



ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ ΞΑΝΘΗΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑΣ ΠΛΗΡΟΦΟΡΙΩΝ

ΔΙΑΧΕΙΡΙΣΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ΠΛΑΤΦΟΡΜΕΣ ΜΕ
ΠΕΡΙΟΡΙΣΜΕΝΟΥΣ ΥΠΟΛΟΓΙΣΤΙΚΟΥΣ ΠΟΡΟΥΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ του ΔΡΟΣΑΤΟΥ ΓΕΩΡΓΙΟΥ

ΕΠΙΒΛΕΠΩΝ: ΛΕΚΤΟΡΑΣ
κ. ΕΦΡΑΙΜΙΔΗΣ ΠΑΥΛΟΣ

ΞΑΝΘΗ 2006

Αφιερώνεται στον καθηγητή και
επιβλέποντα αυτής της διπλωματικής εργασίας,
στην οικογένειά μου,
και σε όσους με βοήθησαν σε
αυτή μου την προσπάθεια.

Περίληψη

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η διαχείριση δεδομένων σε πλατφόρμες με περιορισμένους υπολογιστικούς πόρους. Πιο συγκεκριμένα ο όρος «διαχείριση δεδομένων» αναφέρεται κυρίως στην ύπαρξη τόσο μίας μικρής βάσης δεδομένων στη φορητή πλατφόρμα όσο και στην ύπαρξη μίας απομακρυσμένης κεντρικής βάσης δεδομένων, όπου ανάμεσα στις δύο αυτές βάσεις θα υπάρχει κάποιο είδος επικοινωνίας ανάλογα με τις δυνατότητες που έχει η κάθε πλατφόρμα. Από την άλλη με ο όρος «πλατφόρμες με περιορισμένους υπολογιστικούς πόρους» αναφέρεται κυρίως σε φορητές υπολογιστικές μονάδες (συσκευές) όπως είναι τα PDA και πιο συγκεκριμένα το Pocket PC με λειτουργικό σύστημα Windows Mobile 5.0 (Microsoft).

Αυτό το οποίο πραγματοποιήθηκε ήταν η κατασκευή τεσσάρων πρότυπων εφαρμογών για να δοκιμαστούν οι διάφορες δυνατότητες που παρέχει ο SQL Server 2005 και τα Windows Mobile 5.0. Όσον αφορά την πρώτη εφαρμογή η οποία ονομάζεται «Anakoinwseis_DB» έχει τη δυνατότητα παρουσίασης, εισαγωγής, επεξεργασίας και διαγραφής δεδομένων από έναν πίνακα μίας τοπικής βάσης δεδομένων. Επίσης, έχει την δυνατότητα συγχρονισμού της τοπικής βάσης δεδομένων με την κεντρική βάση όποτε ο (Admin) χρήστης της εφαρμογής το κρίνει απαραίτητο, ανάλογα βέβαια με το γεγονός κατά πόσο εκείνη τη στιγμή έχει διαθέσιμο κάποιο δίκτυο για την παροχή Internet. Ένα τέτοιο δίκτυο μπορεί να είναι κάποιο τοπικό ασύρματο δίκτυο (Wireless LAN), κυψελοειδές δίκτυο κινητής τηλεφωνίας (GPRS), δορυφορικό δίκτυο επικοινωνιών καθώς ακόμη και κάποιο ενσύρματο δίκτυο. Λόγω αυτής της δυνατότητας συγχρονισμού μπορούν και κάποιοι άλλοι χρήστες που διαθέτουν την ίδια εφαρμογή και τα ίδια δικαιώματα στην κεντρική βάση να παρακολουθούν τις διάφορες αλλαγές και να πράττουν ανάλογα. Τέλος, θα πρέπει να σημειωθεί ότι η συγκεκριμένη εφαρμογή είναι για χρήστες οι οποίοι έχουν δικαιώματα διαχειριστή και γενικά μπορούν να πραγματοποιήσουν οποιαδήποτε ενέργεια πρώτα στην τοπική βάση δεδομένων και ύστερα μέσω του συγχρονισμού και στην κεντρική βάση δεδομένων.

Η δεύτερη εφαρμογή η οποία κατασκευάστηκε ονομάζεται «Anakoinwseis_RDB» και έχει μόνο τη δυνατότητα παρουσίασης δεδομένων από έναν πίνακα της τοπικής βάσης δεδομένων, χωρίς να μπορεί να εισαγάγει, επεξεργαστεί και διαγράψει δεδομένα από αυτό τον πίνακα. Πιο συγκεκριμένα στην εφαρμογή αυτή ο απλός χρήστης έχει την δυνατότητα συγχρονισμού με την κεντρική βάση δεδομένων, βλέποντας μόνο τις αλλαγές οι οποίες έχουν γίνει σε αυτή από τους χρήστες της πρώτης εφαρμογής, χωρίς όμως να έχει περαιτέρω δικαιώματα για οτιδήποτε άλλο. Ουσιαστικά δηλαδή αυτή η εφαρμογή είναι μόνο για ανάγνωση (Read Only).

Η τρίτη εφαρμογή η οποία ονομάζεται «Anakoinwseis_DB_ENCR» κατασκευάστηκε με σκοπό να δοκιμαστούν περισσότερο τόσο τα θέματα ασφαλείας στην εφαρμογή και στην τοπική βάση δεδομένων όσο και τα θέματα ασφαλείας στην κεντρική βάση δεδομένων. Στην εφαρμογή αυτή επίσης υπάρχει η δυνατότητα παρουσίασης, εισαγωγής, επεξεργασίας και διαγραφής δεδομένων από έναν πίνακα της τοπικής βάσης δεδομένων, όμως τα δεδομένα αυτά είναι ανάλογα με το χρήστη ο οποίος τη χρησιμοποιεί. Όταν ο χρήστης πραγματοποιεί συγχρονισμό με την κεντρική βάση δεδομένων, αυτά που μπορεί να δει και να επεξεργαστεί είναι μόνο όσα έχει φτιάξει ο ίδιος και όχι των άλλων χρηστών. Επίσης, κάποια από τα δεδομένα του είναι κρυπτογραφημένα έτσι ώστε με κανέναν τρόπο να μην μπορούν να διαβαστούν από κάποιον άλλο τοπικά αλλά και στην κεντρική βάση δεδομένων, ακόμη και από

τον διαχειριστή της βάσης, ο οποίος το μόνο που μπορεί να κάνει είναι να τα διαγράψει. Ο χρήστης δηλαδή αυτής της εφαρμογής έχει πλήρη δικαιώματα μόνο στα δικά του δεδομένα και σε κανενός άλλου. Τέλος, ένα επίσης σημαντικό σημείο είναι ότι στην τοπική βάση δεδομένων υπάρχουν μόνο τα δεδομένα του ενός χρήστη, ενώ στην κεντρική βάση δεδομένων υπάρχουν τα δεδομένα όλων των χρηστών.

Η τέταρτη εφαρμογή η οποία κατασκευάστηκε ονομάζεται «LogBookDive» και αποτελεί ουσιαστικά μια πρακτική εφαρμογή των δυνατοτήτων που μας παρέχει η ανάπτυξη αυτών των τεχνολογιών. Πιο συγκεκριμένα, είναι μια βοηθητική εφαρμογή για την πραγματοποίηση ασφαλών καταδύσεων στον βυθό της θάλασσας. Η εφαρμογή αυτή έχει την δυνατότητα να υπολογίζει τα επίπεδα του αζώτου στο αίμα των δυτών, αλλά και να αποθηκεύει όλες εκείνες τις πληροφορίες που χαρακτηρίζουν μία κατάδυση. Από την άλλη, η εφαρμογή αυτή παρουσιάζει παρόμοιες ιδιότητες, με την τρίτη εφαρμογή που παρουσιάστηκε προηγουμένως. Δηλαδή και εδώ ο κάθε χρήστης (δύτης) μπορεί να βλέπει μόνο τις καταδύσεις που έχει κάνει ο ίδιος, χωρίς όμως να μπορεί να δει καθόλου τις καταδύσεις των άλλων χρηστών (δυτών). Έτσι λοιπόν, με αυτό τον τρόπο διασφαλίζεται το απόρρητο των προσωπικών του δεδομένων.

Οι τέσσερις αυτές εφαρμογές προγραμματιστικά πραγματοποιήθηκαν με την χρήση του Visual Studio 2005 σε γλώσσα προγραμματισμού C# και με βάση το Compact Framework v2.0. Ωστόσο, οι διάφορες δοκιμές έγιναν σε προσομοίωση με την χρήση του SDK Windows Mobile 5.0 Pocket PC. Η κεντρική βάση δεδομένων η οποία χρησιμοποιήθηκε είναι ο SQL Server 2005, ενώ για την τοπική-φορητή βάση δεδομένων έγινε χρήση του SQL Server 2005 Mobile Edition. Για την ομαλή λειτουργία ανάμεσα στην πρώτη και στην δεύτερη εφαρμογή καθώς επίσης και για την ασφάλεια στην τρίτη έγινε χρήση Triggers στην κεντρική βάση δεδομένων. Από την άλλη η επικοινωνία ανάμεσα στις εφαρμογές και στην κεντρική βάση δεδομένων πραγματοποιήθηκε πάνω από το πρωτόκολλο HTTPS για την ασφαλή μεταφορά των δεδομένων. Τέλος, λόγω της χρήσης του παραπάνω πρωτόκολλου χρειάστηκε η χρήση ενός Web Server όπως είναι ο IIS v5.0 και η χρήση Server Certification (RSA).

Έστερα από την ανάπτυξη και την εφαρμογή όλων αυτών των καινούργιων τεχνολογιών θα μπορούσε να πει κανείς ότι θα μπορούσαν να χρησιμοποιηθούν με μεγάλη επιτυχία σε πολλούς εργασιακούς χώρους, ακόμη μάλιστα και στην καθημερινή μας ζωή. Ο προγραμματισμός με την χρήση της γλώσσας C# αποδείχθηκε ένας πολύ εύκολος τρόπος κατασκευής αξιόλογων εφαρμογών σε μικρό σχετικά χρόνο και με εξαιρετικές μάλιστα δυνατότητες. Από την άλλη η ιδέα του συγχρονισμού ανάμεσα σε μία βάση δεδομένων που βρίσκεται μέσα σε μία φορητή συσκευή όπως είναι ένα PDA με μια απομακρυσμένη κεντρική βάση θα μπορούσε να εξυπηρετήσει σε πάρα πολλά πρακτικά θέματα όπως π.χ.: στην εύκολη αναζήτηση μέσα σε μία πολύ μεγάλη αποθήκη, στην πραγματοποίηση παραγγελιών από τους διάφορους τεχνικούς μίας εταιρίας οι οποίοι δουλεύουν σε μία απομακρυσμένη περιοχή εκτός του χώρου της εταιρίας κ.τ.λ.. Επιπλέον, η χρήση του υψηλού αυτού πρωτόκολλου (HTTPS) επικοινωνίας που χρησιμοποιείται κάνει την επικοινωνία πιο εύκολη και ασφαλή χωρίς να γίνει χρήση κάποιων πιο περιέργων πρωτόκολλων που πιθανόν να μην δούλευαν κιόλας στις φορητές συσκευές λόγω των περιορισμένων δυνατοτήτων που διαθέτουν. Έτσι λοιπόν, οι παραπάνω τεχνολογίες είναι αρκετά ώριμες για την ανάπτυξη εφαρμογών βάσεων δεδομένων σε φορητές πλατφόρμες.

Πίνακας Περιεχομένων

1. Βασικές Έννοιες	1
Εισαγωγή.....	1
1.1 Βάσεις δεδομένων και εφαρμογές	1
1.2 Φορητές βάσεις δεδομένων.....	2
1.2.1 Εισαγωγή στις φορητές βάσεις δεδομένων	2
1.2.2 Ορισμός.....	2
1.2.3 Μέρη ενός φορητού περιβάλλοντος βάσεων δεδομένων	2
1.2.4 Θέματα διαχείρισης & επικοινωνίας στις φορητές βάσεις δεδομένων	3
1.2.5 Φορητό (Mobile) DBMS	3
1.3 Γλώσσα SQL και Trigger.....	4
1.3.1 Γλώσσα SQL.....	4
1.3.2 Trigger.....	5
1.4 Φορητές πλατφόρμες	5
1.4.1 Windows Mobile (CE) 5.0.....	5
1.4.2 Pocket PC.....	6
1.5 SQL Server 2005.....	7
1.5.1 Database Engine.....	7
1.5.2 Replication	8
1.5.3 SQL Server Management Studio	8
1.6 SQL Server Mobile Edition	9
1.7 Web server IIS	10
1.8 Τύποι και Αρχιτεκτονική Replication.....	10
1.8.1 Τύποι Replication.....	10
1.8.2 Επιλογή Replication για τον SQL Server Mobile Edition	11
1.8.3 Merge Replication.....	11
1.8.4 Αρχιτεκτονική Replication	12
1.9 Πώς δουλεύει το Replication	13
1.9.1 Δημοσίευση δεδομένων	14
1.9.2 Δημιουργία Subscriptions για το Publication	14
1.9.3 Αναπροσαρμογή δεδομένων στον Subscriber	15
1.9.4 Συγχρονισμός δεδομένων	15
1.10 Εργαλεία Ανάπτυξης.....	16
1.10.1 Visual Studio 2005.....	16
1.10.2 .NET Compact Framework 2.0.....	17
1.10.3 ActiveSync.....	18
1.10.4 Windows Mobile 5.0 SDK για Pocket PC	18
1.11 Λογική σχηματική απεικόνιση των εργαλείων ανάπτυξης.....	19
2. Ασφάλεια Δεδομένων.....	21
Εισαγωγή.....	21
2.1 Βασικές έννοιες της κρυπτογραφίας.....	21
2.1.1 Χρήση της κρυπτογραφίας στην ανταλλαγή δεδομένων	21
2.1.2 Βασική ιδέα της κρυπτογραφίας.....	21
2.1.3 Οι έννοιες Authentication, Integrity και Nonrepudiation	22
2.1.4 Αλγόριθμοι και κλειδιά της κρυπτογραφίας	23
2.1.5 Συμμετρικοί Αλγόριθμοι.....	24
2.1.6 Αλγόριθμοι δημόσιου-κλειδιού	25
2.2 Αλγόριθμος κρυπτογραφίας AES	25

2.2.1 Εισαγωγή.....	25
2.2.2 Αλγόριθμος κρυπτογράφησης AES	26
2.2.3 Αλγόριθμος αποκρυπτογράφησης AES	27
2.2.4 Ασφάλεια AES.....	27
2.3 Αλγόριθμοι κρυπτογραφίας Hash και SHA.....	27
2.3.1 Μονόδρομες Hash συναρτήσεις	27
2.3.2 SHA hash συναρτήσεις	28
2.3.3 SHA-1	29
2.3.4 Ασφάλεια του SHA.....	30
2.4 Αλγόριθμος κρυπτογραφίας RSA	30
2.4.1 Αλγόριθμος RSA	30
2.4.2 Ασφάλεια του RSA	31
2.5 Πιστοποιητικό δημοσίου κλειδιού (Public key certificate)	31
2.6 Πρωτόκολλα ασφαλείας	32
2.6.1 TLS/SSL	32
2.6.2 HTTPS	32
3. Εφαρμογή «Anakoinwseis_DB»	34
Περίληψη Κεφαλαίου	34
3.1 Εισαγωγή.....	34
3.2 Τι κάνει η εφαρμογή;	34
3.3 Πώς δουλεύει η εφαρμογή;	35
3.3.1 Παρουσίαση των περιεχομένων της τοπικής βάσης δεδομένων	36
3.3.2 Εισαγωγή μιας καινούριας ανακοίνωσης.....	36
3.3.3 Παρουσίαση και επεξεργασία μιας υπάρχουσας ανακοίνωσης.....	37
3.3.4 Διαγραφή μιας υπάρχουσας ανακοίνωσης.....	38
3.3.5 Συγχρονισμός με την κεντρική βάση δεδομένων	39
3.3.6 Ρυθμίσεις του συγχρονισμού	40
3.3.7 Οδηγίες της εφαρμογής.....	41
3.4 Διαδικασία κατασκευής	42
3.4.1 Κατασκευή κεντρικής βάσης δεδομένων και πίνακα	43
3.4.2 Δημιουργία Publication.....	43
3.4.3 Ρύθμιση του δικτυακού συγχρονισμού	43
3.4.4 Κατασκευή τοπικής βάσης δεδομένων και δημιουργία Subscription.....	44
3.4.5 Κατασκευή Εφαρμογής.....	44
3.4.6 Τεχνικά χαρακτηριστικά	46
3.5 Συμπεράσματα	46
4. Εφαρμογή «Anakoinwseis_RDB».....	47
Περίληψη κεφαλαίου	47
4.1 Εισαγωγή.....	47
4.2 Τι κάνει η εφαρμογή;	47
4.3 Πώς δουλεύει η εφαρμογή;	48
4.3.1 Παρουσίαση των περιεχομένων της τοπικής βάσης δεδομένων	48
4.3.2 Παρουσίαση μιας υπάρχουσας ανακοίνωσης	49
4.3.3 Συγχρονισμός με την κεντρική βάση δεδομένων	50
4.3.4 Ρυθμίσεις του συγχρονισμού	51
4.3.5 Οδηγίες της εφαρμογής.....	52
4.4 Διαδικασία κατασκευής	53
4.4.1 Δημιουργία του πίνακα και των triggers στην κεντρική βάση δεδομένων.....	54

4.4.2 Δημιουργία Publication.....	55
4.4.3 Τεχνικά χαρακτηριστικά	55
4.5 Συμπεράσματα	56
5. Εφαρμογή «Anakoinwseis_DB_ENCR»	57
Περίληψη κεφαλαίου	57
5.1 Εισαγωγή.....	57
5.2 Τι κάνει η εφαρμογή;	58
5.3 Πώς δουλεύει η εφαρμογή;	58
5.3.1 Παρουσίαση των περιεχομένων της τοπικής βάσης δεδομένων	59
5.3.2 Εισαγωγή μιας καινούριας ανακοίνωσης.....	59
5.3.3 Παρουσίαση και επεξεργασία μιας υπάρχουσας ανακοίνωσης.....	60
5.3.4 Διαγραφή μιας υπάρχουσας ανακοίνωσης.....	61
5.3.5 Συγχρονισμός με την κεντρική βάση δεδομένων	62
5.3.6 Ρυθμίσεις του συγχρονισμού	63
5.3.7 Οδηγίες της εφαρμογής.....	64
5.4 Διαδικασία κατασκευής	65
5.4.1 Δημιουργία του πίνακα και των triggers στην κεντρική βάση δεδομένων.....	66
5.4.2 Δημιουργία Publication.....	67
5.4.3 Κατασκευή Εφαρμογής.....	67
5.4.4 Τεχνικά χαρακτηριστικά	69
5.5 Συμπεράσματα	69
6. Εφαρμογή «LogBookDive»	71
Περίληψη κεφαλαίου	71
6.1 Εισαγωγή.....	71
6.2 Τι κάνει η εφαρμογή;	72
6.3 Πώς δουλεύει η εφαρμογή;	73
6.3.1 Έλεγχος εισόδου στην εφαρμογή.....	73
6.3.2 Υπολογισμός και αποθήκευση μιας κατάδυσης	74
6.3.3 Παρουσίαση των περιεχομένων της τοπικής βάσης δεδομένων	75
6.3.4 Παρουσίαση μιας υπάρχουσας κατάδυσης.....	76
6.3.5 Διαγραφή μιας υπάρχουσας κατάδυσης	77
6.3.6 Συγχρονισμός με την κεντρική βάση δεδομένων	78
6.3.7 Ρυθμίσεις του συγχρονισμού	79
6.3.8 Οδηγίες της εφαρμογής.....	80
6.4 Διαδικασία κατασκευής	82
6.4.1 Δημιουργία του πίνακα και των triggers στην κεντρική βάση δεδομένων.....	82
6.4.2 Κατασκευή Εφαρμογής.....	83
6.4.3 Τεχνικά χαρακτηριστικά	84
6.5 Συμπεράσματα	84
7. Γενικά Συμπεράσματα.....	86
Παράρτημα	89
Α. Βήματα κατασκευής μιας SQL φορητής εφαρμογής με την βοήθεια του Visual Studio 2005 και του SQL Server 2005	89
Β. Επεξεργαστής OMAP850	90
C. Βοηθητικά εργαλεία	93
C.1 Δημιουργία Certificates με χρήση του OpenSSL.....	93

