερες από μία εντολές (CUST_ORDER), ενώ κάθε εντολή αναφέρεται σε έναν μόνο πελάτη (PB_CUSTOMER).

3.5.2. Σχεσιακό Σχήμα



Σχήμα 19: Σχεσιακό Σχήμα Διαδικασίας Ενημέρωσης και Εξυπηρέτησης Πελατείας.

Σύμφωνα με το σχεσιακό σχήμα (σχήμα 19) υπάρχουν οι ακόλουθοι σχεσιακοί πίνακες:

- **PB_CUSTOMER** (CID, PBC_NAME, PBC_SNAME, PBC_FNAME, PBC_BDT, PBC_ADDRESS, PBC_TEL, PBC_TAXCODE, PBC_ADT, CUST_ORG_NAME, CREATION_DT, FLG_EUROPHONE, FLG_CASH_CARD, CREDIT_RATE, FLG_E_BANKING, RISK_PROFILE, PBC_FLG_KIND)
- **EMPLOYEE** (EMPL_ID, EMPL_NAME, EMPL_SNAME, EMPL_FNAME, EMPL_ADT, EMPL_ADDRESS, EMPL_TEL, EMPL_EMAIL, MON_SALARY, UNIT_ID, ROLE_ID)
- **PORTFOLIO** (PRT_ID, CID, CAPITAL_PRESERVATION, RISK_LEVEL, CREATION_DT, RM_ID)
- **INFO_DOCS** (DOC_ID, DSCR, FILE_PATH, EMPL_ID, CREATION_DT)
- **CUST_ORDER** (ORDER_ID, CID, DSCR, CREATION_DT, RM_ID)

PB_CUSTOMER: Πίνακας Πελατών Private Banking

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
CID	NUMBER(15)	ΝΑΙ	ΝΑΙ	Κωδικός Πελάτη
PBC_NAME	VARCHAR2(50)	ΟΧΙ	ΟΧΙ	Όνομα Πελάτη
PBC_SNAME	VARCHAR2(50)	ΟΧΙ	ΝΑΙ	Επώνυμο Πελάτη / Επωνυμία εταιρίας
PBC_FNAME	VARCHAR2(50)	ΟΧΙ	ΟΧΙ	Πατρώνυμο Πελάτη
PBC_BDATE	DATE	ΟΧΙ	ΟΧΙ	Ημ/νία Γέννησης
PBC_TAXCODE	CHAR(15)	ΟΧΙ	ΝΑΙ	Α. Φ. Μ.
PBC_ADT	VARCHAR2(50)	ΟΧΙ	ΟΧΙ	Α. Δ. Τ.
PBC_ADDRESS	VARCHAR2(250)	ΟΧΙ	ΝΑΙ	Διεύθυνση
PBC_TEL	NUMBER(15)	ΟΧΙ	ΝΑΙ	Τηλέφωνο Επικοινωνίας
CREATION_DT	CHAR(10)	ΟΧΙ	ΝΑΙ	Ημ/νία Δημιουργίας
FLG_EUROPHONE	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Υπηρεσίας Europhone 0 = ΟΧΙ, 1 = ΝΑΙ
FLG_CASH_CARD	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Υπηρεσίας Cash Card 0 = ΟΧΙ, 1 = ΝΑΙ
CREDIT_RATE	NUMBER	ΟΧΙ	ΝΑΙ	Ποσό Πίστωσης
FLG_E_BANKING	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Υπηρεσίας e Banking 0 = ΟΧΙ, 1 = ΝΑΙ
RISK_PROFILE	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Προφίλ Επενδυτικού Κινδύνου 0 = Χαμηλό, 1 = Μεσαίο 2 = Υψηλό
PBC_FLG_KIND	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Κατηγορίας Πελάτη 0 = Φυσικό πρόσωπο, 1 = Νομικό πρόσωπο

EMPLOYEE: Πίνακας Υπαλλήλων Τράπεζας

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
EMPL_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Υπαλλήλου
EMPL_NAME	VARCHAR2(50)	ΟΧΙ	ΝΑΙ	Όνομα Υπαλλήλου
EMPL_SNAME	VARCHAR2(50)	ΟΧΙ	ΝΑΙ	Επώνυμο Υπαλλήλου
EMPL_FNAME	VARCHAR2(50)	ΟΧΙ	ΝΑΙ	Πατρώνυμο Υπαλλήλου
EMPL_ADT	CHAR(10)	ΟΧΙ	ΝΑΙ	Α.Δ.Τ. Υπαλλήλου
EMPL_ADDRESS	VARCHAR2(50)	ΟΧΙ	ΟΧΙ	Δ/ση Υπαλλήλου
EMPL_TEL	CHAR(10)	ΟΧΙ	ΟΧΙ	Τηλέφωνο Υπαλλήλου
EMPL_EMAIL	VARCHAR2(50)	ΟΧΙ	ΟΧΙ	e-mail Υπαλλήλου
MON_SALARY	NUMEMBER(5,2)	ΟΧΙ	ΟΧΙ	Μηνιαίος Μισθός
UNIT_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός Μονάδας
ROLE_ID	CHAR(10)	ΟΧΙ	ΟΧΙ	Κωδικός Ρόλου

CUST_ORDER: Πίνακας Εντολών Πελατών

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
ORDER_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Εντολής
CID	NUMBER(15)	ΟΧΙ	ΝΑΙ	Κωδικός Πελάτη
DSCR	VARCHAR2(250)	ΟΧΙ	ΝΑΙ	Περιγραφή Εντολής
CREATION_DT	DATE	ΟΧΙ	ΝΑΙ	Ημ/νία Καταχώρησης
RM_ID	CHAR(10)	ΟΧΙ	ΝΑΙ	Κωδικός RM

INFO_DOCS: Πίνακας Ενημερωτικών Εκθέσεων και Αναφορών

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
DOC_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Αρχείου
DSCR	VARCHAR2(250)	ΟΧΙ	ΝΑΙ	Περιγραφή Αρχείου
FILE_PATH	VARCHAR2(200)	ΟΧΙ	ΝΑΙ	Φυσική Διεύθυνση αποθήκευσης στον server
EMPL_ID	CHAR(10)	ΟΧΙ	ΝΑΙ	Κωδικός Υπαλλήλου
CREATION_DT	DATE	ΟΧΙ	ΝΑΙ	Ημ/νία Καταχώρησης

PORTFOLIO: Πίνακας Χαρτοφυλακίων

ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΚ	Υ	ΠΕΡΙΓΡΑΦΗ
PRT_ID	CHAR(10)	ΝΑΙ	ΝΑΙ	Κωδικός Χαρτοφυλακίου
CID	NUMBER(15)	ΟΧΙ	ΝΑΙ	Κωδικός πελάτη
CAPITAL	NUMBER	ΟΧΙ	ΝΑΙ	Κεφάλαιο
RISK_LEVEL	CHAR(1)	ΟΧΙ	ΝΑΙ	FLAG Επενδυτικού Κινδύνου 0 = Χαμηλό, 1 = Μεσαίο 2 = Υψηλό
CREATION_DT	DATE	ΟΧΙ	ΝΑΙ	Ημ/νία Δημιουργίας
RM_ID	CHAR(10)	ΟΧΙ	ΝΑΙ	Κωδικός RM

4.1. Εισαγωγή

Στη παρούσα ενότητα ασχολούμεθα με τη χρήση των βιβλιοθηκών συλλογών, δηλαδή:

- τη σύνδεση με τη βάση δεδομένων Oracle
- τον Container
- τον πίνακα Table που δημιουργείται με τη χρήση του Oracle

4.2. Σύνδεση με τη βάση δεδομένων Oracle

Στην παρούσα ενότητα ασχολούμεθα με τη σύνδεση με τη βάση δεδομένων Oracle, επιλογώντας με τον JDBC Driver τον Oracle Driver, τον κώδικα που είναι των αποτελεσμάτων, δηλαδή τον κώδικα που θα εκτελεστεί στο Oracle, δηλαδή τον κώδικα που θα εκτελεστεί στο Oracle.

ΚΕΦΑΛΑΙΟ 4° ΑΝΑΛΥΣΗ JAVA ΚΩΔΙΚΑ

```

import java.sql.*;

class Main {
    public static void main(String[] args) {
        try {
            // Create a connection to the Oracle database
            Connection conn = DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:orcl",
                "scott", "tiger");

            // Create a statement to execute the SQL query
            Statement stmt = conn.createStatement();

            // Execute the SQL query
            ResultSet rs = stmt.executeQuery("SELECT * FROM emp");

            // Print the results
            while (rs.next()) {
                System.out.println(rs.getString(1) + " " +
                    rs.getString(2) + " " +
                    rs.getString(3) + " " +
                    rs.getString(4) + " " +
                    rs.getString(5));
            }

            // Close the connection
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

4.1. Εισαγωγή

Στην παρούσα ενότητα αναλύονται τα σημαντικότερα μέρη των προγραμμάτων, δηλαδή

- της σύνδεσης με τη Βάση Δεδομένων Oracle
- των Containers
- του πίνακα JTable που διαχειρίζεται τα αποτελέσματα των queries

4.2. Σύνδεση με τη Βάση Δεδομένων Oracle 9i

Στην παρούσα εφαρμογή χρησιμοποιείται η Βάση Δεδομένων Oracle 9i, η σύνδεση με επιτυγχάνεται με τον JDBC Driver της Java, αφού πρώτα έχει γίνει κλήση των αντίστοιχων βιβλιοθηκών `java.sql.*`; και `oracle.jdbc.driver.OracleDriver`; Παρακάτω παρουσιάζονται οι αντίστοιχες συναρτήσεις σύνδεσης της Βάσης Δεδομένων. Εκτέλεση ενός ερωτήματος και παρουσίαση των αποτελεσμάτων.

```

/** Σύνδεση με τη Βάση Δεδομένων PBank της Oracle */
try {
    Driver MyDriver = (Driver)Class.forName("oracle.jdbc.driver.OracleDriver").newInstance();
    conn = DriverManager.getConnection("jdbc:oracle:thin:@PCAKI:1521:ORCL", "PBANK", "PBANK");
}
catch (ClassNotFoundException ex) {
    System.err.println("Cannot find DataBase Driver Classes");
    System.err.println(ex);
}
catch (SQLException ex) {
    System.err.println("Cannot connect to the Oracle DataBase.");
    System.err.println(ex);
}
Statement stmt = conn.createStatement();
ResultSet rset = stmt.executeQuery("SELECT CID FROM PB_CUSTOMER");
while (rset.next()){
    System.out.println(rset.String(1));
}
rset.close();
stmt.close();
conn.close();

```

Αναλυτικά τίθεται ο JDBC Driver που θα χρησιμοποιηθεί στη συνέχεια γίνεται η σύνδεση με τη βάση. Ακολουθεί η δήλωση του Statement, και με την εκτέλεση της μεθόδου executeQuery("SELECT CID FROM PB_CUSTOMER") εκτελείται το ερώτημα που βρίσκεται μέσα στις παρενθέσεις, ενώ το αποτέλεσμα της αναζήτησης αποθηκεύεται στο ResultSet rset. Η μέθοδος next() χρησιμοποιείται για τη μετακίνηση μεταξύ των εγγραφών που βρίσκονται αποθηκευμένες στο rset. Η σύνδεση με τη βάση δεδομένων γίνεται με τη μέθοδο close().