C.2 DynDNS	97
C.3 Ethereal (Sniffer).....	97
Βιβλιογραφία	99

Πίνακας Σχημάτων

Κεφάλαιο 1

Σχήμα 1.1: Αρχιτεκτονική του φορητού περιβάλλοντος βάσεων δεδομένων.	3
Σχήμα 1.2: Διάγραμμα αρχιτεκτονικής του λειτουργικού συστήματος Windows CE. .	6
Σχήμα 1.3: Pocket PC (Phone).....	7
Σχήμα 1.4: SQL Server Management Studio.....	8
Σχήμα 1.5: Query Analyzer 3.0.	9
Σχήμα 1.6: Αρχιτεκτονική Replication.	12
Σχήμα 1.7: Visual Studio 2005.	17
Σχήμα 1.8: ActiveSync.	18
Σχήμα 1.9: Windows Mobile 5.0 SDK για Pocket PC.	19
Σχήμα 1.10: Λογική σχηματική απεικόνιση με χρήση Emulator.	20
Σχήμα 1.11: Λογική σχηματική απεικόνιση με χρήση αληθινής συσκευής.	20

Κεφάλαιο 2

Σχήμα 2.1: Κρυπτογράφηση και Αποκρυπτογράφηση.....	21
Σχήμα 2.2: Κρυπτογράφηση και αποκρυπτογράφηση με ένα κλειδί.	23
Σχήμα 2.3: Κρυπτογράφηση και αποκρυπτογράφηση με δύο διαφορετικά κλειδιά. ..	24
Σχήμα 2.4: Μονόδρομη Hash συνάρτηση.	28
Σχήμα 2.5: Μία επανάληψη της συνάρτησης συμπίεσης του SHA-1.	29

Κεφάλαιο 3

Σχήμα 3.1: Καρτέλα «List» της εφαρμογής «Anakoinwseis_DB».	36
Σχήμα 3.2: Καρτέλα «Insert» της εφαρμογής «Anakoinwseis_DB».	37
Σχήμα 3.3: Καρτέλα «Edit» της εφαρμογής «Anakoinwseis_DB».	38
Σχήμα 3.4: Καρτέλα «Delete» της εφαρμογής «Anakoinwseis_DB».	39
Σχήμα 3.5: Καρτέλα «Sync» της εφαρμογής «Anakoinwseis_DB».....	40
Σχήμα 3.6: Καρτέλα «Setting» της εφαρμογής «Anakoinwseis_DB».	41
Σχήμα 3.7: Καρτέλα «Help» της εφαρμογής «Anakoinwseis_DB».....	42
Σχήμα 3.8: Φόρμα «About» της εφαρμογής «Anakoinwseis_DB».....	42

Κεφάλαιο 4

Σχήμα 4.1: Καρτέλα «List» της εφαρμογής «Anakoinwseis_RDB».....	49
Σχήμα 4.2: Καρτέλα «View» της εφαρμογής «Anakoinwseis_RDB».	50
Σχήμα 4.3: Καρτέλα «Sync» της εφαρμογής «Anakoinwseis_RDB».....	51
Σχήμα 4.4: Καρτέλα «Setting» της εφαρμογής «Anakoinwseis_RDB».....	52
Σχήμα 4.5: Καρτέλα «Help» της εφαρμογής «Anakoinwseis_RDB».	53
Σχήμα 4.6: Φόρμα «About» της εφαρμογής «Anakoinwseis_RDB».	53

Κεφάλαιο 5

Σχήμα 5.1: Καρτέλα «List» της εφαρμογής «Anakoinwseis_DB_ENCR».....	59
Σχήμα 5.2: Καρτέλα «Insert» της εφαρμογής «Anakoinwseis_DB_ENCR».....	60
Σχήμα 5.3: Καρτέλα «Edit» της εφαρμογής «Anakoinwseis_DB_ENCR».	61

Σχήμα 5.4: Καρτέλα «Delete» της εφαρμογής «Anakoinwseis_DB_ENCR».	62
Σχήμα 5.5: Καρτέλα «Sync» της εφαρμογής «Anakoinwseis_DB_ENCR».	63
Σχήμα 5.6: Καρτέλα «Setting» της εφαρμογής «Anakoinwseis_DB_ENCR».	64
Σχήμα 5.7: Καρτέλα «Help» της εφαρμογής «Anakoinwseis_DB_ENCR».	65
Σχήμα 5.8: Φόρμα «About» της εφαρμογής «Anakoinwseis_DB_ENCR».	65

Κεφάλαιο 6

Σχήμα 6.1: Φόρμα «Password» της εφαρμογής «LogBookDive».	73
Σχήμα 6.2: Καρτέλα «Dive1-2» της εφαρμογής «LogBookDive».	75
Σχήμα 6.3: Καρτέλα «History» της εφαρμογής «LogBookDive».	76
Σχήμα 6.4: Καρτέλα «View» της εφαρμογής «LogBookDive».	77
Σχήμα 6.5: Καρτέλα «Delete» της εφαρμογής «LogBookDive».	78
Σχήμα 6.6: Καρτέλα «Sync» της εφαρμογής «LogBookDive».	79
Σχήμα 6.7: Καρτέλα «Setting» της εφαρμογής «LogBookDive».	80
Σχήμα 6.8: Καρτέλα «Help» της εφαρμογής «LogBookDive».	81
Σχήμα 6.9: Φόρμα «About» της εφαρμογής «LogBookDive».	81

Παράρτημα

Σχήμα B.1: Μπλοκ διάγραμμα του επεξεργαστή OMAP850.	90
Σχήμα C.1: Περιβάλλον λειτουργία του Ethereal.	98

Πίνακας Πινάκων

Κεφάλαιο 1

Πίνακας 1.1: Όροι και ορισμοί του Replication.	14
Πίνακας 2.1: Οι τρεις επίσημες εκδόσεις του AES.	26

Κεφάλαιο 2

Πίνακας 2.2: Τα διάφορα μεγέθη του SHA.	29
---	----

1. Βασικές Έννοιες

Εισαγωγή

Στο παρόν κεφάλαιο γίνεται παρουσίαση όλων εκείνων των βασικών εννοιών που απαιτούνται να γνωρίζει κάποιος για να μπορέσει να κατανοήσει με ευκολία τα κεφάλαια που ακολουθούν.

1.1 Βάσεις δεδομένων και εφαρμογές

Ο όρος βάση δεδομένων (Database) είναι μια συλλογή από σχετιζόμενα δεδομένα. Με τον όρο δεδομένα εννοούμε γνωστά γεγονότα που μπορούν να καταγραφούν και που έχουν κάποια υπονοούμενη σημασία. Ο πιο πάνω ορισμός της βάσης δεδομένων είναι αρκετά γενικός. Για παράδειγμα, μπορούμε να θεωρήσουμε ότι οι λέξεις ενός κειμένου σχετίζονται μεταξύ τους και κατά συνέπεια συνιστούν μια βάση δεδομένων. Ωστόσο, η συνήθης χρήση του όρου βάση δεδομένων είναι αρκετά πιο περιορισμένη [1]. Μια βάση δεδομένων έχει τις ακόλουθες υπονοούμενες ιδιότητες:

- Μια βάση δεδομένων είναι μια λογικά συνεκτική συλλογή δεδομένων που έχει κάποια εγγενή σημασία. Μια τυχαία διευθέτηση δεδομένων δεν είναι σωστό να αναφέρεται ως βάση δεδομένων.
- Μια βάση δεδομένων σχεδιάζεται, κατασκευάζεται και γεμίζει με δεδομένα για κάποιο συγκεκριμένο σκοπό. Προορίζεται για μια συγκεκριμένη ομάδα χρηστών και για κάποιες προκαθορισμένες εφαρμογές για τις οποίες οι χρήστες αυτοί ενδιαφέρονται.

Οι βάσεις δεδομένων γενικά χρησιμοποιούνται ευρέως και αποτελούν ένα απαραίτητο μέρος σχεδόν κάθε επιχείρησης σήμερα [2]. Εδώ θα αναφερθούν μερικές αντιπροσωπευτικές εφαρμογές:

- Τράπεζες: Για πληροφορίες πελατών, λογαριασμών και δανείων και τραπεζικών συναλλαγών.
- Αεροπορικές εταιρείες: Για κρατήσεις θέσεων και πληροφορίες πτήσεων. Οι αεροπορικές εταιρείες ήταν μεταξύ των πρώτων που χρησιμοποίησαν βάσεις δεδομένων.
- Πανεπιστήμια: Για πληροφορίες φοιτητών, εγγραφές σε μαθήματα και βαθμούς.
- Πωλήσεις: Για πληροφορίες πελατών, προϊόντων και πωλήσεων.
- Χρηματοδοτήσεις: Για αποθήκευση πληροφοριών σχετικά με πωλήσεις και αγορές οικονομικών στοιχείων, όπως μετοχών και ομολόγων.
- Ανθρώπινοι πόροι: Για πληροφορίες για εργαζόμενους, μισθούς, φόρους μισθοδοσίας και παροχές και για πληρωμές μισθών.
- Φορητές συσκευές: Για την διαχείριση προσωπικών και εταιρικών δεδομένων, τόσο τοπικά στη φορητή πλατφόρμα, όσο και σε επικοινωνία με την κεντρική βάση δεδομένων.

1.2 Φορητές βάσεις δεδομένων

1.2.1 Εισαγωγή στις φορητές βάσεις δεδομένων

Λόγω του αυξανόμενου αριθμού των κινούμενων εργαζομένων επιβεβαιώνεται μία αυξανόμενη ζήτηση στον φορητό υπολογισμό (mobile computing). Τέτοια άτομα θέλουν να δουλεύουν από απομακρυσμένες περιοχές σαν να ήταν στο γραφείο τους, όπως για παράδειγμα από τα σπίτια τους, τις εγκαταστάσεις των πελατών, ή απλά ενώ βρίσκονται καθοδόν προς απομακρυσμένες περιοχές. Το "γραφείο" μπορεί να συνοδεύσει έναν κινούμενο εργαζόμενο υπό την μορφή ενός Laptop, ενός PDA (Personal Digital Assistant = Προσωπικός Ψηφιακός Βοηθός), ή μίας άλλης συσκευής για πρόσβαση στο διαδίκτυο. Λόγω της γρήγορης επέκτασης των κυψελοειδών (κινητή τηλεφωνία), ασύρματων (Wireless LAN), και δορυφορικών επικοινωνιών, θα είναι σύντομα δυνατό για τους κινούμενους χρήστες να έχουν πρόσβαση σε οποιαδήποτε δεδομένα, οπουδήποτε και εάν βρίσκονται και σε οποιαδήποτε στιγμή. Εντούτοις, λόγω της πρακτικότητας, της ασφάλειας, και των δαπανών η επικοινωνία είναι ακόμη περιορισμένη, με αποτέλεσμα οι χρήστες να μην μπορούν να είναι συνέχεια συνδεδεμένοι στο διαδίκτυο και γενικά όποτε αυτοί το επιθυμούν. Έτσι λοιπόν οι φορητές βάσεις δεδομένων προσφέρουν μια λύση για μερικούς από αυτούς τους περιορισμούς [3][4][5].

1.2.2 Ορισμός

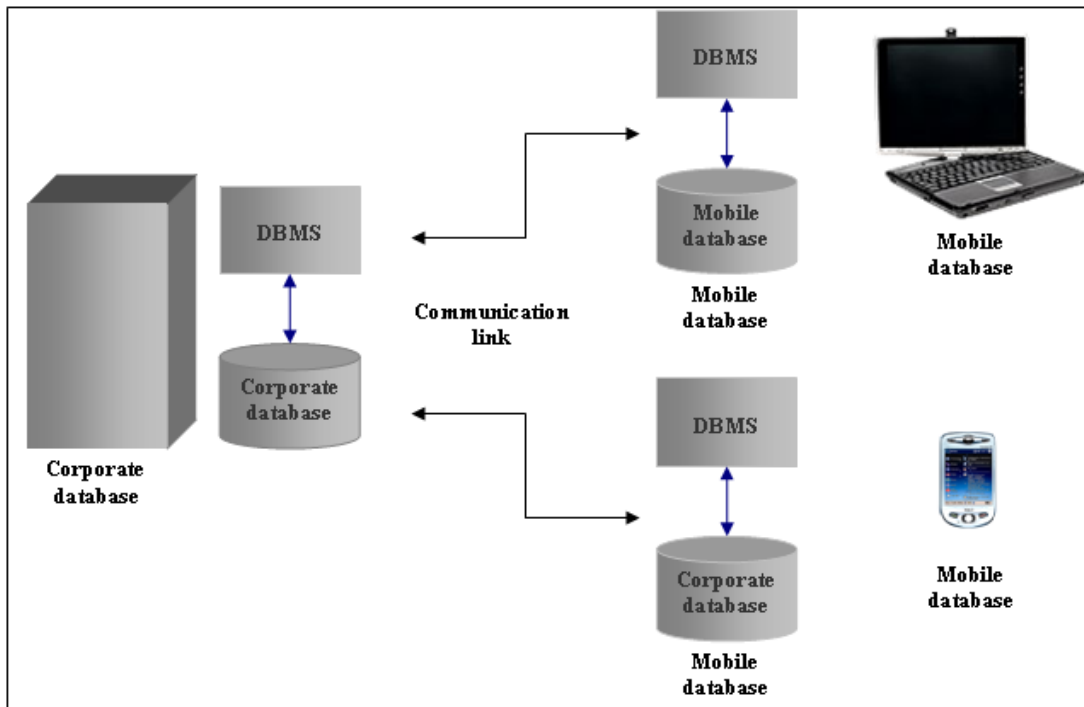
Μία φορητή βάση δεδομένων είναι ουσιαστικά μια βάση δεδομένων που μπορεί να εκτελείται σε μια φορητή πλατφόρμα. Κατά κανόνα η φορητή βάση δεδομένων αποτελεί τμήμα (φυσικά διαχωρισμένο) της κεντρικής βάσης δεδομένων που είναι εγκατεστημένη σε έναν κεντρικό υπολογιστή, με την δυνατότητα όμως συγχρονισμού της από απομακρυσμένες περιοχές, όταν είναι διαθέσιμο κάποιο δίκτυο επικοινωνίας [3].

1.2.3 Μέρη ενός φορητού περιβάλλοντος βάσεων δεδομένων

Με τις φορητές βάσεις δεδομένων, οι χρήστες έχουν πρόσβαση στα κεντρικά - εταιρικά δεδομένα με την χρήση φορητών υπολογιστικών συσκευών μέσω του διαδικτύου από απομακρυσμένες περιοχές [3]. Η χαρακτηριστική αρχιτεκτονική για ένα φορητό περιβάλλον βάσεων δεδομένων παρουσιάζεται στο Σχήμα 1.1. Ένα τέτοιο περιβάλλον αποτελείται από τα παρακάτω μέρη:

- Τον κεντρικό εταιρικό υπολογιστή βάσεων δεδομένων και το DBMS (Data Base Management System = Πρόγραμμα διαχείρισης βάσεων δεδομένων) που διαχειρίζεται και αποθηκεύει τα εταιρικά δεδομένα και παρέχει τις εταιρικές εφαρμογές.
- Την απομακρυσμένη βάση δεδομένων και το DBMS που διαχειρίζεται και αποθηκεύει τα κινητά δεδομένα και παρέχει τις κινητές εφαρμογές.
- Την κινητή πλατφόρμα βάσεων δεδομένων που περιλαμβάνει το Laptop, το PDA, ή άλλες συσκευές με πρόσβαση στο διαδίκτυο.

- Και διπλή κατεύθυνση στις συνδέσεις επικοινωνίας μεταξύ του εταιρικού και κινητού DBMS.



Σχήμα 1.1: Αρχιτεκτονική του φορητού περιβάλλοντος βάσεων δεδομένων.

1.2.4 Θέματα διαχείρισης & επικοινωνίας στις φορητές βάσεις δεδομένων

Τα δύο κύρια ζητήματα που συνδέονται με τις φορητές βάσεις δεδομένων είναι η διαχείριση της φορητής βάσης δεδομένων και η επικοινωνία μεταξύ των φορητών και κεντρικών-εταιρικών βάσεων δεδομένων.

Ανάλογα με τις ιδιαίτερες απαιτήσεις των φορητών εφαρμογών, σε μερικές περιπτώσεις ο χρήστης μιας φορητής συσκευής μπορεί να χρειαστεί σύνδεση στον κεντρικό υπολογιστή και επεξεργασία των δεδομένων σε αυτόν, ενώ σε μερικές άλλες ο χρήστης μπορεί να χρειαστεί μεταφορά των δεδομένων και επεξεργασία στην φορητή συσκευή.

Η επικοινωνία μεταξύ των κεντρικών-εταιρικών και φορητών βάσεων δεδομένων είναι συνήθως διακοπτόμενη σε ανομοιόμορφα χρονικά διάστημα. Αν και ασυνήθιστο, υπάρχουν μερικές εφαρμογές που απαιτούν συνεχή επικοινωνία μεταξύ των φορητών και των κεντρικών βάσεων δεδομένων. Στην παρακάτω παράγραφο θα προσδιορίσουμε τις απαιτήσεις του φορητού DBMS [3].

1.2.5 Φορητό (Mobile) DBMS

Οι σημαντικότεροι κατασκευαστές DBMS (π.χ. Oracle, MS SQL Server, κ.α.) προσφέρουν τώρα φορητό DBMS, που συνοδεύεται με το κανονικό DBMS που διαθέτουν. Στην πραγματικότητα, αυτή η ανάπτυξη είναι εν μέρει υπεύθυνη για την

δραματική αύξηση των πωλήσεων των σημαντικότερων κατασκευαστών DBMS. Οι περισσότεροι κατασκευαστές προωθούν το φορητό DBMS τους ως επιπλέον σημαντικό εργαλείο του DBMS και την παροχή υπηρεσιών των βάσεων δεδομένων που απαιτούν περιορισμένους υπολογιστικούς πόρους για να ταιριάζουν με εκείνες που παρέχονται αυτήν την περίοδο από τις φορητές συσκευές. Η πρόσθετη αυτή λειτουργία που απαιτείται από φορητό DBMS περιλαμβάνει τις εξής δυνατότητες:

- Επικοινωνία με την κεντρική βάση δεδομένων μέσω κάποιου ασύρματου (Wireless) ή άλλου δικτύου.
- Αντιγραφή (**replicate**) δεδομένων μεταξύ της κεντρικής βάσης δεδομένων και της φορητής συσκευής.
- Συγχρονισμός (**synchronize**) δεδομένων μεταξύ της κεντρικής βάσης δεδομένων και της φορητής συσκευής.
- Συλλογή (**capture**) δεδομένων από διάφορες πηγές όπως είναι το διαδίκτυο.
- Διαχείριση (**manage**) δεδομένων μέσα στη φορητή συσκευή.
- Ανάλυση (**analyze**) δεδομένων μέσα στη φορητή συσκευή.
- Και δημιουργία (**create**) προσαρμοσμένων φορητών εφαρμογών.