```

import java.sql.*;

public class jdbc {
    static Connection con;
    static Statement stmt;
    static ResultSet rset;
    static String url = "jdbc:oracle:oci8://localhost:1521/orcl";
    static String user = "scott";
    static String password = "tiger";

    public static void main(String args[]) {
        try {
            con = DriverManager.getConnection(url, user, password);
            stmt = con.createStatement();
            rset = stmt.executeQuery("SELECT CID FROM PB_CUSTOMER");
            while (rset.next()) {
                System.out.println(rset.getString(1));
            }
            rset.close();
            stmt.close();
            con.close();
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }
}

```

Σε περίπτωση που ο κωδικός του προγράμματος είναι σωστός, αλλά δεν υπάρχει αποτέλεσμα, τότε ο κωδικός του προγράμματος είναι σωστός, αλλά ο κωδικός του προγράμματος είναι λάθος. Το βιβλίο αποτελείται από 10 κεφάλαια, τα οποία είναι: 1. Εισαγωγή, 2. JDBC, 3. JDBC Driver, 4. JDBC Driver, 5. JDBC Driver, 6. JDBC Driver, 7. JDBC Driver, 8. JDBC Driver, 9. JDBC Driver, 10. JDBC Driver.

4.3. Δημιουργία πίνακα (JTable)

Στη Java για να μπορεί να γίνει η επεξεργασία και εμφάνιση των αποτελεσμάτων ενός ερωτήματος SQL θα πρέπει τα αποτελέσματα ενός ResultSet να εμφανίζονται σε έναν πίνακα JTable. Για να εμφανιστούν τα δεδομένα πρέπει πρώτα να δημιουργηθεί το αντικείμενο oracleTable, το οποίο είναι ο συνδετικός κρίκος για την σύνδεση με τον πίνακα. Όλα αυτά επιτυγχάνονται αφού πρώτα γίνει η κλήση της βιβλιοθήκης `javax.swing.JTable`;

```
oracleTable data = new oracleTable();
JTable jTable1 = new JTable(data);
jTable1.getTableHeader().setBackground(new Color(235, 235, 235));
jTable1.getTableHeader().setReorderingAllowed(false);
jTable1.getTableHeader().setResizingAllowed(false);
TableColumn aColumn = jTable1.getColumnModel().getColumn(1);
aColumn.setCellEditor(new DefaultCellEditor(jComboBox1));
TableColumn bColumn = jTable1.getColumnModel().getColumn(2);
bColumn.setCellEditor(new DefaultCellEditor(jComboBox3));
TableColumn cColumn = jTable1.getColumnModel().getColumn(2);
jTable1.setRowHeight(26);
jTable1.setFont(new Font("Arial", 0, 14));
jTable1.setGridColor(new Color(204, 204, 204));
jTable1.setColumnSelectionAllowed(true);
jTable1.setCellSelectionEnabled(false);
jTable1.setRowSelectionAllowed(false);
data.setValueAt("", jTable1.getEditingRow(), 2);
data.setValueAt("", jTable1.getEditingRow(), 0);
```

Σε πρώτο στάδιο δημιουργείται το αντικείμενο της κλάσης JTable και στη συνέχεια ορίζονται οι επικεφαλίδες των πεδίων καθώς και τα αριθμητικά πλαίσια του πίνακα. Τα δεδομένα αποθηκεύονται με την βοήθεια του αντικείμενου της κλάσης TableColumn, όπου δημιουργούνται οι ανάλογες στήλες με φορτωμένα τα δεδομένα. Για να αποθηκευτεί ένα δεδομένο σε κάποιο πεδίο, χρησιμοποιούμε την συνάρτηση `setValueAt()`.

4.4. Containers

Για να γίνει μεταφορά δεδομένων από μία δραστηριότητα σε μία άλλη χρησιμοποιούνται Containers, τα οποία είναι προγράμματα ανεξάρτητα μεταξύ τους. Για να λειτουργήσουν σωστά θα πρέπει κληθούν οι βιβλιοθήκες της `ibm com.ibm.workflow.api.*`; και `com.ibm.websphere.*`;

```

/***** MQ Workflow *****/
try {
  Agent agent = new Agent();
  agent.setLocator(Agent.LOC_LOCATOR);
  agent.setName("LOCAL");
  PEA=agent.getExecutionAgent();
}
catch (FmcException fmc){
  System.out.println("Error: "+fmc);
}
catch (PropertyVetoException pve) {}
/*****

```

Αρχικά γίνεται η ρύθμιση του Agent, θέτοντας σαν Locator το LOC_LOCATOR και σαν name το LOCAL. Στην συνέχεια με την συνάρτηση getExecutionAgent() γίνεται η εκκίνηση του.

```

/***** MQ Workflow *****/
ExecutionAgent PEA;
String userid = "";
ReadOnlyContainer input;
ReadWriteContainer PR_CustomersDs;
String CID = "";
String PR_CUSTOMER_NAME = "";
String PR_CUST_SNAME = "";
String PR_CUST_FNAME = "";
String PR_CUST_ADT = "";
String PR_CUST_TAXCODE = "";
String PR_FLG_KIND = "";
String PR_FLG_STATUS = "";
/*****

```

οι μεταβλητές αποτελούν για το μοντέλο μέλη δομής δεδομένων (Data Structure).