Οι κατασκευαστές DBMS οδηγούν τις τιμές των εφαρμογών αυτών σε ένα τέτοιο επίπεδο έτσι ώστε να είναι οικονομικά αποδοτικές για χρήση σε φορητές συσκευές. Αυτήν την περίοδο, τα περισσότερα φορητά DBMSs παρέχουν μόνο τις λειτουργίες της SQL για την φορητή εφαρμογή, για να είναι δυνατή η υποστήριξη οποιουδήποτε ερωτήματος στη βάση δεδομένων ή οποιασδήποτε ανάλυσης δεδομένων. Εντούτοις, η πρόβλεψη είναι ότι στο εγγύς μέλλον οι φορητές συσκευές θα διαθέτουν λειτουργίες που έχουν σήμερα οι κεντρικές βάσεις δεδομένων, αλλά και άλλες [3].

1.3 Γλώσσα SQL και Trigger

1.3.1 Γλώσσα SQL

Η γλώσσα SQL είναι μια γλώσσα προγραμματισμού (ελέγχου) των σχεσιακών βάσεων δεδομένων (DB). Η γλώσσα SQL σε ένα σύστημα βάσης δεδομένων αποτελείται από δύο μέρη, από την γλώσσα ορισμού των δεδομένων, με την οποία καθορίζεται η διάταξη της βάσης δεδομένων, και από την γλώσσα χειρισμού των δεδομένων, με την οποία εκφράζονται οι ερωτήσεις και οι ενημερώσεις στη βάση δεδομένων [2].

Έτσι λοιπόν με την γλώσσα ορισμού των δεδομένων (data-definition language - DDL) καθορίζουμε μια διάταξη βάσης δεδομένων από ένα σύνολο ορισμών που εκφράζονται από αυτή την ειδική γλώσσα. Για παράδειγμα, η παρακάτω εντολή στη γλώσσα SQL δημιουργεί τον πίνακα account:

```
Create table account (
  account-number char(10),
  balance integer
)
```

Η γλώσσα του χειρισμού των δεδομένων υποστηρίζει τις ακόλουθες λειτουργίες:

- Η ανάκληση (select) πληροφοριών που είναι αποθηκευμένες στην βάση δεδομένων.
- Η εισαγωγή (insert) νέων πληροφοριών στη βάση δεδομένων.
- Η διαγραφή (delete) πληροφοριών από τη βάση δεδομένων.
- Η τροποποίηση (update) πληροφοριών στη βάση δεδομένων.

Μια γλώσσα χειρισμού των δεδομένων (data-manipulation language - DML) είναι μια γλώσσα που επιτρέπει στους χρήστες να έχουν πρόσβαση ή να χειρίζονται δεδομένα όπως είναι οργανωμένα από το κατάλληλο μοντέλο δεδομένων [2]. Σαν παράδειγμα, το παρακάτω ερώτημα στην SQL βρίσκει το «balance» του «account» του οποίου ο «account-number» είναι '192-83-746':

```
select balance
from account
where account-number = '192-83-746'
```

1.3.2 Trigger

Ένας trigger είναι μια εντολή που εκτελείται στο σύστημα αυτόματα, ως αποτέλεσμα μιας τροποποίησης στη βάση δεδομένων. Οι βάσεις δεδομένων στις οποίες βρίσκει χρήση ονομάζονται ενεργές βάσεις δεδομένων (Active DB) [2]. Για να σχεδιαστεί ένας μηχανισμός trigger, θα πρέπει πρώτα να ικανοποιούνται οι παρακάτω δυο απαιτήσεις:

1. Να καθορίσετε πότε θα εκτελεστεί ένα trigger. Αυτό χωρίζεται σε ένα συμβάν που κάνει το trigger να ελεγχθεί και μια συνθήκη που πρέπει να ικανοποιείται για να προχωρήσει η εκτέλεση του trigger.
2. Να καθορίσετε τις ενέργειες που θα ακολουθηθούν όταν εκτελεστεί το trigger.

Το παραπάνω μοντέλο των trigger αναφέρεται ως μοντέλο συμβάντος-συνθήκης-ενέργειας για τα trigger. Η βάση δεδομένων αποθηκεύει τα trigger σαν να ήταν κανονικά δεδομένα, έτσι ώστε να είναι μόνιμα και να είναι προσπελάσιμα σε όλες τις λειτουργίες της βάσης δεδομένων. Αφού εισάγουμε ένα trigger στην βάση δεδομένων, η βάση δεδομένων έχει την ευθύνη να το εκτελέσει όποτε πραγματοποιείται το συγκεκριμένο συμβάν και ικανοποιείται η αντίστοιχη συνθήκη.

1.4 Φορητές πλατφόρμες

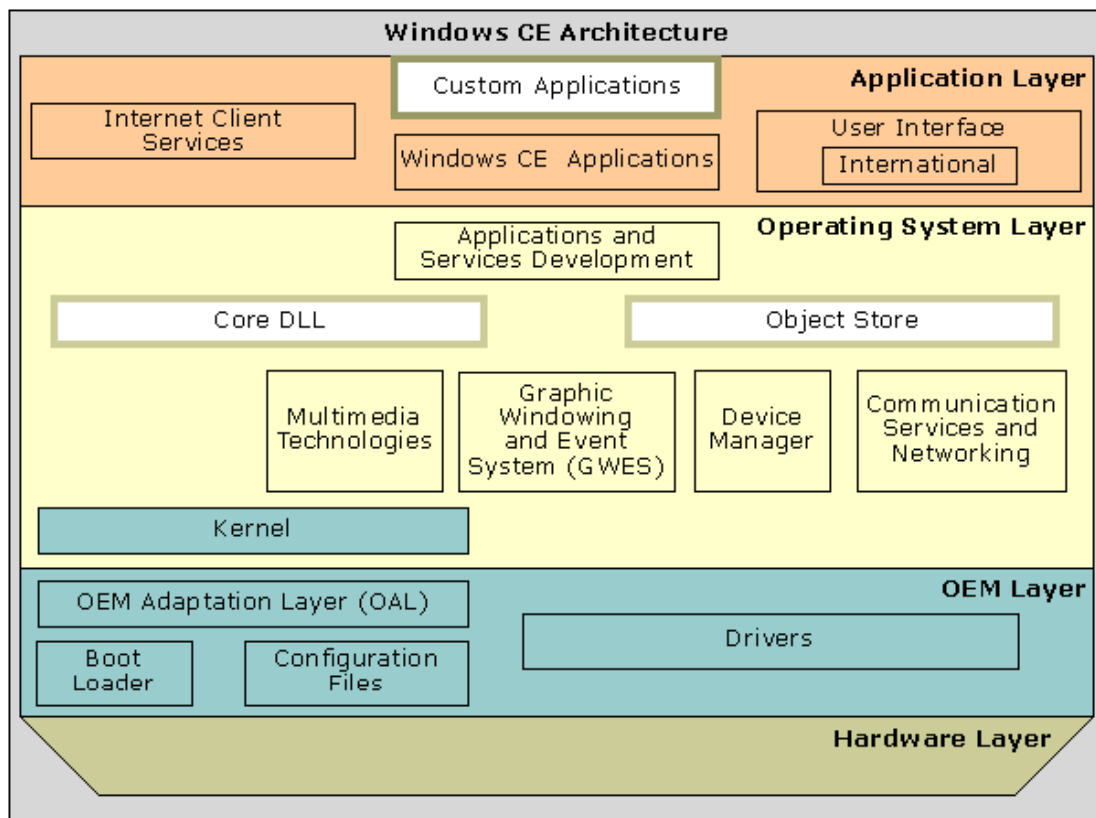
1.4.1 Windows Mobile (CE) 5.0

Τα Windows CE είναι ένα 32-bit λειτουργικό σύστημα (OS) που σχεδιάστηκε για να ικανοποιήσει τις ανάγκες μιας ευρείας σειράς έξυπνων συσκευών, από επιχειρηματικά εργαλεία όπως οι βιομηχανικοί ελεγκτές, hubs επικοινωνιών, και καταναλωτικά προϊόντα όπως οι φωτογραφικές μηχανές, οι συσκευές διαδικτύου, τα

Pocket PCs, τα έξυπνα κινητά τηλέφωνα (Smartphones) και τα φορητά κέντρα πολυμέσων (Portable Media Centers) [6].

Τα Windows CE αποτελούν μια τυπική ενσωματωμένη εικόνα (image) εκτέλεσης που στοχεύει σε μια συγκεκριμένη χρήση και αποτελεί ένα μικρό λειτουργικό σύστημα που έχει ντετερμινιστική απάντηση στο χειρισμό διακοπών (interrupts).

Τα Windows CE προσφέρουν ευκολία στην ανάπτυξη μίας εφαρμογής από τον προγραμματιστή, με την χρήση διαφόρων γλωσσών (C++, C#, κ.τ.λ.), μαζί με την ευπροσάρμοστη διεπαφή των εφαρμογών με το Microsoft Win32 (API). Προσφέρει επίσης υποστήριξη για πολυμέσα, διαδίκτυο, τοπικό δίκτυο (LAN), κινητή επικοινωνία και υπηρεσίες ασφάλειας. Το παρακάτω διάγραμμα του σχήματος 1.2 παρουσιάζει την αρχιτεκτονική του λειτουργικού συστήματος Windows CE.



Σχήμα 1.2: Διάγραμμα αρχιτεκτονικής του λειτουργικού συστήματος Windows CE.

1.4.2 Pocket PC

Το Pocket PC είναι ένας μικρός φορητός υπολογιστής που τρέχει μια συγκεκριμένη έκδοση του λειτουργικού συστήματος Windows CE. Έχει πολλές από τις ικανότητες ενός σύγχρονου υπολογιστή γραφείου (Desktop PC). Υπάρχει ήδη ένας μεγάλος αριθμός εφαρμογών για το Pocket PC, πολλές από τις οποίες είναι δωρεάν λογισμικό. Μερικές από αυτές τις συσκευές, τρέχοντας τα Windows Mobile 5.0 Phone Edition, περιλαμβάνουν επίσης χαρακτηριστικά γνωρίσματα ενός κινητού τηλεφώνου. Το Pocket PC μπορεί επίσης να χρησιμοποιηθεί σε πολλές άλλες εφαρμογές όπως είναι σαν δέκτης GPS, αναγνώστης barcode, αναγνώστης RFID,

κ.τ.λ.. Ένα ενδεικτικό παράδειγμα Pocket PC διαθέτει τα εξής τεχνικά χαρακτηριστικά: CPU ARM με Clock 200 MHz, Ram 64 MB και 128 MB Flash μνήμη (βλέπε στο Παράρτημα Β). Στο σχήμα 1.3 φαίνεται ένα τέτοιο Pocket PC [7]. Οι εφαρμογές που αναπτύχθηκαν στα πλαίσια αυτής της διπλωματικής εργασίας, εκτελέστηκαν σε ένα τέτοιο Pocket PC (Qtek 9100).



Σχήμα 1.3: Pocket PC (Phone).

1.5 SQL Server 2005

Ο SQL Server 2005 είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (RDBMS). Η γλώσσα που χρησιμοποιεί για την διατύπωση ερωτημάτων (SQL) είναι η Transact-SQL, που αποτελεί μια επέκταση της πρότυπης γλώσσας SQL. Τα εργαλεία τα οποία διαθέτει είναι: Database Engine, Analysis Services, Integration Services, Replication, Reporting Services, Notification Services, Service Broker, Full-Text Search και το SQL Server Management Studio. Στην συγκεκριμένη όμως διπλωματική εργασία αυτά που θα μας απασχολήσουν περισσότερο είναι η Database Engine, το Replication και το SQL Server Management Studio [8].

1.5.1 Database Engine

Η Database Engine είναι ουσιαστικά ο πυρήνας της βάσης δεδομένων και αποτελεί μια υπηρεσία για την αποθήκευση, την επεξεργασία και την ασφάλεια των

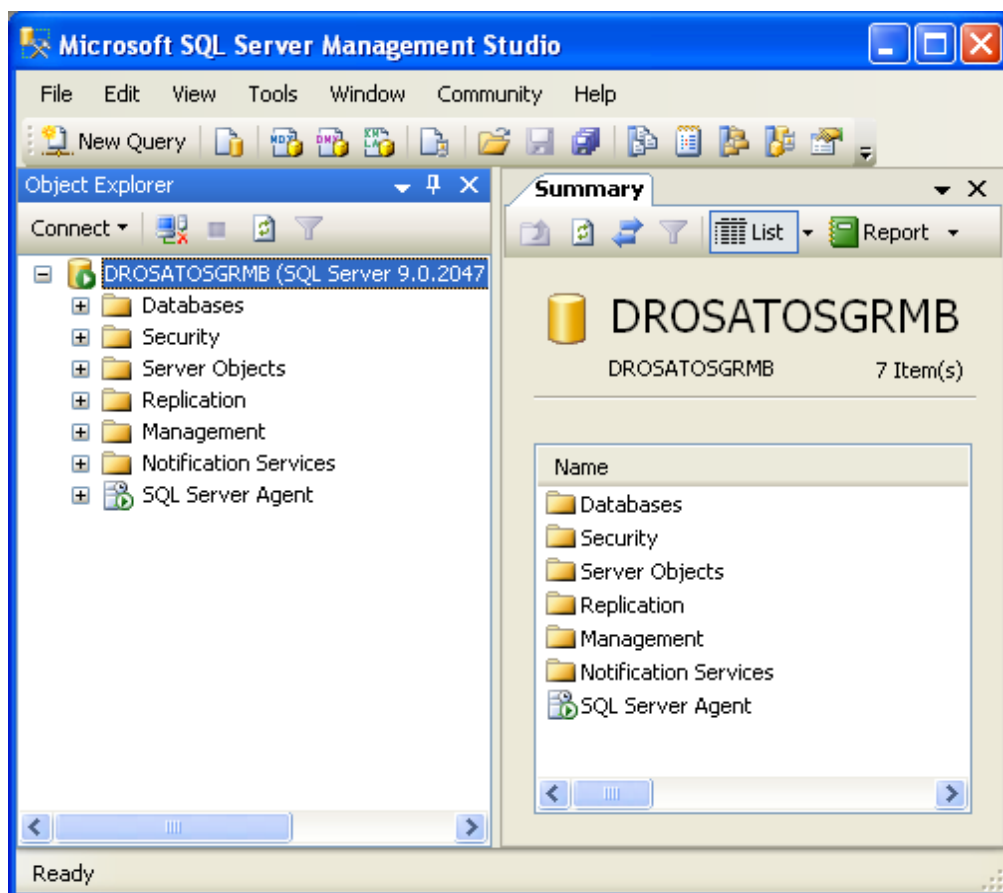
δεδομένων. Η Database Engine είναι αυτή που παρέχει ελεγχόμενη πρόσβαση για επεξεργασία στους χρήστες [9].

1.5.2 Replication

Το Replication είναι ένα σύνολο τεχνολογιών για την αντιγραφή και τη διανομή δεδομένων και ολόκληρων αντικειμένων βάσεων δεδομένων από μια βάση δεδομένων σε άλλη, και έπειτα το συγχρονισμό μεταξύ των βάσεων δεδομένων για την διατήρηση της συνοχής τους. Με τη χρησιμοποίηση του Replication, μπορείτε να διανείμετε τα δεδομένα σε διαφορετικές θέσεις και σε μακρινούς ή κινητούς χρήστες με τη βοήθεια δικτύων τοπικής και ευρείας περιοχής, συνδέσεων dial-up, ασύρματων συνδέσεων και του διαδικτύου [10].

1.5.3 SQL Server Management Studio

Το SQL Server Management Studio είναι ένα περιβάλλον για τη πρόσβαση, τη διαμόρφωση, τη διαχείριση, και τη χορήγηση όλων των δυνατοτήτων του SQL Server. Το SQL Server Management Studio συνδυάζει μια ομάδα γραφικών εργαλείων με έναν συντάκτη κειμένου για να παρέχει πρόσβαση στους προγραμματιστές και διαχειριστές του SQL Server. Το SQL Server Management Studio παρουσιάζεται στο σχήμα 1.4.



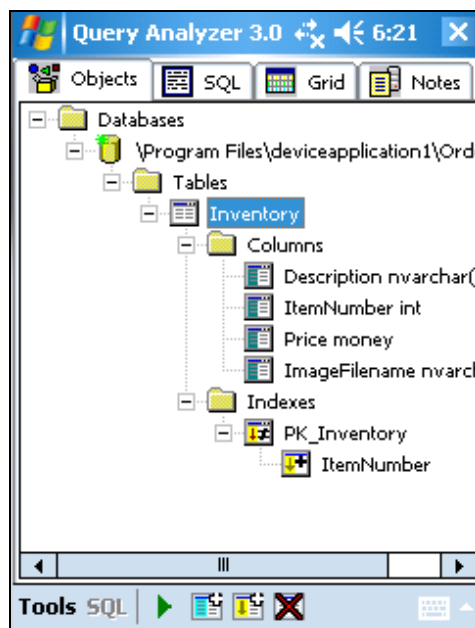
Σχήμα 1.4: SQL Server Management Studio.

1.6 SQL Server Mobile Edition

Ο SQL Server Mobile Edition είναι μία μικρή (compact) βάση δεδομένων για την εύκολη ανάπτυξη εφαρμογών διαχείρισης δεδομένων σε φορητές συσκευές (π.χ. Pocket PC). Ο SQL Server Mobile είναι ένα ισχυρό εργαλείο που το καθιστά εύκολο για την ανάπτυξη φορητών εφαρμογών με την υποστήριξη της πρότυπης γλώσσας SQL και την παροχή ενός προτύπου ανάπτυξης API σύμφωνα με τον SQL Server.

Η SQL Server Mobile Engine διαθέτει ένα σύνολο χαρακτηριστικών γνωρισμάτων βάσεων δεδομένων, όπως την βελτιστοποίηση ερωτημάτων επεξεργασίας και την υποστήριξη εκτέλεσης και ταξινόμησης διαφόρων τύπων δεδομένων, διατηρώντας όμως έναν μικρό (compact) καταλαμβανόμενο χώρο στους υπολογιστικούς πόρους του συστήματος. Η απομακρυσμένη πρόσβαση δεδομένων (RDA = Remote Data Access) και το Merge Replication εξασφαλίζουν ότι τα δεδομένα από την κεντρική βάση δεδομένων του SQL Server παραδίδονται χωρίς λάθη, μπορούν να διαχειριστούν χωρίς σύνδεση και επίσης να συγχρονιστούν αργότερα με την κεντρική βάση δεδομένων, κατασκευάζοντας τον SQL Server Mobile Edition κατάλληλα για φορητές και ασύρματες συσκευές [11].

Ο SQL Server Mobile είναι ενσωματωμένος στο .NET Compact Framework όπου μέσα από το Visual Studio 2005, απλοποιεί την ανάπτυξη εφαρμογών βάσεων δεδομένων για έξυπνες συσκευές. Επιπλέον, χρησιμοποιώντας την εφαρμογή διαχείρισης δεδομένων (Query Analyzer) του SQL Server Mobile Edition σε συνδυασμό με την εφαρμογή που έχουμε φτιάξει οι ίδιοι με το Visual Studio μπορούμε να καταφέρουμε μια ικανοποιητική διαχείριση των δεδομένων στην φορητή μας συσκευή (π.χ. Pocket PC, Smartphone, κ.τ.λ.). Στο σχήμα 1.5 παρουσιάζεται μια ενδεικτική εικόνα του Query Analyzer 3.0. Στην πρώτη καρτέλα (Objects) του Query Analyzer εμφανίζεται η τοπική βάση δεδομένων της φορητής συσκευής μαζί με τους πίνακες και τις στήλες τους από τις οποίες αποτελούνται. Τέλος, στην δεύτερη καρτέλα (SQL) εκτελούμε τα SQL ερωτήματα και στην τρίτη (Grid) εμφανίζονται τα αποτελέσματα των ερωτημάτων, ενώ στην τέταρτη καρτέλα (Notes) εμφανίζονται τα διάφορα συμβάντα της εκτέλεσης.



Σχήμα 1.5: Query Analyzer 3.0.

1.7 Web server IIS

Ο IIS (Internet Information Services) είναι ένας διακομιστής διαδικτύου (Web Server) με πλήθος δικτυακών υπηρεσιών και είναι ενσωματωμένος σαν εφαρμογή (System Software) σε διάφορες εκδόσεις του λειτουργικού συστήματος Microsoft Windows Server. Αυτή την περίοδο ο IIS περιλαμβάνει υπηρεσίες όπως FTP, SMTP, NNTP και HTTP/HTTPS. Στην παρούσα διπλωματική εργασία θα μας απασχολήσει κυρίως η υπηρεσία HTTPS για την ασφαλή επικοινωνία της κεντρικής βάσης δεδομένων με την φορητή βάση δεδομένων με τη χρήση πιστοποιητικών (Certificates) [12].

1.8 Τύποι και Αρχιτεκτονική Replication

1.8.1 Τύποι Replication

Ο SQL Server 2005 υποστηρίζει τους ακόλουθους τύπους replication για χρήση στις διανεμημένες εφαρμογές:

- Transactional replication.
- Merge replication.
- Snapshot replication.

Ο τύπος replication που επιλέγεται για μια εφαρμογή εξαρτάται από πολλούς παράγοντες, συμπεριλαμβανομένου του φυσικού περιβάλλοντος του replication, τον τύπο και την ποσότητα δεδομένων που αντιγράφονται (replicated), και εάν τα δεδομένα ενημερώνονται στο Subscriber (συνδρομητή). Το φυσικό περιβάλλον περιλαμβάνει τον αριθμό και τη θέση των υπολογιστών που βρίσκονται στο replication και εάν αυτοί οι υπολογιστές είναι clients (πελάτες) (τερματικοί σταθμοί, laptop, ή φορητές συσκευές) ή servers (κεντρικοί υπολογιστές).

Κάθε τύπος replication αρχίζει χαρακτηριστικά με έναν αρχικό συγχρονισμό των δημοσιευμένων αντικειμένων μεταξύ του Publisher (εκδότη) και των Subscribers. Αυτός ο αρχικός συγχρονισμός μπορεί να εκτελεστεί από το replication με ένα snapshot (στιγμιότυπο), το οποίο είναι ένα αντίγραφο όλων των αντικειμένων και δεδομένων που διευκρινίζονται από ένα publication (δημοσίευση). Αφότου δημιουργείται το snapshot, παραδίδεται στους Subscribers. Για μερικές εφαρμογές, το snapshot replication είναι μόνο αυτό που απαιτείται. Για άλλους τύπους εφαρμογών, είναι σημαντικό οι επόμενες αλλαγές δεδομένων που θα συμβούν να μεταφέρονται στο Subscriber αυξητικά κατά τη διάρκεια του χρόνου. Μερικές εφαρμογές επίσης απαιτούν οι αλλαγές να μεταφέρονται από τον Subscriber πίσω στον Publisher. Το transactional replication και το merge replication παρέχουν τη λύση για αυτούς τους τύπους εφαρμογών, αντίστοιχα.

Οι αλλαγές δεδομένων δεν υποστηρίζονται από το snapshot replication, με αποτέλεσμα κάθε φορά που εφαρμόζεται ένα snapshot, να επικαλύπτει εντελώς τα υπάρχοντα δεδομένα. Το transactional replication παρακολουθεί τις αλλαγές που συμβαίνουν στον SQL Server και τις περνάει στους Subscribers, και το merge replication παρακολουθεί τις αμφίδρομες αλλαγές δεδομένων μέσω triggers και πινάκων metadata (μεταδεδομένων) [13].

1.8.2 Επιλογή Replication για τον SQL Server Mobile Edition

Το merge replication επιτρέπει την πιστή αντιγραφή δεδομένων στον SQL Server Mobile, παρέχοντας έναν κατάλληλο τρόπο στους κινητούς χρήστες να συγχρονίζουν τις αλλαγές των δεδομένων τους με μια κεντρική βάση δεδομένων του SQL Server. Πολλές εφαρμογές απαιτούν τα δεδομένα να είναι διαθέσιμα σε μακρινούς χρήστες, συμπεριλαμβανομένων των πωλητών, των οδηγών παράδοσης, κ.τ.λ.. Αυτές οι εφαρμογές περιλαμβάνουν σενάρια εφαρμογών όπως την διαχείριση σχέσης πελατών (CRM = Customer Relationship Management), την αυτοματοποίηση δύναμης πωλήσεων (SFA = Sales Force Automation), και την αυτοματοποίηση δύναμης τομέων (FFA = Field Force Automation).

Το merge replication συγχρονίζει τα δεδομένα του SQL Server Mobile χρησιμοποιώντας Web συγχρονισμό. Η μορφή στη οποία οι αλλαγές των δεδομένων παραδίδονται στον SQL Server Mobile είναι διαφορετική ανάμεσα σε διαφορετικές εκδόσεις του SQL Server, αλλά η αρχιτεκτονική είναι ουσιαστικά η ίδια: ένας Subscriber στέλνει τις αλλαγές του στον Publisher μέσω του διακομιστή διαδικτύου IIS, και λαμβάνει τις αλλαγές του Publisher μέσω του ίδιου διακομιστή διαδικτύου IIS [14].

1.8.3 Merge Replication

Το Merge Replication αρχίζει χαρακτηριστικά με ένα snapshot (στιγμιότυπο) των αντικειμένων και των δεδομένων της βάσης δεδομένων που είναι publication. Οι επόμενες αλλαγές των δεδομένων και οι τροποποιήσεις του σχήματος που γίνονται στον Publisher και στους Subscribers παρακολουθούνται με triggers. Ο Subscriber συγχρονίζεται με τον Publisher όταν συνδέεται με το δίκτυο και ανταλλάσσει όλες εκείνες τις σειρές (rows) του πίνακα που έχουν αλλάξει μεταξύ του Publisher και του Subscriber από την προηγούμενη φορά που έγινε ο συγχρονισμός.

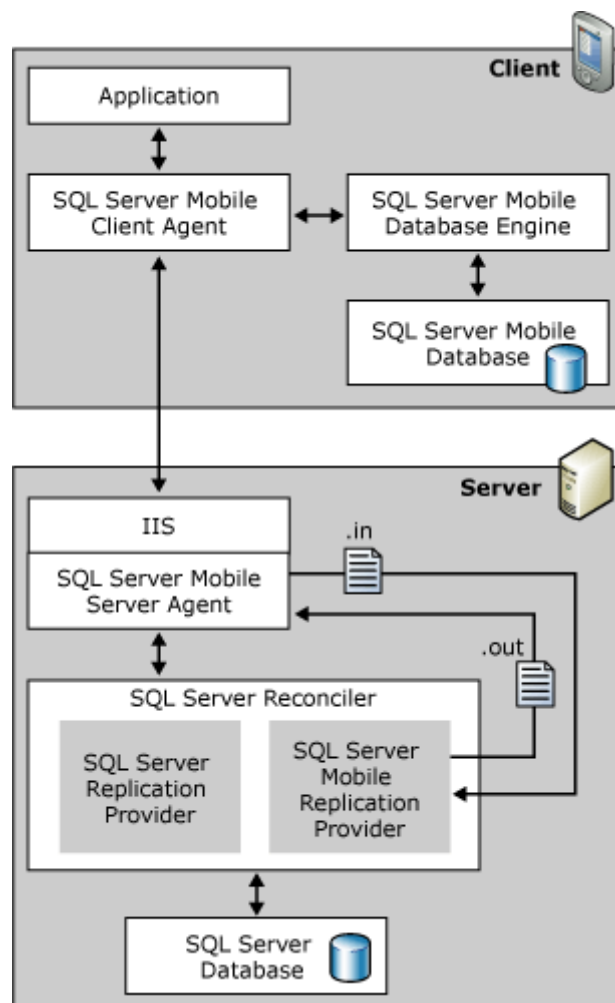
Το Merge Replication χρησιμοποιείται χαρακτηριστικά σε περιβάλλοντα server-με-client. Το Merge Replication είναι κατάλληλο σε οποιεσδήποτε από τις ακόλουθες καταστάσεις:

- Πολλαπλοί Subscribers να ενημερώνουν ίδια δεδομένα σε διαφορετικούς χρόνους και να διαδίδουν τις αλλαγές στον Publisher και μέσω αυτού σε άλλους Subscribers.
- Οι Subscribers μπορούν να λαμβάνουν τα δεδομένα, να κάνουν όποιες αλλαγές θέλουν χωρίς σύνδεση και να συγχρονίσουν αργότερα τις αλλαγές με τον Publisher και τους άλλους Subscribers.
- Κάθε Subscriber χρειάζεται ένα διαφορετικό partition (χώρισμα) των δεδομένων του.
- Εάν εμφανιστούν συγκρούσεις, υπάρχει η δυνατότητα να ανιχνευθούν και να επιλυθούν.
- Η εφαρμογή χρειάζεται μια δικτυακή αλλαγή δεδομένων παρά να έχει πρόσβαση σε ενδιάμεσες αλλαγές δεδομένων. Για παράδειγμα, εάν μια σειρά αλλάξει πέντε φορές σε έναν Subscriber προτού να συγχρονιστεί με τον Publisher, η σειρά θα αλλάξει στον Publisher μόνο για να απεικονίσει την τελευταία αλλαγή (δηλαδή την πέμπτη).

Το Merge Replication επιτρέπει στους διάφορους Subscribers να λειτουργούν αυτόνομα και μόνο οι πιο πρόσφατες αναπροσαρμογές να συγχωνεύονται σε ένα ενιαίο ομοιόμορφο αποτέλεσμα. Επειδή οι αναπροσαρμογές γίνονται σε περισσότερους από έναν Subscriber, τα ίδια δεδομένα μπορεί να είχαν ενημερωθεί στον Publisher από περισσότερους από έναν Subscribers. Επομένως, οι συγκρούσεις που μπορεί να εμφανιστούν όταν συγχωνεύονται οι αναπροσαρμογές, μπορούν να αντιμετωπιστούν με διάφορους μηχανισμούς που παρέχει το Merge Replication [15].

1.8.4 Αρχιτεκτονική Replication

Το Replication στον SQL Server 2005 Mobile Edition χρησιμοποιεί την αρχιτεκτονική που παρουσιάζεται στο σχήμα 1.6.



Σχήμα 1.6: Αρχιτεκτονική Replication.

Τα βασικά μέρη από τα οποία αποτελείται αυτή η αρχιτεκτονική Replication για τον SQL Server 2005 Mobile Edition είναι:

- **SQL Server Mobile Database Engine.** Η SQL Server Mobile Database Engine διαχειρίζεται την τοπική βάση δεδομένων για μια συσκευή. Για τις subscription βάσεις δεδομένων, η SQL Server Mobile Database Engine

παρακολουθεί όλες τις εγγραφές στην βάση δεδομένων που εισάγονται, ενημερώνονται, ή διαγράφονται με τη διατήρηση των πληροφοριών αλλαγής με κάθε εγγραφή.

- **SQL Server Mobile Client Agent.** Ο SQL Server Mobile Client Agent είναι το πρωτεύον τμήμα του Replication στον SQL Server Mobile σε μια συσκευή. Ο SQL Server Mobile Client Agent εφαρμόζει στο Replication του SQL Server Mobile το αντικείμενο διασύνδεσης. Οι εφαρμογές καλούν αυτήν την διασύνδεση για να ελέγξουν το Replication.
- **SQL Server Mobile Server Agent.** Ο SQL Server Mobile Server Agent είναι το αρμόδιο τμήμα για τη διαχείριση της επικοινωνίας μεταξύ του SQL Server και του SQL Server Mobile. Ο SQL Server Mobile Server Agent βρίσκεται στον κεντρικό υπολογιστή που τρέχει τον διακομιστή διαδικτύου IIS, και χειρίζεται όλα τα αιτήματα HTTP που υποβάλλονται από τον SQL Server Mobile Client Agent.
- **SQL Server Reconciler και SQL Server Mobile Replication Provider.** Ο SQL Server Reconciler συμπεριλαμβάνει τον SQL Server Mobile Replication Provider όταν εκτελείται ο συγχρονισμός. Ο SQL Server Reconciler και ο SQL Server Mobile Replication Provider βρίσκονται στον κεντρικό υπολογιστή που τρέχει τον IIS, στο οποίο εγκαθίστανται επίσης τα SQL Server Mobile Server Tools. Όταν ο SQL Server Reconciler ξεκινάει, ένας Merge Agent στον Publisher συνδέεται με το subscription [16].

1.9 Πώς δουλεύει το Replication

Το Merge Replication στον SQL Server 2005 Mobile Edition (κεντρικός υπολογιστής SQL κινητός) ακολουθεί την παρακάτω διαδικασία [17]:

1. Δημοσιεύονται τα δεδομένα στον SQL Server.
2. Δημιουργούνται Subscriptions στα Publications.
3. Αναπροσαρμόζονται τα δεδομένα στον Subscriber.
4. Και συγχρονίζονται τα δεδομένα.

Ο ακόλουθος πίνακας 1.1 καθορίζει μερικούς από τους όρους που χρησιμοποιούνται για το Replication.

Πίνακας 1.1: Όροι και ορισμοί του Replication.

Όρος	Ορισμός
Article	Ένας πίνακας βάσεων δεδομένων που επιτρέπεται για το Replication.
Publication	Μια συλλογή από Articles.
Partition	Ένα συγκεκριμένο σύνολο σειρών ή στηλών μέσα στο Publication.
Publisher	Μια βάση δεδομένων που καθιστά ένα Publication μέσω του Replication.
Subscriber	Μια βάση δεδομένων που λαμβάνει τα αντιγραμμένα δεδομένα και μπορεί να περάσει τις αλλαγές σε αυτά πίσω στον Publisher.
Subscription	Ένα αίτημα για ένα αντίγραφο του Publication που παραδίδεται στον Subscriber.
Subscription βάση δεδομένων	Μια βάση δεδομένων που λαμβάνει τα Articles που δημοσιεύονται από έναν Publisher.

1.9.1 Δημοσίευση δεδομένων

Τα Publications διευκρινίζουν ποια δεδομένα θα δημοσιευθούν. Τα Publications προσαρμόζουν τα γνωρίσματα με βάση τους διαφορετικούς χρήστες ή τις ομάδες χρηστών. Μερικές φορές, όλοι οι χρήστες χρειάζονται τα ίδια δεδομένα, π.χ. κάθε υπάλληλος χρειάζεται ένα αντίγραφο του καταλόγου των υπαλλήλων της επιχείρησης. Σε άλλες περιπτώσεις, οι διαφορετικές ομάδες χρηστών χρειάζονται τα διαφορετικά Partitions των δεδομένων, π.χ. οι αντιπρόσωποι πωλήσεων να χρειάζονται ένα σύνολο δεδομένων, ενώ οι τεχνικοί υποστήριξης πελατών χρειάζονται ένα διαφορετικό σύνολο δεδομένων. Επίσης κάποια άτομα χρειάζονται τα δεδομένα που τους αφορούν, π.χ. ένας αντιπρόσωπος πωλήσεων χρειάζεται μόνο τα δεδομένα για τους απολογισμούς των πελατών του.

Όταν δημιουργείται το Publication, διευκρινίζεται ποια Articles περιέχει. Αν και τα Publications του SQL Server περιέχουν άλλα αντικείμενα βάσεων δεδομένων, όπως είναι Stored Procedures, Views και καθορισμένες Functions των χρηστών, το Replication του SQL Server Mobile αγνοεί αυτά τα αντικείμενα και περιλαμβάνει μόνο τους πίνακες στο Subscription του. Τέλος, μέσω του φιλτραρίσματος μπορείτε να διευκρινίσετε ποιες σειρές και στήλες συμπεριλαμβάνονται στο Article [17].

1.9.2 Δημιουργία Subscriptions για το Publication

Ύστερα από την δημιουργία του Publication, μια εφαρμογή δημιουργεί ένα νέο Subscription στη συσκευή, είτε προγραμματιστικά με την κλήση της μεθόδου (SqlCeReplication) για το Replication στον SQL Server Mobile, είτε με τη χρησιμοποίηση του ειδικού εργαλείου (New Subscription Wizard) στο SQL Server

Management Studio. Όταν το Subscription δημιουργηθεί, το αρχικό snapshot εφαρμόζεται στην Subscription βάση δεδομένων του Subscriber [17].

1.9.3 Αναπροσαρμογή δεδομένων στον Subscriber

Το Merge Replication επιτρέπει σε κάθε Subscription της βάση δεδομένων να πραγματοποιεί αυτόνομα την διαχείριση δεδομένων (εντολές DML). Οι αναπροσαρμογές εμφανίζονται είτε η συσκευή είναι συνδεδεμένη, είτε όχι με το δίκτυο. Κάθε βάση δεδομένων του SQL Server Mobile χρησιμοποιεί για να αλλάξει τις εγγραφές της, τις ακόλουθες εντολές: INSERT, UPDATE, και DELETE [17].

1.9.4 Συγχρονισμός δεδομένων

Τυπικά, οι χρήστες συνδέουν περιοδικά τη συσκευή με το διαδίκτυο. Αυτές οι συνδέσεις επιτρέπουν στην εφαρμογή να συγχρονίζει τις αλλαγές που γίνονται στον Subscriber με τις αλλαγές που γίνονται στον Publisher. Η εφαρμογή αρχίζει το συγχρονισμό με την κλήση των μεθόδων (SqlCeReplication) συγχρονισμού που περιέχονται στο αντικείμενο του Replication. Ο συγχρονισμός είναι μια διαδικασία, η οποία περιγράφεται από τα εξής τέσσερα βήματα:

1. **Εξαγωγή των αλλαγών και δημιουργία του αρχείου μηνυμάτων εισαγωγής (input).** Ο SQL Server Mobile Client Agent εξάγει όλες τις εισαγωγές, ενημερώσεις, και διαγραφές που εγγράφηκαν στο Subscription της βάσης δεδομένων του Subscriber στον SQL Server Mobile και τους διαδίδει στον SQL Server Mobile Server Agent μέσω HTTP. Ο SQL Server Mobile Server Agent δημιουργεί ένα νέο αρχείο μηνυμάτων εισαγωγής και γράφει σε αυτό το αρχείο τις αλλαγές εισαγωγής, αναπροσαρμογής, και διαγραφής που στέλνονται από τον SQL Server Mobile Client Agent.
2. **Ο SQL Server Reconciler τρέχει την διαδικασία και εφαρμόζονται οι αλλαγές στο publication της βάση δεδομένων.** Όταν όλα τα αιτήματα γραφτούν στο αρχείο μηνυμάτων εισαγωγής, ο SQL Server Mobile Server Agent αρχίζει τον SQL Server Reconciler. Ο SQL Server Reconciler φορτώνει τον SQL Server Mobile Replication Provider, ο οποίος διαβάζει το αρχείο μηνυμάτων εισαγωγής και ενημερώνει τον SQL Server Reconciler για τις αλλαγές που έγιναν στο Subscription της βάσης δεδομένων του SQL Server Mobile και που πρέπει να εφαρμοστούν στο Publication της βάσης δεδομένων του Publisher. Κατά τη διάρκεια της επεξεργασίας, ο SQL Server Reconciler ανιχνεύει και επιλύει τις συγκρούσεις. Όταν εμφανιστεί μια σύγκρουση ενημερώνει περισσότερους από ένας Subscriber ή Publisher με την ίδια εγγραφή.
3. **Δημιουργία του αρχείου μηνυμάτων εξαγωγής (output).** Ο SQL Server Reconciler ενημερώνει τον SQL Server Mobile Replication Provider για τις αλλαγές που έγιναν στον Publisher και που πρέπει να εφαρμοστούν στο Subscription της βάση δεδομένων στην συσκευή. Ο SQL Server Mobile Replication Provider γράφει αυτές τις αλλαγές σε ένα αρχείο μηνυμάτων εξαγωγής στον κεντρικό υπολογιστή που τρέχει τον IIS.
4. **Ανάγνωση του αρχείου μηνυμάτων εξαγωγής και εφαρμογή των αλλαγών στο Subscription της βάσης δεδομένων του SQL Server Mobile.** Όταν ο

SQL Server Reconciler τελειώσει την διαδικασία, ο SQL Server Mobile Server Agent εντοπίζει το αρχείο μηνυμάτων εξαγωγής που δημιουργήθηκε από τον SQL Server Mobile Replication Provider. Αυτό το αρχείο περιέχει τις αλλαγές που έχουν εμφανιστεί στον Publisher και που πρέπει να εφαρμοστούν Subscription της βάση δεδομένων στη συσκευή. Ο SQL Server Mobile Server Agent διαβάζει το αρχείο μηνυμάτων εξαγωγής και το διαβιβάζει στον SQL Server Mobile Client Agent στη συσκευή. Ο SQL Server Mobile Client Agent εφαρμόζει τις αλλαγές από το αρχείο μηνυμάτων εξαγωγής στο Subscription της βάσης δεδομένων του SQL Server Mobile.