```

/***** MQ Workflow *****/
PEA = a;
try {
  // Αποθήκευση του UserID του χρήστη
  userid = PEA.userID();
  // Ορισμός του input container
  input = PEA.inContainer();
  // Ορισμός του output container
  PR_CustomersDs = PEA.outContainer();
  // Καθορισμός της predefined μεταβλητής "_RC" του output container
  PR_CustomersDs.setLong("_RC", 0);
}

```

```
}  
catch (FmcException fmc) {  
    System.out.println("Error: "+fmc);  
}  
/*****/
```

Η συνάρτηση `inContainer()` δηλώνει τη δομή που θα δίνει τα δεδομένα, ενώ με την `outContainer()` ορίζεται η δομή στην θα αποθηκεύονται τα νέα δεδομένα. Στην προκειμένη περίπτωση η δομή είναι η `PR_CustomersDs`.

5.1. Εισαγωγή

Στην παρούσα εισαγωγή γίνεται μια εισαγωγή στις βασικές αρχές της μεθόδου των ελαστικών στοιχείων (FEA) και παρουσιάζονται οι βασικές αρχές της μεθόδου των ελαστικών στοιχείων (FEA).

5.2. Ανάλυση

Το πρώτο βήμα της μεθόδου των ελαστικών στοιχείων (FEA) είναι η διαίρεση του αντικείμενου σε μικρά στοιχεία. Η διαίρεση αυτή γίνεται με βάση την γεωμετρία του αντικείμενου και την αναμενόμενη συμπεριφορά του. Η διαίρεση αυτή γίνεται με βάση την γεωμετρία του αντικείμενου και την αναμενόμενη συμπεριφορά του. Η διαίρεση αυτή γίνεται με βάση την γεωμετρία του αντικείμενου και την αναμενόμενη συμπεριφορά του.

ΚΕΦΑΛΑΙΟ 5^ο ΣΥΜΠΕΡΑΣΜΑΤΑ

Το πρώτο βήμα της μεθόδου των ελαστικών στοιχείων (FEA) είναι η διαίρεση του αντικείμενου σε μικρά στοιχεία. Η διαίρεση αυτή γίνεται με βάση την γεωμετρία του αντικείμενου και την αναμενόμενη συμπεριφορά του. Η διαίρεση αυτή γίνεται με βάση την γεωμετρία του αντικείμενου και την αναμενόμενη συμπεριφορά του. Η διαίρεση αυτή γίνεται με βάση την γεωμετρία του αντικείμενου και την αναμενόμενη συμπεριφορά του.

Η μεθόδος των ελαστικών στοιχείων (FEA) είναι μια μέθοδος που χρησιμοποιείται για την ανάλυση της συμπεριφοράς των δομών υπό φορτίση. Η μεθόδος αυτή βασίζεται στην διαίρεση της δομής σε μικρά στοιχεία, τα οποία αναλύονται ξεχωριστά. Η μεθόδος αυτή βασίζεται στην διαίρεση της δομής σε μικρά στοιχεία, τα οποία αναλύονται ξεχωριστά. Η μεθόδος αυτή βασίζεται στην διαίρεση της δομής σε μικρά στοιχεία, τα οποία αναλύονται ξεχωριστά.

Αν και η μεθόδος των ελαστικών στοιχείων (FEA) είναι μια μέθοδος που χρησιμοποιείται για την ανάλυση της συμπεριφοράς των δομών υπό φορτίση, η μεθόδος αυτή βασίζεται στην διαίρεση της δομής σε μικρά στοιχεία, τα οποία αναλύονται ξεχωριστά. Η μεθόδος αυτή βασίζεται στην διαίρεση της δομής σε μικρά στοιχεία, τα οποία αναλύονται ξεχωριστά. Η μεθόδος αυτή βασίζεται στην διαίρεση της δομής σε μικρά στοιχεία, τα οποία αναλύονται ξεχωριστά.

5.1. Εισαγωγή

Στην παρούσα ενότητα γίνεται μία ανασκόπηση όλης της εργασίας και κατόπιν παρουσιάζονται συμπεράσματα και προτάσεις της μεθόδου που χρησιμοποιήθηκε.

5.2. Ανασκόπηση

Σύμφωνα με την αρχιτεκτονική SOA κάθε επιχειρησιακή διαδικασία (business process) αντιστοιχεί σε μία υπηρεσία (service), η οποία αποτελείται από επιμέρους δραστηριότητες (Activities) και επικοινωνεί με τις υπόλοιπες επιχειρησιακές διαδικασίες με προτυποποιημένο τρόπο. Έτσι κάθε τέτοια επιχειρησιακή διαδικασία είναι αυτόνομη επαναχρησιμοποιήσιμη και ανεξάρτητη από τις εφαρμογές και τις υπολογιστικές πλατφόρμες στις οποίες λειτουργεί.

Στα παραπάνω κεφάλαια περιγράφηκαν οι τρεις επιχειρησιακές διαδικασίες (προσέγκυσης πελατείας, αποδοχής πελάτη και ενημέρωσης και εξυπηρέτησης πελατείας) της διεύθυνσης Private Banking μιας Τράπεζας. Αναλύθηκαν οι επιμέρους δραστηριότητες που απαρτίζουν την κάθε business process, καθώς και οι ρόλοι που μπορούν να μπορούν να την εκτελούν - διαχειρίζονται. Για την υλοποίηση των επιχειρησιακών διαδικασιών χρησιμοποιήθηκαν τα εργαλεία WebSphere MQ Workflow Buildtime και client της IBM καθώς και η βάση δεδομένων DB2, ενώ η αποθήκευση των δεδομένων γίνεται στη βάση δεδομένων Oracle 9i.

Με την αρχιτεκτονική Service Oriented Architecture εκτός του ότι δίνεται η δυνατότητα σε κάποιον να αντιληφθεί μια επιχείρηση (στο συγκεκριμένο παράδειγμα μια Τράπεζα) ως ένα σύνολο από συνδεδεμένες υπηρεσίες που αντιστοιχούν σε επιχειρησιακές διαδικασίες, προσφέρει επίσης την δυνατότητα της επαναχρησιμοποίησης αυτών των αυτόνομων μεταξύ τους διαδικασιών, γεγονός που μπορεί να μειώσει τόσο το κόστος όσο και το χρόνο υλοποίησης νέων εφαρμογών.

Αν και αρχικά, η διάσπαση ενός συστήματος σε επιμέρους τμήματα (business process) και η υλοποίηση κάθε business process ως αυτόνομη οντότητα φαίνεται εύκολη υπόθεση. Ωστόσο, η συνήθεια της γραμμικής προσέγγισης μιας εφαρμογής δυσκολεύει αρκετά την υλοποίηση της, τουλάχιστον σε αρχικό στάδιο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. H.M. Deitel, P.J. Deitel, "JAVA How To Program", 5th Edition, Prentice Hall ISBN:0-13-120236-7
2. W. Dieter, A. Winfried, B. Dallas "MQSeries Workflow for Windows NT for Beginners", ibm.com/redbooks
3. H.M. Deitel, P.J. Deitel, B. Duwaldt, L.K. Trees, "Web Services A Technical Introduction", Prentice Hall ISBN:0-13-046135-0.
4. Jeffrey Hoffer, Joey George, Joseph Valacich, "Modern System Analysis and Design", 4th Edition, Prentice Hall ISBN: 0-13-145461-7.
5. OPEN newsletter "Αφιέρωμα IBM Service Oriented Architecture" 3ο/2006, έκδοση ΤΕΧΝΟΠΟΛΙΣ Α.Ε.

Πανεπιστήμιο Πειραιώς

ΠΑΡΑΡΤΗΜΑ Α

Περιγραφή Λογισμικού

Κώδικας Java: **CheckForNewCustomers.java**