Ύστερα από αυτά ο SQL Server Mobile Client Agent έχει ενσωματώσει όλες τις αλλαγές στο Subscription της βάσης δεδομένων στην συσκευή και όποιες συγκρούσεις έχουν επιλυθεί, το Publication και το Subscription της βάσης δεδομένων είναι συγχρονισμένα, με τα δεδομένα να συγκλίνουν. Εντούτοις, επειδή οι αναπροσαρμογές μπορεί να εμφανίζονται συνεχώς, οι τιμές των δεδομένων μπορεί να μην είναι ίδιες στον Publisher και στους Subscribers [17].

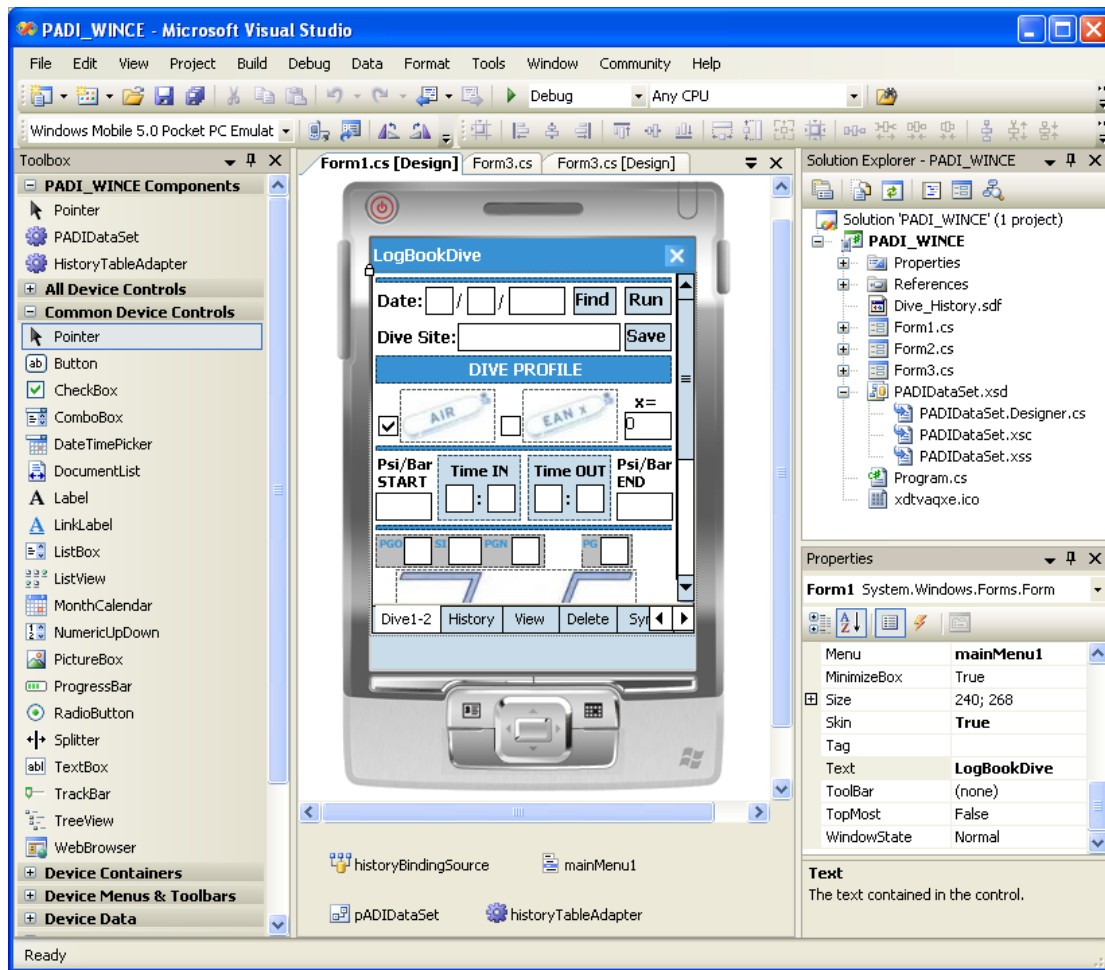
1.10 Εργαλεία Ανάπτυξης

1.10.1 Visual Studio 2005

Το Visual Studio της Microsoft είναι ένα προηγμένο περιβάλλον ανάπτυξης εφαρμογών για διάφορες πλατφόρμες. Έτσι λοιπόν δίνει στους προγραμματιστές την δυνατότητα να δημιουργήσουν προγράμματα, Web Sites, Web εφαρμογές, και Web υπηρεσίες που τρέχουν σε Microsoft Windows, Pocket PC, Smartphones, και στον παγκόσμιο ιστό [18]. Το Visual Studio υποστηρίζει αρκετές γλώσσες προγραμματισμού όπως είναι:

- Visual Basic
- Visual C++
- Visual C#
- Visual J#

Στη συγκεκριμένη διπλωματική εργασία θα ασχοληθούμε κυρίως με την Visual C# και τη χρήση του .NET Compact Framework 2.0 που απευθύνονται για εφαρμογές που δουλεύουν στο λειτουργικό σύστημα Windows Mobile 5.0. Μια χαρακτηριστική εικόνα του Visual Studio 2005 φαίνεται στο σχήμα 1.7.



Σχήμα 1.7: Visual Studio 2005.

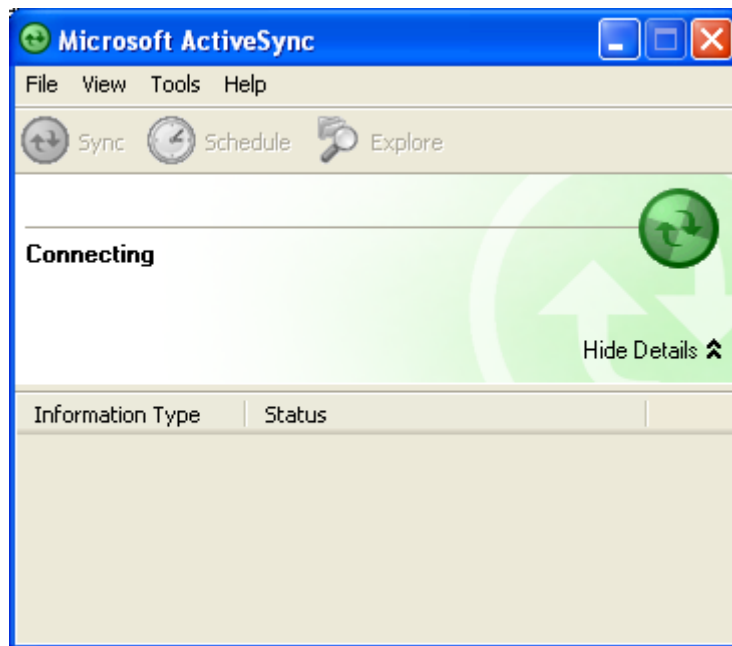
1.10.2 .NET Compact Framework 2.0

Το .NET Compact Framework είναι μια έκδοση του .NET Framework που σχεδιάστηκε για να τρέχει σε φορητές συσκευές όπως PDAs και κινητά τηλέφωνα. Το .NET Compact Framework χρησιμοποιεί μερικές ίδιες βιβλιοθήκες από το .NET Framework και επίσης μερικές βιβλιοθήκες που σχεδιάστηκαν συγκεκριμένα για φορητές συσκευές. Η ανάπτυξη εφαρμογών με χρήση του .NET Compact Framework μπορεί να γίνει στο Visual Studio 2003 ή στο Visual Studio 2005, με χρήση της γλώσσας C# ή της Visual Basic .NET. Οι εφαρμογές που προκύπτουν μπορούν να τρέξουν σε φορητές συσκευές με τη χρήση του μεταγλωττιστή υψηλής απόδοσης JIT.

Για να είναι σε θέση να τρέχουν οι εφαρμογές που χρησιμοποιούν το .NET Compact Framework, η πλατφόρμα θα πρέπει να υποστηρίζει το .NET Compact Framework. Μερικά λειτουργικά συστήματα όπως τα Windows CE 4.1, Microsoft Pocket PC, Microsoft Pocket PC - Smartphone 2003 και Windows Mobile 5.0 Pocket PC - Smartphone περιλαμβάνουν το .NET Compact Framework και μπορούν να τρέξουν τέτοιες εφαρμογές. Τέλος, αυτές οι εφαρμογές μπορούν να τρέξουν ακόμη και σε υπολογιστές γραφείου με το .NET Framework, επειδή τα εκτελέσιμα αρχεία τους παρουσιάζουν δυαδική (Binary) συμβατότητα [19].

1.10.3 ActiveSync

Το ActiveSync (Σχήμα 1.8) είναι ένα πρόγραμμα συγχρονισμού που αναπτύχθηκε από την Microsoft. Ο σκοπός του προγράμματος είναι να συγχρονίζει αρχεία, favourites, προσωπικά δεδομένα πληροφοριών κ.τ.λ. μεταξύ ενός Windows PC και ενός PDA που τρέχει Windows Mobile. Επίσης το ActiveSync είναι διαθέσιμο και για λειτουργικό σύστημα Symbian. Τέλος, ο συγχρονισμός αυτός γίνεται με τη σύνδεση της φορητής συσκευής στον υπολογιστή μέσω υπέρυθρων, Bluetooth και σειριακής ή USB θύρας [20].



Σχήμα 1.8: ActiveSync.

1.10.4 Windows Mobile 5.0 SDK για Pocket PC

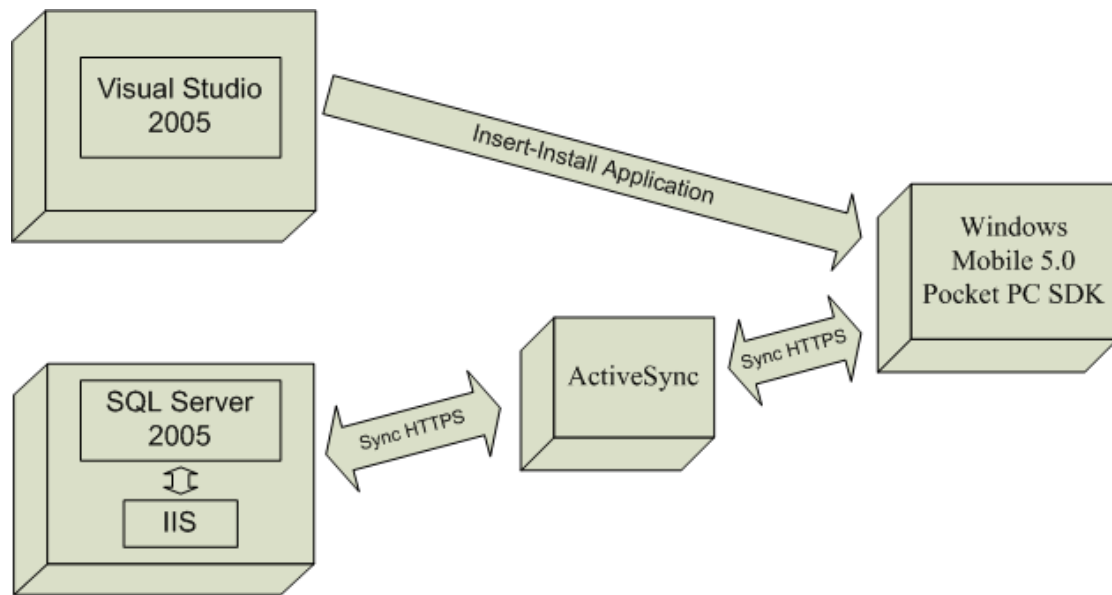
Τα Windows Mobile 5.0 SDK (Software Development Kit) για Pocket PC είναι ουσιαστικά ένας εξομοιωτής (Emulator) του Pocket PC που περιέχει το λειτουργικό σύστημα Windows Mobile 5.0. Σε αυτόν τον εξομοιωτή, μπορούμε να τρέξουμε ό,τι εφαρμογές έχουμε κατασκευάσει και θέλουμε να τις δοκιμάσουμε, χωρίς να απαιτείται η ύπαρξη κάποιου πραγματικού (Hardware) Pocket PC. Μια ενδεικτική εικόνα ενός τέτοιου εξομοιωτή φαίνεται στο σχήμα 1.9.



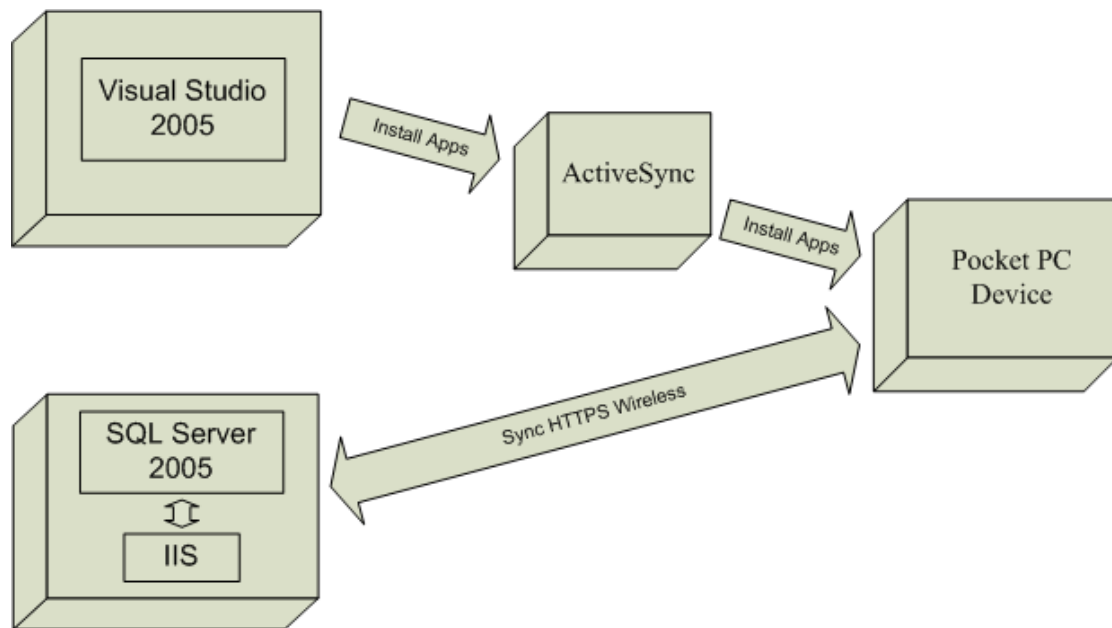
Σχήμα 1.9: Windows Mobile 5.0 SDK για Pocket PC.

1.11 Λογική σχηματική απεικόνιση των εργαλείων ανάπτυξης

Στην ενότητα αυτή θα αναφερθούμε στο πως συνεργάζονται τα διάφορα εργαλεία ανάπτυξης, ανάλογα με το είδος (εξομοιωτής ή πραγματική συσκευή) της συσκευής που χρησιμοποιούμε. Στην περίπτωση που έχουμε εξομοιωτή (Emulator), η λογική σχηματική απεικόνιση των εργαλείων ανάπτυξης φαίνεται στο σχήμα 1.10. Σε αυτό το σχήμα βλέπουμε ότι από την στιγμή που η εφαρμογή έχει κατασκευαστεί στο Visual Studio περνάει απευθείας και εγκαθίσταται στον εξομοιωτή, χωρίς την μεσολάβηση κάποιου άλλου εργαλείου. Ύστερα από αυτό μπορεί να γίνει πλέον η επικοινωνία (συγχρονισμός) της εφαρμογής με την κεντρική βάση δεδομένων (μέσω του IIS). Για την επίτευξη αυτής της αμφίδρομης επικοινωνίας μεσολαβεί το ActiveSync, το οποίο διασυνδέει τα δύο αυτά εργαλεία. Ενώ στην περίπτωση που έχουμε μία πραγματική συσκευή, η λογική σχηματική απεικόνιση των εργαλείων ανάπτυξης φαίνεται στο σχήμα 1.11. Σε αυτό το σχήμα βλέπουμε ότι από την στιγμή που η εφαρμογή μας έχει κατασκευαστεί στο Visual Studio, μπορούμε να την εγκαταστήσουμε πλέον μέσω όμως του ActiveSync στην συσκευή μας. Ύστερα από αυτό μπορεί να γίνει πλέον η επικοινωνία (συγχρονισμός) της εφαρμογής με την κεντρική βάση δεδομένων (μέσω του IIS). Η επικοινωνία αυτή μπορεί να γίνει μέσω κάποιου ασύρματου δικτύου (όπως Wireless LAN, Bluetooth, κ.τ.λ.), χωρίς να χρειάζεται η μεσολάβηση κάποιου επιπλέον εργαλείου.



Σχήμα 1.10: Λογική σχηματική απεικόνιση με χρήση Emulator.



Σχήμα 1.11: Λογική σχηματική απεικόνιση με χρήση αληθινής συσκευής.

2. Ασφάλεια Δεδομένων

Εισαγωγή

Στον παρόν κεφάλαιο θα γίνει παρουσίαση όλων εκείνων των εννοιών και αλγορίθμων της κρυπτογραφίας που απαιτούνται να γνωρίζει κάποιος για να μπορέσει να κατανοήσει τα κεφάλαια των εφαρμογών που ακολουθούν.

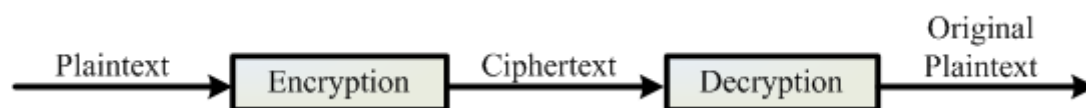
2.1 Βασικές έννοιες της κρυπτογραφίας

2.1.1 Χρήση της κρυπτογραφίας στην ανταλλαγή δεδομένων

Η ανάγκη για την αποστολή ενός μηνύματος από κάποιον αποστολέα σε έναν παραλήπτη, χωρίς να υπάρχει κίνδυνος να διαβαστεί από κάποιον άλλον, δηλαδή αυτό το μήνυμα να αποσταλεί με ασφάλεια, αποτελεί αντικείμενο της κρυπτογραφίας [21].