```

import java.awt.*;
import java.awt.event.*;
import java.awt.event.WindowListener;
import java.awt.event.WindowEvent;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.Event;
import java.awt.Dimension;
import java.awt.Rectangle;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ItemListener;
import java.awt.event.ItemEvent;
import javax.swing.*;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JMenuBar;
import javax.swing.JMenu;
import javax.swing.JMenuItem;
import javax.swing.BorderFactory;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JButton;
import javax.swing.JRootPane;
import javax.swing.JRadioButton;
import javax.swing.border.BevelBorder;
import javax.swing.border.EtchedBorder;
import javax.swing.KeyStroke;
import javax.swing.JCheckBox;
import javax.swing.table.*;
import javax.swing.DefaultCellEditor;
import javax.swing.Timer;

import java.sql.*;
import oracle.jdbc.driver.OracleDriver;
import oracle.jdeveloper.layout.XYConstraints;
import oracle.jdeveloper.layout.XYLayout;

/***** MQ Workflow *****/
import com.ibm.workflow.api.*;
import com.*;
import com.ibm.websphere.*;
/*****

public class CheckForNewCustomers extends JFrame implements ActionListener {

```

```

/***** MQ Workflow *****/
ExecutionAgent PEA;
String userid = "";
int dept = 0;
ReadOnlyContainer input;
ReadWriteContainer Check_For_New_CustomersDs;
/*****

JPanel jPanel1 = new JPanel();
XYLayout xYLayout1 = new XYLayout();
JLabel jLabel1 = new JLabel();
JMenuBar menuBar1 = new JMenuBar();
JMenu jMenuItem1 = new JMenu();
JMenu jMenuItem2 = new JMenu();
JMenuItem jMenuItem1 = new JMenuItem();
JMenuItem jMenuItem2 = new JMenuItem();
JLabel jLabel2 = new JLabel();
JLabel jLabel3 = new JLabel();
JPanel jPanel2 = new JPanel();
XYLayout xYLayout2 = new XYLayout();
JScrollPane jScrollPane1 = new JScrollPane();

private String[] columns = {"Κωδικός Πελάτη", "Επώνυμο Πελάτη", "Όνομα Πελάτη", "Πατρώνυμο
Πελάτη", "Α.Φ.Μ.", "Επιλογή Πελάτη"};
oracleTable data = new oracleTable();
JTable jTable1 = new JTable();

JButton jButton2 = new JButton();
JLabel jLabel4 = new JLabel();
JLabel jLabel5 = new JLabel();
JLabel jLabel6 = new JLabel();
JLabel jLabel7 = new JLabel();
Timer timer;
Connection conn;
private JRadioButton jButton1 = new JRadioButton();
private JRadioButton jButton2 = new JRadioButton();
private ButtonGroup radioGroup;
JCheckBox jButton1 = new JCheckBox();
JButton jButton3 = new JButton();

public CheckForNewCustomers(ExecutionAgent a) { // Constructor

/***** MQ Workflow *****/
PEA = a;
try {
// Αποθήκευση του UserID του χρήστη
userid = PEA.userID();
// Ορισμός του input container
input = PEA.inContainer();
// Ορισμός του output container
Check_For_New_CustomersDs = PEA.outContainer();
// Καθορισμός της predefined μεταβλητής "_RC" του output container
Check_For_New_CustomersDs.setLong("_RC", 0);
}
catch (FmcException fmc) {
System.out.println("Error: "+fmc);
}
/*****

```

```

try{
    jbInit();
}
catch(Exception e){
    e.printStackTrace();
}

} // end constructor

private void jbInit() throws Exception {

    /** Σύνδεση με τη Βάση Δεδομένων PBank της Oracle */
    try {
        Driver MyDriver = (Driver)Class.forName("oracle.jdbc.driver.OracleDriver").newInstance();
        conn = DriverManager.getConnection("jdbc:oracle:thin:@PcAKI:1521:ORCL", "PBANK", "PBANK");
    }
    catch (ClassNotFoundException cnfe) {}

    int delay = 1000;
    timer = new Timer(delay, this);
    timer.setInitialDelay(0);

    // Στοιχεία του Header του πίνακα (JTable1)
    JTable1.getTableHeader().setBackground(new Color(235, 235, 235));
    JTable1.getTableHeader().setReorderingAllowed(false);
    JTable1.getTableHeader().setResizingAllowed(false);
    //TableColumn bColumn = JTable1.getColumnModel().getColumn(6);
    //bColumn.setCellEditor(new DefaultCellEditor(jCheckBox1));
    JTable1.setPreferredSize(new Dimension(50, 180));
    JTable1.setFont(new Font("Arial", 0, 14));
    JTable1.setGridColor(new Color(204, 204, 204));
    JTable1.setRowHeight(30);
    JTable1.setToolTipText("Λίστα διαθέσιμων υποψήφιων πελατών");