2.1.2 Βασική ιδέα της κρυπτογραφίας

Το αρχικό αυτό μήνυμα είναι το plaintext (ή cleartext, δηλ. καθαρό κείμενο). Η διαδικασία με την οποία στο μήνυμα αυτό αποκρύβεται η ουσία του περιεχόμενου του ονομάζεται κρυπτογράφηση (encryption). Το κρυπτογραφημένο μήνυμα είναι το ciphertext. Η διαδικασία της αλλαγής του ciphertext πίσω στο plaintext ονομάζεται αποκρυπτογράφηση. (Σημείωση: Σύμφωνα με το πρότυπο ISO 7498-2, χρησιμοποιούνται οι όροι "encipher" και "decipher" αντί για τους όρους "encrypt" και "decrypt", αντίστοιχα.) Όλα αυτά παρουσιάζονται στο σχήμα 2.1.



Σχήμα 2.1: Κρυπτογράφηση και Αποκρυπτογράφηση.

Η επιστήμη που έχει ως αντικείμενο την ασφάλεια των μηνυμάτων είναι η κρυπτογραφία (cryptography), και πραγματοποιείται από τους cryptographers (κρυπτογράφους). Οι Cryptanalysts ασχολούνται με την κρυπτανάλυση (cryptanalysis), το σπάσιμο του ciphertext, δηλαδή να βλέπουν μέσα από τα κρυπτογραφημένα δεδομένα. Ο κλάδος της επιστήμης που ασχολείται με την κρυπτογραφία και την κρυπτανάλυση, είναι η κρυπτολογία (cryptology) και αυτοί που εξασκούν το επάγγελμα αυτό είναι οι cryptologists. Οι σύγχρονοι cryptologists εκπαιδεύονται γενικά σε θεωρητικό μαθηματικό επίπεδο [21].

Το plaintext δηλώνεται με το γράμμα M, από το μήνυμα, ή με το γράμμα P, από το plaintext. Το plaintext μπορεί να είναι μία ροή (stream) από bits, ένα αρχείο κειμένου, ένα αρχείο εικόνας, μία ροή ψηφιακής φωνής, μια ψηφιακή βίντεο εικόνα

κ.τ.λ.. Όσον αφορά σε έναν υπολογιστή, το M είναι απλά δυαδικά δεδομένα. Το plaintext μπορεί να αποτελεί είτε τη μεταφορά είτε την αποθήκευση δεδομένων. Ουσιαστικά δηλαδή το M είναι το μήνυμα που κρυπτογραφείται.

Ας υποθέσουμε τώρα ότι το ciphertext δηλώνεται με το γράμμα C . Είναι επίσης δυαδικά δεδομένα, μερικές φορές του ίδιου μεγέθους με το M και κάποιες άλλες μπορεί και μεγαλύτερου. (Η κρυπτογράφηση όμως σε συνδυασμό με τη συμπίεση, μπορεί να οδηγήσει ώστε το C να είναι μικρότερο από το M , χωρίς όμως η κρυπτογράφηση να μπορεί να εκτελέσει αυτό απευθείας.) Η συνάρτηση κρυπτογράφησης E , επιδρά πάνω στο M για να παράγει το C . Η μαθηματική αυτή σχέση είναι:

$$E(M) = C$$

Στην αντίστροφη διαδικασία, η συνάρτηση αποκρυπτογράφησης D επιδρά πάνω στο C για να παραγάγει το M :

$$D(C) = M$$

Δεδομένου ότι η ουσία της κρυπτογράφησης και έπειτα της αποκρυπτογράφησης ενός μηνύματος είναι να ανακτηθεί το αρχικό plaintext, η ακόλουθη σχέση αποδεικνύει αυτό:

$$D(E(M)) = M$$

2.1.3 Οι έννοιες Authentication, Integrity και Nonrepudiation

Εκτός από την παροχή της εμπιστευτικότητας, το σύστημα της κρυπτογραφίας καλείται συχνά να παρέχει και άλλες εργασίες όπως [21]:

- **Authentication (Αυθεντικοποίηση).** Πρέπει να είναι σε θέση ο δέκτης να εξακριβώνει εάν το μήνυμα ανήκει πράγματι στον αποστολέα ή ότι ο αποστολέας είναι αυτός που ισχυρίζεται ότι είναι.
- **Integrity (Ακεραιότητα).** Πρέπει να είναι σε θέση ο δέκτης να ελέγχει εάν το μήνυμα δεν έχει τροποποιηθεί κατά τη μεταφορά, ώστε να μην μπορεί κάποιος εισβολέας να αντικαταστήσει το μήνυμα με ένα ψεύτικο και αυτό να φαίνεται ως αληθινό.
- **Nonrepudiation.** Ένας αποστολέας δεν μπορεί αργότερα να αρνηθεί ψευδώς ότι έστειλε το μήνυμα.

Αυτές οι έννοιες είναι ζωτικής σημασίας για την κοινωνική αλληλεπίδραση με την χρήση υπολογιστών, και είναι ανάλογες με τις διαπροσωπικές αλληλεπιδράσεις. Οι έννοιες αυτές παρέχουν μερικές από τις παρακάτω λειτουργίες:

- Ότι κάποιος είναι αυτός που λέει ότι είναι (Αυθεντικοποίηση).
- Ότι η άδεια ενός οδηγού, το ιατρικό δίπλωμα, και το διαβατήριο είναι έγκυρα (Υπογραφή).
- Ότι ένα έγγραφο είναι απόλυτα βέβαιο ότι έχει προέλθει από κάποιο πρόσωπο (Nonrepudiation).

- Ότι ένα έγγραφο δεν έχει αλλοιωθεί ή τροποποιηθεί (Ακεραιότητα).

2.1.4 Αλγόριθμοι και κλειδιά της κρυπτογραφίας

Ένας κρυπτογραφικός αλγόριθμος (αποκαλείται και ως cipher), είναι η μαθηματική συνάρτηση που χρησιμοποιείται για την κρυπτογράφηση και την αποκρυπτογράφηση [21]. Γενικά, υπάρχουν δύο σχετικές συναρτήσεις: μια για την κρυπτογράφηση και μια άλλη για την αποκρυπτογράφηση.

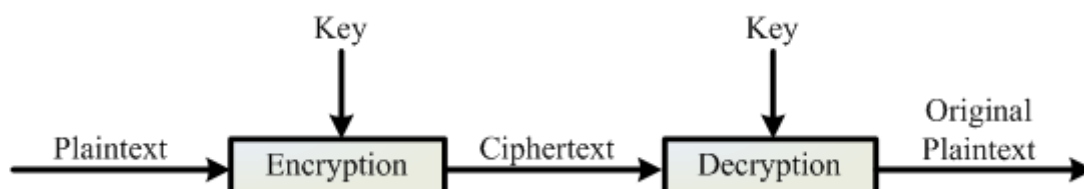
Εάν η ασφάλεια ενός αλγορίθμου είναι βασισμένη στο μυστικό τρόπο λειτουργίας του αλγορίθμου, τότε ένας τέτοιος αλγόριθμος είναι περιορισμένος (restricted). Τέτοιοι περιορισμένοι αλγόριθμοι έχουν μόνο ιστορικό ενδιαφέρον και θεωρούνται ανεπαρκείς για τα σημερινά πρότυπα. Μια μεγάλη ή μεταβαλλόμενη ομάδα χρηστών δεν μπορεί να τους χρησιμοποιήσει, επειδή κάθε φορά που ένας χρήστης αλλάζει ομάδα θα πρέπει να χρησιμοποιήσει και έναν διαφορετικό αλγόριθμο. Επίσης εάν κάποιος αποκαλύψει τυχαία το μυστικό, τότε ο καθένας θα πρέπει να αλλάξει τον αλγόριθμό.

Επιπλέον, οι περιορισμένοι αυτοί αλγόριθμοι δεν επιτρέπουν κανέναν ποιοτικό έλεγχο ή τυποποίηση. Κάθε ομάδα χρηστών πρέπει να έχει έναν μοναδικό αλγόριθμο. Μια τέτοια ομάδα δεν μπορεί να χρησιμοποιήσει έτοιμα προϊόντα υλικού ή λογισμικού, επειδή οποιοσδήποτε θα μπορούσε να αγοράσει το ίδιο προϊόν και να μάθει τον αλγόριθμο, οπότε θα πρέπει να γράψουν οι ίδιοι τους αλγορίθμους και τις εφαρμογές τους. Εάν κανένας στην ομάδα δεν είναι καλός cryptographer, δεν θα μπορούν να ξέρουν εάν ο αλγόριθμος τους είναι ασφαλής.

Παρόλα αυτά τα σημαντικά μειονεκτήματα, οι περιορισμένοι αλγόριθμοι είναι πάρα πολύ δημοφιλείς για εφαρμογές χαμηλής ασφάλειας. Στις περιπτώσεις αυτές οι χρήστες είτε δεν αντιλαμβάνονται είτε δεν ενδιαφέρονται για τα προβλήματα ασφάλειας που υπάρχουν στο σύστημά τους.

Τα σύγχρονα συστήματα κρυπτογραφίας λύνουν αυτό το πρόβλημα με τη χρήση ενός κλειδιού, που δηλώνεται εάν υποθέσουμε με το γράμμα K . Αυτό το κλειδί να είναι οποιοδήποτε από έναν μεγάλο αριθμό τιμών. Η περιοχή των πιθανών τιμών του κλειδιού καλείται keyspace. Οπότε τώρα οι διαδικασίες της κρυπτογράφησης και της αποκρυπτογράφησης χρησιμοποιούν αυτό το κλειδί (δηλ., εξαρτώνται από αυτό το κλειδί και αυτό το γεγονός δηλώνεται από τον δείκτη K), έτσι οι συναρτήσεις τώρα γίνονται [21] (δείτε το σχήμα 2.2):

$$\begin{aligned}E_K(M) &= C \\D_K(C) &= M \\D_K(E_K(M)) &= M\end{aligned}$$

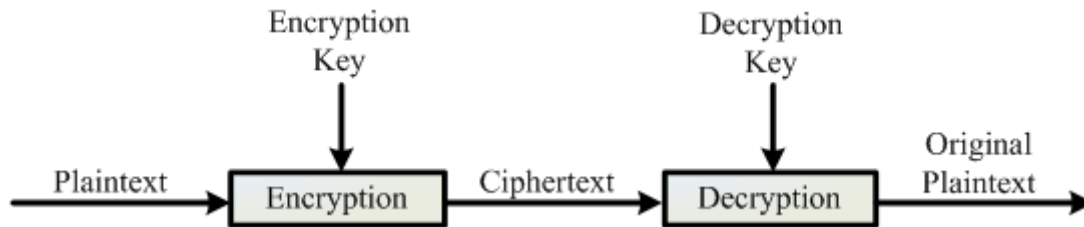


Σχήμα 2.2: Κρυπτογράφηση και αποκρυπτογράφηση με ένα κλειδί.

Μερικοί αλγόριθμοι χρησιμοποιούν διαφορετικά κλειδιά στην κρυπτογράφηση και στην αποκρυπτογράφηση (δείτε το σχήμα 2.3). Δηλαδή το κλειδί

κρυπτογράφησης, K_1 , είναι διαφορετικό από το αντίστοιχο κλειδί αποκρυπτογράφησης, K_2 . Σε αυτήν την περίπτωση οι συναρτήσεις γίνονται:

$$\begin{aligned}E_{K_1}(M) &= C \\D_{K_2}(C) &= M \\D_{K_2}(E_{K_1}(M)) &= M\end{aligned}$$



Σχήμα 2.3: Κρυπτογράφηση και αποκρυπτογράφηση με δύο διαφορετικά κλειδιά.

Όλη η ασφάλεια σε αυτούς τους αλγόριθμους είναι βασισμένη στο κλειδί (ή τα κλειδιά) και όχι στις λεπτομέρειες του αλγορίθμου. Αυτό σημαίνει ότι ο αλγόριθμος μπορεί να δημοσιευθεί και να αναλυθεί, οπότε και τα προϊόντα που χρησιμοποιούν τον αλγόριθμο μπορούν να παραχθούν μαζικά. Επίσης, δεν πειράζει εάν κάποιος ξέρει τον αλγόριθμό σας, επειδή δεν ξέρει το ιδιαίτερο κλειδί σας, δεν μπορεί να διαβάσει τα μηνύματά σας. Έτσι λοιπόν, ένα σύγχρονο κρυπτογραφικό σύστημα αποτελείται πλέον από τον αλγόριθμο, όλα τα πιθανά plaintexts, τα ciphertexts και τα κλειδιά [21].

2.1.5 Συμμετρικοί Αλγόριθμοι

Υπάρχουν δύο γενικοί τύποι αλγορίθμων βασισμένων σε κλειδιά: ο συμμετρικός αλγόριθμος και ο αλγόριθμος δημόσιου-κλειδιού (public-key). Οι συμμετρικοί αλγόριθμοι, οι οποίοι αποκαλούνται μερικές φορές και συμβατικοί αλγόριθμοι, είναι αλγόριθμοι όπου το κλειδί κρυπτογράφησης μπορεί να υπολογιστεί από το κλειδί αποκρυπτογράφησης και αντίστροφα. Στους περισσότερους συμμετρικούς αλγορίθμους το κλειδί κρυπτογράφησης και το κλειδί αποκρυπτογράφησης είναι τα ίδια. Αυτοί οι αλγόριθμοι αποκαλούνται επίσης και αλγόριθμοι μοναδικού-κλειδιού και απαιτούν ο αποστολέας και ο δέκτης να συμφωνούν σχετικά με ένα κλειδί προτού να μπορέσουν να επικοινωνήσουν με ασφάλεια. Η ασφάλεια ενός συμμετρικού αλγορίθμου στηρίζεται στο κλειδί, χωρίς να υπάρχει λόγος απόκρυψης των βασικών σημείων κρυπτογράφησης και αποκρυπτογράφησης. Εφόσον πρέπει να παραμείνει η επικοινωνία μυστική, το κλειδί πρέπει να παραμείνει μυστικό [21].

Η κρυπτογράφηση και η αποκρυπτογράφηση με έναν συμμετρικό αλγόριθμο δείχνονται από τις σχέσεις:

$$\begin{aligned}E_K(M) &= C \\D_K(C) &= M\end{aligned}$$

Οι συμμετρικοί αλγόριθμοι μπορούν να διαιρεθούν σε δύο κατηγορίες: 1) Σε αυτούς που το plaintext είναι ένα bit (ή μερικές φορές ένα byte) την κάθε στιγμή και ονομάζονται αλγόριθμοι ροής (stream) ή ροής (stream) ciphers. 2) Και σε αυτούς που

το plaintext είναι ομάδες από bits. Οι ομάδες των bits καλούνται blocks, και οι αλγόριθμοι αυτοί ονομάζονται αλγόριθμοι block ή blocks ciphers. Για τους σύγχρονους αλγορίθμους υπολογιστών, ένα χαρακτηριστικό μέγεθος blocks είναι 64 bits, αρκετά μεγάλο για να αποτρέψει την ανάλυση και αρκετά μικρό για να είναι εφαρμόσιμο.

2.1.6 Αλγόριθμοι δημόσιου-κλειδιού

Οι αλγόριθμοι δημόσιου-κλειδιού (public-key) (επίσης αποκαλούνται και σαν ασύμμετροι αλγόριθμοι) σχεδιάστηκαν έτσι ώστε το κλειδί που χρησιμοποιείται για την κρυπτογράφηση είναι διαφορετικό από το κλειδί που χρησιμοποιείται για την αποκρυπτογράφηση. Ακόμη, το κλειδί αποκρυπτογράφησης δεν μπορεί (τουλάχιστον σε λογικό χρονικό διάστημα) να υπολογιστεί από το κλειδί κρυπτογράφησης. Οι αλγόριθμοι καλούνται "public-key" επειδή το κλειδί κρυπτογράφησης μπορεί να δημοσιοποιηθεί: Ένας άγνωστος μπορεί να χρησιμοποιήσει το κλειδί κρυπτογράφησης για να κρυπτογραφήσει ένα μήνυμα, αλλά μόνο ένα συγκεκριμένο πρόσωπο με το αντίστοιχο κλειδί αποκρυπτογράφησης μπορεί να αποκρυπτογραφήσει το μήνυμα. Σε αυτά τα συστήματα, το κλειδί κρυπτογράφησης καλείται συχνά δημόσιο κλειδί, και το κλειδί αποκρυπτογράφησης καλείται συχνά ιδιωτικό κλειδί. Το ιδιωτικό κλειδί μερικές φορές επίσης καλείται μυστικό (secret) κλειδί, αλλά για να αποφευχθεί η σύγχυση με τους συμμετρικούς αλγόριθμους, αυτή η φράση γενικά δεν χρησιμοποιείται [21]. Η κρυπτογράφηση που χρησιμοποιεί το δημόσιο κλειδί K μπορεί να περιγραφεί με την ακόλουθη συνάρτηση:

$$E_K(M) = C$$

Η αποκρυπτογράφηση με το ιδιωτικό κλειδί περιγράφεται με τη συνάρτηση:

$$D_K(C) = M$$

Μερικές φορές, τα μηνύματα κρυπτογραφούνται με το ιδιωτικό κλειδί και αποκρυπτογραφούνται με το δημόσιο κλειδί. Συγκεκριμένα, αυτό χρησιμοποιείται στις ψηφιακές υπογραφές. Παρά την πιθανή σύγχυση, αυτές οι διαδικασίες περιγράφονται με τις συναρτήσεις:

$$\begin{aligned} E_K(M) &= C \\ D_K(C) &= M \end{aligned}$$

2.2 Αλγόριθμος κρυπτογραφίας AES

2.2.1 Εισαγωγή

Ο AES (Advanced Encryption Standard) είναι ένας block cipher συμμετρικός αλγόριθμος κρυπτογραφίας, που υιοθετήθηκε από το Εθνικό Ίδρυμα Προτύπων και Τεχνολογίας (NIST = National Institute of Standards and Technology) τον Νοέμβριο του 2001 ύστερα από μία πενταετή διαδικασία τυποποίησης. Ο AES είναι επίσης

γνωστός και σαν Rijndael, λόγω του γεγονότος ότι ο AES είναι ουσιαστικά ο αλγόριθμος κρυπτογραφίας Rijndael, με μικρές μόνο τροποποιήσεις [22]. Πιο συγκεκριμένα, η βασική τους διαφορά είναι ότι ο AES έχει ένα σταθερό μέγεθος block των 128 bits και ένα μέγεθος κλειδιού των 128, 192 ή 256 bits, ενώ Rijndael μπορεί να χρησιμοποιήσει διάφορα μεγέθη κλειδιών και blocks σε οποιοδήποτε πολλαπλάσιο των 32 bits, με ελάχιστο τα 128 bits και μέγιστο τα 256 bits. Η επιλογή του Rijndael ως AES έγινε κυρίως λόγω της ευκολίας εφαρμογής του σε υλικό (hardware) και της ισχυρής απόδοσής του σχεδόν σε όλες τις πλατφόρμες. Στον πίνακα 2.1 συνοψίζονται οι τρεις επίσημες εκδόσεις του AES. Το N_p αναφέρεται στο μήκος των blocks (σε αριθμό 32-bit λέξεων), το N_k στο μήκος των κλειδιών (σε αριθμό 32-bit λέξεων), και το N_r στον αριθμό των κύκλων (rounds) [23].

Πίνακας 2.1: Οι τρεις επίσημες εκδόσεις του AES.