    ImageIcon logo = new ImageIcon("D:/DIPLOMATIKI/PROGRAMS_EXE/PROCESS_01/images/Bank-128.png");

    jRadioButton1.setText("Φυσικά Πρόσωπα");
    jRadioButton1.setFont(new Font("Dialog", 1, 12));
    jRadioButton1.setForeground(new Color(34, 63, 114));
    jRadioButton1.setBackground(new Color(181, 181, 235));
    jRadioButton2.setText("Νομικά Πρόσωπα");
    jRadioButton2.setFont(new Font("Dialog", 1, 12));
    jRadioButton2.setMaximumSize(new Dimension(128, 26));
    jRadioButton2.setMinimumSize(new Dimension(128, 26));
    jRadioButton2.setPreferredSize(new Dimension(128, 26));
    jRadioButton2.setForeground(new Color(34, 63, 114));
    jRadioButton2.setBackground(new Color(181, 181, 235));
    jRadioButton1.setSelected(true);
    jRadioButton2.setSelected(false);
    radioGroup = new ButtonGroup();
    jButton3.setText("Διόρθωση");
    jButton3.setBorder(BorderFactory.createEtchedBorder(EtchedBorder.LOWERED));
    jButton3.setFont(new Font("Arial", 1, 16));
    jButton3.setBackground(new Color(235, 235, 235));
    jButton3.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {

```



```

        jButton2_actionPerformed(e);
    }
});

jButton2.setMaximumSize(new Dimension(85, 25));
jButton2.setMinimumSize(new Dimension(85, 25));
jButton2.setActionCommand("Επιλογή");
radioGroup.add(jRadioButton1);
radioGroup.add(jRadioButton2);

jRadioButton1.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        jRadioButton1_actionPerformed(e);
    }
});

jRadioButton2.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        jRadioButton2_actionPerformed(e);
    }
});

this.getContentPane().setLayout(null);
this.setSize(new Dimension(850, 700));
this.setTitle("Οθόνη 1: Έλεγχος Διαθέσιμων Υποψήφιων Πελατών");
this.setJMenuBar(menuBar1);
this.setResizable(false);

this.addWindowListener(new java.awt.event.WindowAdapter(){
    public void windowOpened(WindowEvent e){
        this_windowOpened(e);
    }
});
jPanel1.setBounds(new Rectangle(0, 0, 843, 654));
jPanel1.setBackground(new Color(181, 181, 235));
jPanel1.setLayout(xYLayout1);
jPanel1.setPreferredSize(new Dimension(843, 654));
jPanel1.setMinimumSize(new Dimension(843, 654));
jLabel1.setIcon(logó);
menuBar1.setBackground(Color.white);
menuBar1.setFont(new Font("Dialog", 0, 12));
jMenu1.setText("Επιλογές");
jMenu1.setBackground(Color.white);
jMenu1.setFont(new Font("Dialog", 0, 12));
jMenu2.setText("Βοήθεια");
jMenu2.setBackground(Color.white);
jMenu2.setFont(new Font("Dialog", 0, 12));
jMenuItem1.setText("Σχετικά...");
jMenuItem1.setBackground(Color.white);
jMenuItem1.setFont(new Font("Dialog", 0, 12));
jMenuItem1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_S, Event.ALT_MASK, false));

jMenuItem1.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        jMenuItem1_actionPerformed(e);
    }
});

```

```

jMenuItem2.setText("Κλείσιμο");
jMenuItem2.setBackground(Color.white);
jMenuItem2.setFont(new Font("Dialog", 0, 12));
jMenuItem2.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_E, Event.ALT_MASK, false));

jMenuItem2.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        jMenuItem2_actionPerformed(e);
    }
});

jLabel2.setText("ΤΜΗΜΑ: Προσέλευσης Πελατείας");
jLabel2.setFont(new Font("Arial", 1, 14));
jLabel2.setForeground(Color.black);
jLabel3.setFont(new Font("Arial", 1, 18));
jLabel3.setForeground(Color.black);
jPanel2.setBackground(new Color(181, 181, 235));
jPanel2.setBorder(BorderFactory.createTitledBorder("Διαθέσιμοι Υποψήφιοι Πελάτες"));
jPanel2.setLayout(xYLayout2);
jScrollPane1.setBorder(BorderFactory.createTitledBorder(""));
jScrollPane1.setBackground(new Color(199, 210, 235));
jButton2.setText("Επιλογή");
jButton2.setBorder(BorderFactory.createEtchedBorder(EtchedBorder.LOWERED));
jButton2.setFont(new Font("Arial", 1, 16));
jButton2.setBackground(new Color(235, 235, 235));
jLabel4.setText("Ημερομηνία:");
jLabel4.setFont(new Font("Arial", 0, 14));
jLabel4.setForeground(Color.black);
jLabel5.setText("Ωρα:");
jLabel5.setFont(new Font("Arial", 0, 14));
jLabel5.setForeground(Color.black);
jLabel6.setForeground(new Color(34, 63, 114));
jLabel6.setFont(new Font("Arial", 1, 12));
jLabel7.setForeground(new Color(34, 63, 114));
jLabel7.setFont(new Font("Arial", 1, 12));
jPanel1.add(jButton3, new XYConstraints(435, 535, 115, 50));
jPanel1.add(jLabel7, new XYConstraints(205, 5, 65, 15));
jPanel1.add(jLabel6, new XYConstraints(95, 5, 75, 15));
jPanel1.add(jLabel5, new XYConstraints(170, 5, 35, 15));
jPanel1.add(jLabel4, new XYConstraints(10, 5, 80, 15));
jPanel1.add(jButton2, new XYConstraints(305, 535, 115, 50));
jScrollPane1.getViewport().add(jTable1, null);
jPanel2.add(jScrollPane1, new XYConstraints(10, 20, 785, 185));
jPanel1.add(jPanel2, new XYConstraints(20, 150, 815, 245));
jPanel1.add(jLabel3, new XYConstraints(165, 75, 180, 15));
jPanel1.add(jLabel2, new XYConstraints(445, 45, 230, 20));
jPanel1.add(jLabel1, new XYConstraints(700, 0, 130, 140));
jPanel1.add(jRadioButton2, new XYConstraints(430, 425, 130, 25));
jPanel1.add(jRadioButton1, new XYConstraints(295, 425, 130, 25));
this.getContentPane().add(jPanel1, null);
jMenu1.add(jMenuItem2);
menuBar1.add(jMenu1);
jMenu2.add(jMenuItem1);
menuBar1.add(jMenu2);

jButton2.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        jButton2_actionPerformed(e);
    }
});

```

```

});

} // end jbInit()

void jMenuItem1_actionPerformed(ActionEvent e){
    center c;
    JFrame j = new about(this);
    c = new center(this,j);
    j.setLocation(c.x, c.y);
    this.setEnabled(false);
    j.show();
} // end method JMenuItem_actionPerformed()

void this_windowOpened(WindowEvent e){
    this.startAnimation();
    // Ανάκτηση δεδομένων για τον πίνακα της φόρμας
    data.executeQuery("select CID, PRC_SNAME, PRC_NAME, PRC_FNAME, PRC_TAXCODE,
PRC_FLG_STATUS FROM PROSPECT_CUSTOMER WHERE PRC_FLG_KIND = 0 AND PRC_FLG_STATUS =0",
columns);
    jTable1.setModel(data);
}

void jMenuItem2_actionPerformed(ActionEvent e){
    // Καθορισμός της predefined μεταβλητής "_RC" του output container
    try {
        Check_For_New_CustomersDs.setLong("_RC", 1); // Το πρόγραμμα κλείνει αυτόματα
        PEA.setOutContainer(Check_For_New_CustomersDs);
        System.exit(0);
    }
    catch (FmcException fmc){}
}

void jButton2_actionPerformed(ActionEvent e){
    try {
        Check_For_New_CustomersDs.setLong("_RC", 1); // Το πρόγραμμα κλείνει αυτόματα
        PEA.setOutContainer(Check_For_New_CustomersDs);
        System.exit(0);
    }
    catch (FmcException fmc){}
}

public void actionPerformed(ActionEvent e){
    // Καθορισμός της τρέχουσας ημερομηνίας και ώρας
    jLabel6.setText(new getDate().day+"/"+new getDate().month+"/"+new getDate().year);
    jLabel7.setText(new getDate().hour+": "+new getDate().min+": "+new getDate().sec);
}

// Εκκίνηση του animation για την ημερομηνία και την ώρα
public void startAnimation(){
    if (!timer.isRunning()) timer.start();
}

void jButton1_actionPerformed(ActionEvent e){
    // Αλλάζω το Query για να πάρω μόνο τα φυσικά πρόσωπα
    String[] columns = {"Κωδικός Πελάτη", "Επίνομο Πελάτη", "Όνομα Πελάτη", "Πατρώνυμο Πελάτη",
"A.Φ.Μ.", "Επιλογή Πελάτη"};
    data.executeQuery("select CID, PRC_SNAME, PRC_NAME, PRC_FNAME, PRC_TAXCODE,
PRC_FLG_STATUS FROM PROSPECT_CUSTOMER WHERE PRC_FLG_KIND = 0 AND PRC_FLG_STATUS =0",
columns);
}

```



```

jTable1.setModel(data);
}

void jButton2_actionPerformed(ActionEvent e){
// Αλλάζω το Query για να πάρω μόνο τα νομικά πρόσωπα
String[] columns = {"Κωδικός Εταιρίας", "Επωνυμία Εταιρίας", "Α.Φ.Μ.", "Επιλογή Πελάτη" };
data.executeQuery("select CID, PRC_SNAME, PRC_TAXCODE, PRC_FLG_STATUS FROM
PROSPECT_CUSTOMER WHERE PRC_FLG_KIND = 1 AND PRC_FLG_STATUS =0", columns);
jTable1.setModel(data);
}
}
// end class

```

Κώδικας Java: oracleTable.java

```

/**
 * @Description Κλάση για την ανάκτηση τιμών από τη Βάση Δεδομένων PBank της Oracle
 * ενός query και εμφάνισή τους σε πίνακα jTable
 * Η αρίθμηση στον πίνακα jTable ξεκινά από το 0, ενώ σε ένα ResultSet από το 1.
 * Γι' αυτό είναι απαραίτητο να προσθέτουμε +1 σε κάθε γραμμή και στηλη
 */

import javax.swing.table.AbstractTableModel;
import javax.swing.event.TableModelEvent;
import java.util.Vector;
import java.sql.*;
import oracle.*;
import oracle.jdbc.driver.OracleDriver;
import gr.instore.ipis.util.*;
import java.io.*;

public class oracleTable extends AbstractTableModel {
    Connection conn;
    Statement stmt;
    ResultSet rset;
    String[] columnNames = {};
    Vector rows = new Vector();
    ResultSetMetaData metaData;
    int column;

    public oracleTable() { // Constructor
        // Σύνδεση με τη Βάση Δεδομένων της Oracle
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            DriverManager.registerDriver (new oracle.jdbc.driver.OracleDriver());
            conn = DriverManager.getConnection("jdbc:oracle:thin:@PCAKI:1521:ORCL", "PBANK", "PBANK");
        }
        catch (ClassNotFoundException ex) {
            System.err.println("Cannot find DataBase Driver Classes");
            System.err.println(ex);
        }
        catch (SQLException ex) {
            System.err.println("Cannot connect to the Oracle DataBase.");
            System.err.println(ex);
        }
    }
}

```