	N_p	N_k	N_r
AES-128	4	4	10
AES-192	4	6	12
AES-256	4	8	14

2.2.2 Αλγόριθμος κρυπτογράφησης AES

Οι μετασχηματισμοί που χρησιμοποιούνται στον αλγόριθμο κρυπτογράφησης AES καλούνται ShiftRows(), SubBytes(), MixColumns() και AddRoundKey(). Στην αρχή του αλγόριθμου κρυπτογράφησης AES αντιγράφονται 16 bytes (128 bits $\rightarrow$ 1 block) σε έναν 4x4 πίνακα S που ονομάζεται State. Μετά από μια αρχική εφαρμογή του μετασχηματισμού AddRoundKey(), ο State μετασχηματίζεται σε $N_r = 10, 12$ ή 14 κύκλους (ανάλογα με το μήκος του κλειδιού που χρησιμοποιείται), με τον τελικό κύκλο να διαφέρει ελαφρώς από τους προηγούμενους $N_r - 1$ κύκλους (δηλ., ο τελικός κύκλος δεν περιλαμβάνει τον μετασχηματισμό MixColumns()). Το περιεχόμενο του State που λαμβάνεται τελικά, αντιπροσωπεύει την κρυπτογράφιση του AES [22].

Ο κάθε κύκλος αποτελείται από τους ακόλουθους τέσσερις μετασχηματισμούς:

1. Τα bytes του State αντικαθίστανται σύμφωνα με έναν δεδομένο πίνακα αντικατάστασης (αυτός ο μετασχηματισμός καλείται SubBytes() σύμφωνα με τις προδιαγραφές AES).
2. Οι σειρές του State μετατοπίζονται αριστερά από τα διαφορετικά offset (αυτός ο μετασχηματισμός καλείται ShiftRows() σύμφωνα με τις προδιαγραφές AES).
3. Τα δεδομένα μέσα σε κάθε στήλη του State ανακατεύονται (αυτός ο μετασχηματισμός καλείται MixColumns() σύμφωνα με τις προδιαγραφές AES).
4. Το κλειδί του κύκλου προστίθεται στο State (αυτός ο μετασχηματισμός καλείται AddRoundKey() σύμφωνα με τις προδιαγραφές AES).

2.2.3 Αλγόριθμος αποκρυπτογράφησης AES

Οι μετασχηματισμοί που χρησιμοποιούνται από τον αλγόριθμο κρυπτογράφησης AES μπορούν να αναστραφούν και να εφαρμοστούν σε αντίστροφη διάταξη για να παραχθεί ο αλγόριθμος αποκρυπτογράφησης AES. Οι μετασχηματισμοί που χρησιμοποιούνται στον αλγόριθμο αποκρυπτογράφησης AES καλούνται `InvShiftRows()`, `InvSubBytes()`, `InvMixColumns()` και `AddRoundKey()`. Ο μετασχηματισμός `AddRoundKey()` είναι ο αντίστροφος της κρυπτογράφησης, με την προσθήκη μόνο του modulo 2. Επίσης, οι αντίστροφοι μετασχηματισμοί `InvSubBytes()` και `InvShiftRows()` εναλλάσσουν θέσεις, σε σύγκριση με τους μετασχηματισμούς `SubBytes()` και `ShiftRows()` της κρυπτογράφησης [23].

2.2.4 Ασφάλεια AES

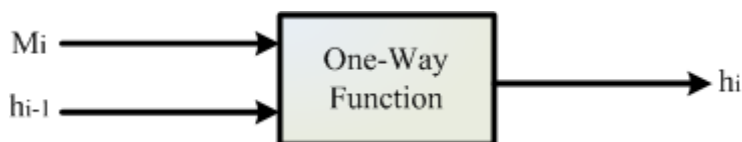
Ο AES όσο αφορά την ασφάλεια του, παρουσιάζει τόσο θετικά όσο και αρνητικά σημεία. Τα θετικά είναι ότι το σύστημα κρυπτογράφησης σχεδιάστηκε σε χαμηλό επίπεδο για να παρουσιάζει υψηλή πολυπλοκότητα απέναντι στη διαφορική κρυπτολογική ανάλυση, στη γραμμική κρυπτολογική ανάλυση, και μερικών άλλων σχετικών επιθέσεων που μπορούν να χρησιμοποιηθούν. Συνεπώς, το σύστημα κρυπτογράφησης AES είναι ανθεκτικό ενάντια σε αυτές τις επιθέσεις. Τα αρνητικά είναι ότι πολλές νέες κρυπτολογικές τεχνικές έχουν αναπτυχθεί (και θα συνεχίσουν πιθανότατα να αναπτύσσονται) για να σπάσουν το συμμετρικό σύστημα κρυπτογράφησης AES. Στην πραγματικότητα, ο AES είναι ένας ενεργός τομέας μελέτης, και θα ήταν ενδιαφέρον να μάθουμε για πιθανές μελλοντικές εκδόσεις του. Από μία άλλη άποψη οι Cryptanalysts, θεωρούν το AES ελκυστικό, επειδή θα είναι ευρέως χρησιμοποιούμενο στο μέλλον και επειδή το συμμετρικό σύστημα κρυπτογράφησης έχει μια απλή μαθηματική δομή [23].

2.3 Αλγόριθμοι κρυπτογραφίας Hash και SHA

2.3.1 Μονόδρομες Hash συναρτήσεις

Η Hash συνάρτηση είναι μια συνάρτηση που δέχεται σαν είσοδο δεδομένα αυθαίρετου μήκους και τα μετατρέπει σε μονόδρομα δεδομένα (χωρίς να υπάρχει δυνατότητα επαναφοράς στην αρχική τους μορφή). Στην πραγματικότητα, οι μονόδρομες Hash συναρτήσεις στηρίζονται στην ιδέα μιας συνάρτησης συμπίεσης. Αποτελέσματα αυτής της μονόδρομης συνάρτησης είναι μια μεταβλητή hash με μήκος n , για δεδομένη είσοδο μεγαλύτερου μήκους m . Οι είσοδοι της συνάρτησης συμπίεσης είναι ένα block μηνύματος και η έξοδος του προηγούμενου block από το κείμενο [21] (Σχήμα 2.4). Η έξοδος της συνάρτησης είναι το hash όλων των blocks μέχρι εκείνο το σημείο. Δηλαδή το hash του block M_i θα είναι:

$$h_i = f(M_i, h_{i-1})$$



Σχήμα 2.4: Μονόδρομη Hash συνάρτηση.

Αυτή η hash μεταβλητή, μαζί με το επόμενο block μήνυμα, γίνεται η επόμενη είσοδος της συνάρτησης συμπίεσης. Το Hash ολόκληρου του μηνύματος είναι το hash του τελευταίου block.

Η προ-εικόνα (δηλ., πριν ξεκινήσει ακόμη η διαδικασία) πρέπει να περιέχει κάποια δυαδική αντιπροσώπευση του μήκους ολόκληρου του μηνύματος. Αυτή η τεχνική ξεπερνά ένα πιθανό πρόβλημα ασφάλειας σε μηνύματα που έχουν ενδεχομένως διαφορετικά μήκη hashing για την ίδια μεταβλητή. Αυτή η τεχνική μερικές φορές καλείται και ως MD-strengthening.

Διάφοροι ερευνητές έχουν εξετάσει σε θεωρητικό επίπεδο ότι εάν η συνάρτηση συμπίεσης είναι ασφαλής, τότε και η μέθοδος hashing αυθαίρετου-μήκους με προ-εικόνα είναι επίσης ασφαλής, αλλά τίποτα δεν έχει αποδειχθεί [21].

2.3.2 SHA hash συναρτήσεις

Η οικογένεια του SHA (Secure Hash Algorithm = Ασφαλής Hash Αλγόριθμος) είναι ένα σύνολο σχετικών κρυπτογραφικών hash συναρτήσεων. Η συνηθέστερα χρησιμοποιούμενη συνάρτηση της οικογένειας αυτής, ο SHA-1, υιοθετείται σε μια μεγάλη ποικιλία δημοφιλών εφαρμογών και πρωτοκόλλων ασφάλειας, συμπεριλαμβανομένου του TLS, SSL, PGP, SSH, S/MIME, και IPSec. Ο SHA-1 θεωρείται ο διάδοχος MD5, μια προηγούμενη, ευρέως χρησιμοποιούμενη hash συνάρτηση. Σε μερικές περιπτώσεις, με ιδιαίτερες απαιτήσεις ασφάλειας, προτείνεται η χρήση του SHA-256 ή μεγαλύτερου. Οι αλγόριθμοι SHA σχεδιάστηκαν από την NSA (National Security Agency) και δημοσιεύθηκαν ως πρότυπα της αμερικανικής κυβέρνησης.

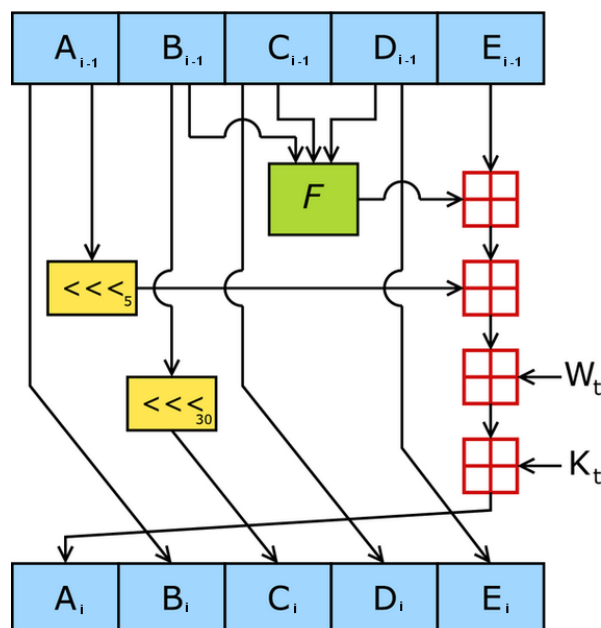
Το πρώτο μέλος της οικογένειας, δημοσιεύτηκε το 1993 και ονομάζεται επίσημα SHA. Εντούτοις, καλείται συχνά ως SHA-0 για να αποφευχθεί η σύγχυση με τους διαδόχους του. Δύο έτη αργότερα δημοσιεύθηκε ο SHA-1, που είναι ο πρώτος διάδοχος του SHA. Από τότε, τέσσερις παραλλαγές του SHA έχουν δημοσιευτεί για να καλύψουν τις αυξανόμενες ανάγκες ασφαλείας: SHA-224, SHA-256, SHA-384, και SHA-512 (μερικές φορές συλλογικά αναφέρονται ως SHA-2). Στο πίνακα 2.2 φαίνονται τα διάφορα είδη του SHA [24].

Πίνακας 2.2: Τα διάφορα μεγέθη του SHA.

Algorithm	Output size	Internal state size	Block size	Length size	Word size	Passes	Operations	Collision
SHA-0	160	160	512	64	32	80	+,and,or,xor,rotl	Yes
SHA-1	160	160	512	64	32	80	+,and,or,xor,rotl	With flaws
SHA-256/224	256/224	256	512	64	32	64	+,and,or,xor,shr,rotr	No
SHA-512/384	512/384	512	1024	128	64	80	+,and,or,xor,shr,rotr	No

2.3.3 SHA-1

Ο SHA-1 παράγει μια συγχώνευση 160-bits από ένα μήνυμα με μέγιστο μέγεθος 2^{64} bits, και είναι βασισμένο σε αρχές παρόμοιες με εκείνες που χρησιμοποιήθηκαν από τους αλγόριθμους MD4 και MD5. Το SHA-1 είναι ασφαλές επειδή έχει ως σκοπό να είναι υπολογιστικά ανέφικτο να ανακτηθεί ένα μήνυμα που αντιστοιχεί σε μια δεδομένη συγχώνευση μηνυμάτων, ή για να βρεθούν δύο διαφορετικά μηνύματα που να παράγουν την ίδια συγχώνευση μηνυμάτων. Οποιαδήποτε αλλαγή συμβεί σε ένα μήνυμα κατά τη μεταφορά, με μεγάλη πιθανότητα θα οδηγήσει σε μια διαφορετική συγχώνευση μηνυμάτων, και ο έλεγχος της υπογραφής θα αποτύχει. Στο σχήμα 2.5 φαίνεται μία επανάληψη της συνάρτησης συμπίεσης του SHA-1. Εκεί τα A, B, C, D και E είναι 32-bit λέξεις από το State, το F είναι μια μη γραμμική συνάρτηση, το $\lll_n$ δηλώνει μια αριστερή εναλλαγή bit σε n θέσεις, το + δείχνει την πρόσθεση modulo 2^{32} , το i-1 είναι η παρούσα κατάσταση, ενώ το i η επόμενη κατάσταση, και το K_t είναι μια σταθερά [24].



Σχήμα 2.5: Μία επανάληψη της συνάρτησης συμπίεσης του SHA-1.

2.3.4 Ασφάλεια του SHA

Οι επιθέσεις έχουν βρεθεί για τον SHA-0 και τον SHA-1. Καμία επίθεση δεν έχει αναφερθεί ακόμα για τις διάφορες παραλλαγές του SHA-2, αλλά δεδομένου ότι είναι παρόμοιοι με τον SHA-1 οι ερευνητές ανησυχούν. Για αυτό τον λόγο αναπτύσσουν νέα καλύτερα hashing πρότυπα του SHA, όπως φαίνεται άλλωστε και στη στήλη «Collision» του πίνακα 2.2 [24].

2.4 Αλγόριθμος κρυπτογραφίας RSA

2.4.1 Αλγόριθμος RSA

Ένα από τα πιο επιτυχημένα κρυπτογραφικά συστήματα είναι το κρυπτογραφικό σύστημα RSA, που επινοήθηκε το 1978 από τους Ronald Rivest, Adi Shamir και Leonard Adelman και πήρε το όνομά του από τα αρχικά των επιθέτων τους. Το RSA είναι ένα κρυπτογραφικό σύστημα με δημόσιο κλειδί και μέχρι σήμερα θεωρείται αδύνατο να σπάσει με τη χρήση σύγχρονων υπολογιστών. Σήμερα χρησιμοποιείται ευρύτατα κυρίως στις οικονομικές και στις τραπεζικές συναλλαγές [25]. Η βασική δομή του RSA είναι η εξής:

1. Επιλέγουμε δύο ακέραιους πρώτους αριθμούς τον p και q και υπολογίζουμε το γινόμενο τους $n = p \cdot q$.
2. Επιλέγουμε ένα τυχαίο αριθμό τον d ο οποίος είναι πρώτος ως προς τους $(p-1)$ και $(q-1)$, δηλαδή ο μέγιστος κοινός διαιρέτης των d , $(p-1)$ και $(q-1)$ είναι το ένα.
3. Υπολογίζουμε τον αριθμό e από τη σχέση: $e \cdot d \bmod (p-1)(q-1) = 1$. Δηλαδή, ο e είναι ο αντίστροφος του d , $\bmod (p-1)(q-1)$.
4. Το ζεύγος των αριθμών (e, n) είναι το δημόσιο κλειδί.
5. Το ζεύγος των αριθμών (d, n) είναι το ιδιωτικό κλειδί.

Το δημόσιο κλειδί στέλνεται στον αποστολέα του μηνύματος (π.χ. στον πελάτη της τράπεζας που θέλει να κάνει μία συναλλαγή μέσω του διαδικτύου). Το ιδιωτικό κλειδί το κρατά ο παραλήπτης του μηνύματος (π.χ. η τράπεζα). Ο αποστολέας κρυπτογραφεί το μήνυμα (π.χ. τα στοιχεία της συναλλαγής) με το δημόσιο κλειδί και στέλνει το μήνυμα στον παραλήπτη. Για να αποκρυπτογραφηθεί το μήνυμα, χρειάζεται και το δημόσιο και το ιδιωτικό κλειδί. Το ιδιωτικό κλειδί δε φεύγει ποτέ από τον παραλήπτη, οπότε δεν υπάρχει κίνδυνος υποκλοπής.

Μία καλή παραστατική ερμηνεία του κρυπτογραφικού συστήματος RSA είναι η εξής: ο παραλήπτης του μηνύματος (η τράπεζα) στέλνει στον αποστολέα (στον πελάτη) ένα κιβώτιο που κλειδώνει με ένα λουκέτο. Το λουκέτο είναι ανοιχτό και ο παραλήπτης κρατάει το κλειδί του. Ο αποστολέας παραλαμβάνει το κιβώτιο με το λουκέτο ανοιχτό, βάζει μέσα το μήνυμα του και κλειδώνει το λουκέτο. Τώρα δεν μπορεί κανείς, ούτε ο ίδιος ο αποστολέας, να το ανοίξει. Στη συνέχεια στέλνει το κλειδωμένο κιβώτιο στον παραλήπτη ο οποίος με το κλειδί ανοίγει το λουκέτο και βγάζει το μήνυμα από το κιβώτιο. Στο ερμηνευτικό αυτό μοντέλο το λουκέτο είναι το ζεύγος των αριθμών (e, n) , και το κλειδί του λουκέτου είναι το ζεύγος των αριθμών (d, n) [25].

2.4.2 Ασφάλεια του RSA

Για να καταλάβουμε καλύτερα την ασφάλεια που μας παρέχει το κρυπτογραφικό σύστημα RSA θα αναφέρουμε πρώτα πως μπορούμε να το σπάσουμε, δηλαδή τι πρέπει να κάνουμε, για να αποκρυπτογραφήσουμε ένα μήνυμα που κρυπτογραφήθηκε με το σύστημα RSA. Μπορούμε πολύ εύκολα να βρούμε το δημόσιο κλειδί, δηλαδή το ζεύγος των αριθμών (e, n) . Αφού τώρα γνωρίζουμε τον αριθμό n , δεν έχουμε παρά να τον αναλύσουμε σε γινόμενο δύο πρώτων αριθμών για να βρούμε τους αριθμούς p και q . Μόλις τους βρούμε, η αποκρυπτογράφηση γίνεται αμέσως, αφού η μέθοδος του συστήματος RSA είναι γνωστή [25].

Ενώ είναι πολύ εύκολο να πολλαπλασιάσουμε δύο πρώτους αριθμούς για να βρούμε το γινόμενό τους, είναι πάρα πολύ δύσκολο να αναλύσουμε έναν αριθμό σε γινόμενο δύο πρώτων αριθμών και είναι πρακτικά αδύνατον αν ο αριθμός έχει πολλά ψηφία.

Για να αποδείξουν ότι το κρυπτογραφικό τους σύστημα δεν μπορεί να σπάσει, οι Rivest, Shamir και Adelman ζήτησαν από όποιον νομίζει ότι μπορεί, να αναλύσει σε γινόμενο δύο πρώτων αριθμών έναν ακέραιο με 129 ψηφία. Μετά από 17 χρόνια ο αριθμός αναλύθηκε από ένα δίκτυο 1.600 υπολογιστών. Έτσι λοιπόν με τα σημερινά τεχνολογικά δεδομένα, το πρόβλημα της ανάλυσης ενός αριθμού σε γινόμενο δύο πρώτων αριθμών είναι αδύνατον να λυθεί με τη χρήση σύγχρονων υπολογιστών και πόσο μάλλον όταν ο αριθμός έχει πολλά ψηφία [25].

2.5 Πιστοποιητικό δημοσίου κλειδιού (Public key certificate)

Στην κρυπτογραφία, το πιστοποιητικό δημοσίου κλειδιού (ή πιστοποιητικό ταυτότητας) είναι ένα πιστοποιητικό που χρησιμοποιεί μια ψηφιακή υπογραφή για να δεσμεύσει μαζί ένα δημόσιο κλειδί με μια ταυτότητα (δηλ., πληροφορίες όπως το όνομα ενός προσώπου ή ενός οργανισμού, τη διεύθυνσή τους, κ.τ.λ.). Το πιστοποιητικό αυτό μπορεί να χρησιμοποιηθεί για να ελέγξει εάν ένα δημόσιο κλειδί ανήκει σε ένα άτομο [26].