```

}
}

public void executeQuery(String query, String[] col) {
    if (conn == null) {
        System.err.println("There is no DataBase to execute the Query.");
        return;
    }
    try {
        stmt = conn.createStatement();
        rset = stmt.executeQuery(query); // Εκτελώ το query που πήρα σαν παράμετρο

        metaData = rset.getMetaData(); // εξασφαλίζω τα metadata για το ResultSet

        int numberOfColumns = metaData.getColumnCount();
        columnNames = new String[numberOfColumns];

        for(column = 0; column < numberOfColumns; column++) {
            columnNames[column] = col[column];
        }

        rows = new Vector();
        while (rset.next()) {
            Vector newRow = new Vector();
            for (int i = 1; i <= getColumnCount(); i++) {
                newRow.addElement(rset.getObject(i));
            }
            rows.addElement(newRow);
        }

        fireTableChanged(null);
    }
    catch (SQLException ex){
        System.err.println(ex);
    }
    Vector more = new Vector();

    while(this.getRowCount()<5){
        for (int i = 1; i <= getColumnCount(); i++) {
            more.addElement(null);
        }
        rows.addElement(more);
    }
} // end method executeQuery()

// Ανοσίγηση από τη Βάση Δεδομένων
public void close() throws SQLException {
    System.out.println("Close the DataBase connection");
    rset.close();
    stmt.close();
    conn.close();
}

protected void finalize() throws Throwable {
    close();
    super.finalize();
}

/** Επιστρέφει το όνομα μιας συγκεκριμένης στήλης στο ResultSet */
public String getColumnName(int column) {

```

```

if (columnNames[column] != null) {
    return columnNames[column];
} else {
    return "";
}
} //end method getColumnName()

/** Επιστρέφει την κλάση που αναπαριστά τον τύπο της στήλης */
public Class getColumnClass(int column) {

    int type;

    try {
        type = metaData.getColumnType(column+1);
    }
    catch (SQLException e) {
        return super.getColumnClass(column);
    }

    switch(type) {
        case Types.CHAR:
        case Types.VARCHAR:
        case Types.LONGVARCHAR:
            return String.class;
        case Types.BIT:
            return Boolean.class;
        case Types.TINYINT:
        case Types.SMALLINT:
        case Types.INTEGER:
            return Integer.class;
        case Types.BIGINT:
            return Long.class;
        case Types.FLOAT:
        case Types.DOUBLE:
            return Double.class;
        case Types.DATE:
            return java.sql.Date.class;
        default:
            return Object.class;
    } // end switch

} // end method getColumnClass()

/**Μετατρέπει τα κελία του πίνακα σε writable */
public boolean isCellEditable(int row, int column) {

    try {
        return metaData.isWritable(column+1);
    }
    catch (SQLException e) {
        return false;
    }
}

/** Επιστρέφει τον αριθμό των στηλών του ResultSet*/
public int getColumnCount() {
    return columnNames.length;
}

```



```

/** Επιστρέφει τον αριθμό των γραμμών του ResultSet */
public int getRowCount() {
    return rows.size();
}

/** Επιστρέφει την τιμή μιας συγκεκριμένης γραμμής και στηλης */
public Object getValueAt(int aRow, int aColumn) {

    Vector row = (Vector)rows.elementAt(aRow);
    return row.elementAt(aColumn);
}

public String dbRepresentation(int column, Object value) {
    int type;

    if (value == null) {
        return "null";
    }

    try {
        type = metaData.getColumnType(column+1);
    }
    catch (SQLException e) {
        return value.toString();
    }

    switch(type) {
        case Types.CHAR:
        case Types.VARCHAR:
        case Types.LONGVARCHAR:
            return value.toString();
        case Types.INTEGER:
        case Types.DOUBLE:
        case Types.FLOAT:
            return value.toString();
        case Types.BIT:
            return ((Boolean)value).booleanValue() ? "1" : "0";
        case Types.DATE:
            return value.toString();
        default:
            return "\"" + value.toString() + "\"";
    } // end switch
} // end method dbRepresentation ()

/** Περνώ τιμές για συγκεκριμένη γραμμή και στήλη */
public void setValueAt(Object value, int row, int column) throws IllegalStateException {
    try {
        String tableName = null;
        tableName = metaData.getTableName(1);
        metaData.getSchemaName(1);

        if (tableName == null) {
            System.out.println("There is no Table NAME !!! ");
        }
        System.out.println("row --> " + row);
        System.out.println("col --> " + column);
        System.out.println("tableName --> " + tableName);
        System.out.println("getColumnName --> " + metaData.getColumnName(column+1));
    }
}

```

```

String columnName = metaData.getColumnName(column+1);

String query ="update "+tableName+
              " set "+columnName+
              " = "+dbRepresentation(column, value)+
              " where ";

for( int col = 1; col < getColumnCount(); col++) {

    String colName = metaData.getColumnName(col+1);

    if (colName.equals("")) {
        continue;
    }
    if (col != 1) {
        query = query + " and ";
    }
    try{
        String var = SqlText.parse(dbRepresentation(col, getValueAt(row, col)));
        var = new String(var.getBytes(),"ISO-8859-7");

        query = query + colName + " = "+ var+"";
    }
    catch(UnsupportedEncodingException uee){
        uee.printStackTrace();
    }

} //end for loop

dbRepresentation(column, getValueAt(row, column));

System.out.println(query);
System.out.println("Not sending update to DataBase");

}
catch (SQLException e) {
    System.err.println("Update Failed !!!");
}
Vector dataRow = (Vector)rows.elementAt(row);
dataRow.setElementAt(value, column);

} // end method setValueAt()
}

```