Σε ένα τυπικό σχήμα δημοσίου κλειδιού υποδομής (PKI = Public Key Infrastructure), η υπογραφή θα είναι μιας αρχής πιστοποίησης (CA = Certificate Authority). Σε έναν σχήμα εμπιστοσύνης WEB, η υπογραφή είναι είτε του χρήστη (ένα self-signed certificate), είτε άλλων χρηστών ("επικυρώσεις"). Σε καθεμία περίπτωση, οι υπογραφές σε ένα πιστοποιητικό είναι επιβεβαιώσεις από τον υπογράφοντα πιστοποιητικών ότι οι πληροφορίες ταυτότητας και του δημοσίου κλειδιού ανήκουν στον ίδιο. Το πιο κοινό πρότυπο πιστοποιητικών είναι το ITU-T X.509.

Ένα πιστοποιητικό μπορεί να ανακληθεί εάν ανακαλυφθεί ότι το σχετικό ιδιωτικό κλειδί του έχει παραχωρηθεί, ή εάν η συγγένεια (μεταξύ μιας οντότητας και ενός δημόσιου κλειδιού) που ενσωματώνεται στο πιστοποιητικό ανακαλύπτεται ότι είναι ανακριβής ή έχει αλλάξει (π.χ., εάν ένα πρόσωπο αλλάζει τη δουλειά ή το όνομα). Εάν και η ανάκληση είναι ένα σπάνιο περιστατικό, ο χρήστης πρέπει πάντα να ελέγχει την ισχύ του πιστοποιητικού. Αυτό μπορεί να γίνει με τη σύγκριση του πιστοποιητικού με έναν κατάλογο ανάκλησης πιστοποιητικών (CRL = Certificate Revocation List), δηλαδή με ένα κατάλογο ανακλημένων ή ακυρωμένων πιστοποιητικών. Ένας άλλος τρόπος να ελεγχθεί η ισχύς των πιστοποιητικών είναι να ερωτηθεί η αρχή των πιστοποιητικών (CA) που χρησιμοποιεί το Online Certificate Status Protocol (OCSP) [26].

Τα βασικά χαρακτηριστικά από τα οποία αποτελείται ένα πιστοποιητικό είναι τα εξής:

- Το δημόσιο κλειδί που υπογράφεται.
- Ένα όνομα, το οποίο μπορεί να αναφέρεται σε ένα πρόσωπο, έναν υπολογιστή ή ενός οργανισμού.
- Μια περίοδος ισχύος.
- Την θέση (URL) ενός κέντρου ανάκλησης (revocation center).

2.6 Πρωτόκολλα ασφαλείας

2.6.1 TLS/SSL

Το SSL (Secure Sockets Layer) και ο διάδοχός του TLS (Transport Layer Security), είναι κρυπτογραφικά πρωτόκολλα που παρέχουν ασφαλή επικοινωνία στο διαδίκτυο για ειδικές περιπτώσεις όπως το ηλεκτρονικό ταχυδρομείο (e-mail), το internet fax, και για άλλες μεταφορές δεδομένων. Υπάρχουν μικρές διαφορές μεταξύ του SSL 3.0 και του TLS 1.0, αλλά το πρωτόκολλο παραμένει ουσιαστικά το ίδιο. Ο όρος "SSL" όπως χρησιμοποιείται εδώ ισχύει και για τα δύο πρωτόκολλα [27].

Το SSL παρέχει πιστοποίηση και απομονωμένη επικοινωνία μέσω του διαδικτύου χρησιμοποιώντας κρυπτογραφία. Σε μια τυπική χρήση του, μόνο ο server πιστοποιείται (δηλ. η ταυτότητά του κατοχυρώνεται) ενώ ο client παραμένει ανώνυμος. Η αμοιβαία πιστοποίηση απαιτείται και σχήμα δημοσίου κλειδιού υποδομής (PKI) στους clients. Τα πρωτόκολλα αυτά επιτρέπουν εφαρμογές επικοινωνίας client/server με τέτοιο τρόπο ώστε να αποτρέπεται η παρακολούθηση και η παραποίηση των μηνυμάτων.

Το SSL αποτελείται από τρεις βασικές φάσεις:

1. Διαπραγμάτευση για την υποστήριξη του αλγόριθμου.
2. Δημόσιο κλειδί κρυπτογράφησης (βασισμένο σε κλειδί ανταλλαγής) και πιστοποιητικό (βασισμένο σε κάποια πιστοποίηση).
3. Συμμετρική κρυπτογραφία (βασισμένη σε κρυπτογράφηση επικοινωνίας).

Κατά τη διάρκεια της πρώτης φάσης, η διαπραγμάτευση client και server χρησιμοποιεί κρυπτογραφικούς αλγορίθμους [27]. Οι αλγόριθμοι που υποστηρίζονται για την κάθε φάση είναι:

- Για δημόσιου κλειδιού κρυπτογράφηση: RSA, Diffie-Hellman, DSA ή Fortezza.
- Για συμμετρική κρυπτογραφία: RC2, RC4, IDEA, DES, Triple DES ή AES.
- Για μονόδρομες hash συναρτήσεις: MD5 ή SHA.

2.6.2 HTTPS

Το HTTPS είναι ένα πρωτόκολλο που είναι συντακτικά ίδιο με το HTTP (σχέδιο που χρησιμοποιείται κανονικά για την πρόσβαση σε πόρους που

χρησιμοποιούν το HTTP). Το URL του HTTPS δηλώνει ότι το HTTP πρόκειται να χρησιμοποιηθεί σε διαφορετικό port (το 443 αντί για το 80) και με ένα πρόσθετο επίπεδο κρυπτογράφησης/πιστοποίησης (SSL) μεταξύ του HTTP και του TCP. Αυτό το σύστημα εφευρέθηκε από την εταιρία επικοινωνιών Netscape για να παρέχει κρυπτογραφημένη και πιστοποιημένη επικοινωνία. Χρησιμοποιείται ευρέως στο παγκόσμιο ιστό για την ασφάλεια ευαίσθητων επικοινωνιών, όπως συναλλαγές πληρωμής [28].

Το επίπεδο προστασίας εξαρτάται από την ορθότητα υλοποίησης του web browser (ξεφυλλιστής ιστού), το λογισμικό του server και των κρυπτογραφικών αλγορίθμων που υποστηρίζονται.

Μια εσφαλμένη εντύπωση των χρηστών πιστωτικών καρτών στον διαδίκτυο είναι ότι το HTTPS προστατεύει πλήρως τον αριθμό των καρτών τους από τους διάφορους κλέφτες. Στην πραγματικότητα, μια κρυπτογραφημένη σύνδεση του Web Server προστατεύει μόνο τον αριθμό των πιστωτικών καρτών κατά τη μεταφορά μεταξύ του υπολογιστή μας και του server. Δεν εγγυάται ότι ο ίδιος ο server είναι ασφαλής, ή ακόμα και ότι δεν έχει καταληφθεί ήδη από κάποιον Hacker.

Οι επιθέσεις στα Web sites που αποθηκεύουν δεδομένα πελατών είναι και ευκολότερες και πιο απλές από την προσπάθεια παρεμπόδισης των δεδομένων κατά τη μεταφορά. Επίσης, συνηθίζεται να πραγματοποιούνται επιθέσεις ακόμη και στους μη ασφαλείς υπολογιστές των χρηστών.

Επειδή το SSL λειτουργεί κάτω από το HTTP και δεν έχει καμία γνώση του υψηλού αυτού πρωτοκόλλου, οι SSL servers μπορούν μόνο να παρουσιάσουν ένα πιστοποιητικό (certificate) για έναν ιδιαίτερο συνδυασμό IP/port. Αυτό σημαίνει ότι στις περισσότερες περιπτώσεις δεν είναι εφικτό να χρησιμοποιηθεί το όνομα που είναι βασισμένο στο virtual hosting (εικονική φιλοξενία) με HTTPS [28]. (Αυτή η αλλαγή υπόκειται στον TLS 1.1, που θα επιτρέπει την χρήση του ονόματος βασισμένου στο virtual hosting.)

3. Εφαρμογή «Anakoinwseis_DB»

Περίληψη Κεφαλαίου

Στο κεφάλαιο αυτό θα αναφερθούμε στην εφαρμογή «Anakoinwseis_DB», που ουσιαστικά είναι μια εφαρμογή για την διαχείριση ανακοινώσεων μέσω μίας φορητής πλατφόρμας (όπως είναι το Pocket PC), από μια απομακρυσμένη περιοχή και χωρίς να είναι απαραίτητο να υπάρχει διαρκή επικοινωνία με τον server. Πιο συγκεκριμένα θα αναφερθούμε στο τι κάνει, με ποιο τρόπο δουλεύει, πως κατασκευάστηκε και στα συμπεράσματα στα οποία καταλήξαμε από την ανάπτυξη αυτής της εφαρμογής.

3.1 Εισαγωγή

Λόγω της γρήγορης επέκτασης των κυψελοειδών (κινητή τηλεφωνία), ασύρματων (Wireless LAN), και δορυφορικών επικοινωνιών, δόθηκε η δυνατότητα σε κινούμενους χρήστες να έχουν πρόσβαση σε οποιαδήποτε δεδομένα, οπουδήποτε και εάν βρίσκονται, και σε οποιαδήποτε στιγμή. Αυτή η εξέλιξη των επικοινωνιών σε συνδυασμό με τις ραγδαίες εξελίξεις στην τεχνολογία των φορητών συσκευών, οδήγησε και στην ανάπτυξη των φορητών βάσεων δεδομένων (Mobile Database). Οι φορητές βάσεις δεδομένων μπορούν να υπάρχουν μέσα σε φορητές συσκευές και να δουλεύουν εκεί αυτόνομα, αλλά και σε επικοινωνία ανά τακτά χρονικά διαστήματα με μία κεντρική βάση δεδομένων, με την οποία ανταλλάσσουν δεδομένα.

Έτσι λοιπόν η εφαρμογή που παρουσιάζεται σε αυτό το κεφάλαιο, αλλά και των εφαρμογών των επόμενων τριών κεφαλαίων (Κεφ. 4, 5, και 6) είναι αποτέλεσμα αυτής της θεαματικής εξέλιξης. Η εφαρμογή αυτή αποτελεί ένα εργαλείο για την διαχείριση των ανακοινώσεων που βρίσκονται σε μία κεντρική βάση δεδομένων, από μια φορητή συσκευή (Pocket PC) που μπορεί να βρίσκεται οπουδήποτε σε μια περιοχή που διαθέτει κάποιο δίκτυο επικοινωνίας ή ακόμη και όχι, και να επικοινωνεί με την κεντρική βάση όποτε αυτό είναι εφικτό. Από τη στιγμή που πρόκειται για μια εφαρμογή διαχείρισης, αυτός που τη χειρίζεται πρέπει να έχει δικαιώματα διαχειριστή (Administrator), γιατί σε μία διαφορετική περίπτωση θα μπορούσε να γίνει αλλοίωση των δεδομένων της βάσης.

Ο διαχειριστής αυτής της εφαρμογής μπορεί να διαβάσει, εισαγάγει, ενημερώσει και να διαγράψει οποιαδήποτε ανακοίνωση και οποτεδήποτε το κρίνει απαραίτητο ακόμη και όταν δεν είναι διαθέσιμο κάποιο δίκτυο. Στην περίπτωση αυτή θα ενημερωθεί η κεντρική βάση δεδομένων με τις ανακοινώσεις, όποτε αυτό θα είναι δυνατό. Τέλος, λόγω αυτών των δυνατοτήτων που διαθέτει αυτή η εφαρμογή, την κάνουν ιδιαίτερα ευέλικτη στις διάφορες δυσκολίες που μπορεί να εμφανιστούν στην επικοινωνία.

3.2 Τι κάνει η εφαρμογή;

Ο χρήστης που χρησιμοποιεί αυτή την εφαρμογή έχει την δυνατότητα να βλέπει τις ανακοινώσεις που βρίσκονται στην τοπική βάση, είτε σαν γραμμές σε ένα

πίνακα περιεχομένων, είτε να τις δει αναλυτικά χρησιμοποιώντας τον κωδικό της ανακοίνωσης. Επίσης, μπορεί να εισάγει μία καινούρια ανακοίνωση στον πίνακα της τοπικής βάσης, συμπληρώνοντας βέβαια όμως τα απαραίτητα στοιχεία που απαιτούνται. Εκτός βέβαια από την δυνατότητα, του να εισάγει μια καινούρια ανακοίνωση, έχει ακόμη την δυνατότητα να επεξεργαστεί τις ήδη υπάρχουσες ανακοινώσεις που πιθανόν να τις είχε ακόμη εισάγει και κάποιος άλλος, αλλά και αυτή που μόλις εισήγαγε. Εκτός βέβαια από τα παραπάνω έχει και την δυνατότητα να σβήσει μία ανακοίνωση από την τοπική βάση δεδομένων, την οποία την είχε εισάγει ο ίδιος ή υπήρχε ήδη στην βάση και την είχε εισαγάγει ακόμη και κάποιος άλλος.

Η σημαντικότερη όμως δυνατότητα που παρέχει αυτή η εφαρμογή είναι ο συγχρονισμός της τοπικής βάσης δεδομένων με την απομακρυσμένη κεντρική βάση δεδομένων. Για να γίνει όμως αυτό, το πιο σημαντικό από όλα είναι το Pocket PC να έχει την δυνατότητα σύνδεσης στο διαδίκτυο (Internet). Αυτό όμως εξαρτάται από τις δυνατότητές του, έτσι λοιπόν θα μπορούσε να συνδεθεί: 1) σε κάποιο τοπικό ασύρματο δίκτυο (Wireless LAN, Bluetooth, κ.τ.λ.), 2) σε κάποιο κυψελοειδές δίκτυο κινητής τηλεφωνίας (GPRS , 3G, κ.τ.λ.), 3) σε κάποιο δορυφορικό δίκτυο επικοινωνιών και 4) ακόμη και σε κάποιο ενσύρματο δίκτυο (Dial-up, LAN, ActiveSync, κ.τ.λ.). Από τη στιγμή που υπάρχει στο Pocket PC κάποια σύνδεση στο διαδίκτυο, τότε το επόμενο βήμα είναι να γίνουν οι απαραίτητες ρυθμίσεις στην εφαρμογή και από την στιγμή που ο χρήστης διαθέτει κάποιο Username και Password για την κεντρική βάση, μπορεί να πραγματοποιήσει το συγχρονισμό. Ύστερα από την πραγματοποίηση του συγχρονισμού ό,τι αλλαγές έγιναν στην τοπική βάση δεδομένων περνούν και στην κεντρική, αλλά και το αντίθετο. Τέλος, η διαδικασία αυτή του συγχρονισμού είναι η ίδια με αυτή που θα χρησιμοποιηθεί και στις εφαρμογές των επόμενων κεφαλαίων, οπότε κρίνεται σκόπιμο να μην αναφερθεί ξανά στα επόμενα κεφάλαια.

Τώρα όμως, θα πρέπει να αναφέρουμε και το υπόβαθρο που χρειάζεται για να δουλέψει αυτή η εφαρμογή. Συγκεκριμένα, θα πρέπει να υπάρχει σε μία κεντρική βάση δεδομένων ένας πίνακας με το όνομα «Anakoinwseis», ο οποίος περιέχει τις στήλες «ID, Sunkths, Hmeromhnia, Titlos, Keimeno» και στον οποίο αποθηκεύονται διάφορες ανακοινώσεις με χαρακτηριστικά όπως: έναν αυτόματο κωδικό, το όνομα του συντάκτη, την ημερομηνία, τον τίτλο και το κείμενο της ανακοίνωσης. Επίσης θα πρέπει να υπάρχει και μία τοπική βάση δεδομένων (είναι κάποιο αρχείο με κατάληξη *.sdf) μέσα στο Pocket PC, που θα περιέχει τον ίδιο ακριβώς πίνακα με την κεντρική βάση, χωρίς όμως απαραίτητα να περιέχει και τα ίδια δεδομένα. Τέλος, θα πρέπει να επισημάνουμε ότι εδώ ο χρήστης που χειρίζεται την εφαρμογή έχει πλήρη δικαιώματα και μπορεί να κάνει ό,τι εκείνος επιθυμεί, σε αντίθεση με τα δικαιώματα που έχει ο χρήστης στην εφαρμογή του επόμενου κεφαλαίου.

3.3 Πώς δουλεύει η εφαρμογή;

Στην ενότητα αυτή θα παρουσιάσουμε πως δουλεύει η εφαρμογή αυτή χωρίς να μπορούμε σε τεχνικά θέματα που δεν αφορούν τον απλό χρήστη. Ο τρόπος με τον οποίο αυτό μπορεί να γίνει, είναι εξηγώντας τι ακριβώς κάνει η κάθε καρτέλα (tab page) της εφαρμογής, ακολουθώντας τη σειρά με την οποία εμφανίζονται στην εφαρμογή.

3.3.1 Παρουσίαση των περιεχομένων της τοπικής βάσης δεδομένων

Μια από τις δυνατότητες που διαθέτει αυτή η εφαρμογή είναι η παρουσίαση των περιεχομένων της τοπικής βάσης δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στην καρτέλα «List» (Σχήμα 3.1) που εμφανίζεται πρώτη στην εφαρμογή και στην οποία βλέπουμε τα περιεχόμενα της τοπικής βάσης δεδομένων σε ένα πίνακα με στήλες «ID, Suntakths, Hmeromhnia, Titlos, Keimeno». Στην καρτέλα αυτή επίσης βλέπουμε ένα κουμπί (Button) «Refresh» κάτω δεξιά, το οποίο πρέπει να πατάμε κάθε φορά που πραγματοποιείται μία αλλαγή στην τοπική βάση και θέλουμε να δούμε τα περιεχόμενά της.

ID	Suntakths	Hmeromhnia	Titlos	Keimeno
26065	drosatos	5/8/06	Pa	
24066	sa	5/6/08	AM	
24067	sa	1/1/30	en	
26066	drosatos	5/7/08	Pa	

Σχήμα 3.1: Καρτέλα «List» της εφαρμογής «Anakoinwseis_DB».

3.3.2 Εισαγωγή μιας καινούριας ανακοίνωσης

Μια από τις δυνατότητες που διαθέτει επίσης αυτή η εφαρμογή είναι η εισαγωγή μίας καινούριας ανακοίνωσης στη τοπική βάση δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στη δεύτερη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή και είναι η «Insert» (Σχήμα 3.2). Η καρτέλα αυτή αποτελείται από τέσσερα πεδία κειμένου και ένα κουμπί «Insert» κάτω δεξιά για την εισαγωγή μίας καινούργιας ανακοίνωσης. Πιο συγκεκριμένα στο πρώτο πεδίο κειμένου, από πάνω προς τα κάτω, μπορούμε να εισάγουμε το όνομα αυτού που έχει συντάξει την ανακοίνωση (το όνομα πρέπει να είναι μέχρι 50 χαρακτήρες). Στο δεύτερο πεδίο κειμένου εισάγουμε την ημερομηνία που έγινε η σύνταξη σε συγκεκριμένη όμως μορφή, όπως υποδηλώνεται, δηλαδή στην μορφή «YYYY-MM-DD hh:mm», χωρίς όμως να είναι απαραίτητο να συμπληρώσουμε όλα αυτά τα στοιχεία, π.χ. «2006-07-20», «2006». Επίσης, στο τρίτο και το τέταρτο πεδίο κειμένου μπορούμε να εισάγουμε τον τίτλο (μέχρι 100 χαρακτήρες) και το κείμενο της ανακοίνωσης,

αντίστοιχα. Τέλος, για να γίνει αυτή η καινούργια εισαγωγή στην τοπική βάση δεδομένων, αρκεί να πατήσουμε το κουμπί «Insert».

Σχήμα 3.2: Καρτέλα «Insert» της εφαρμογής «Anakoinwseis_DB».

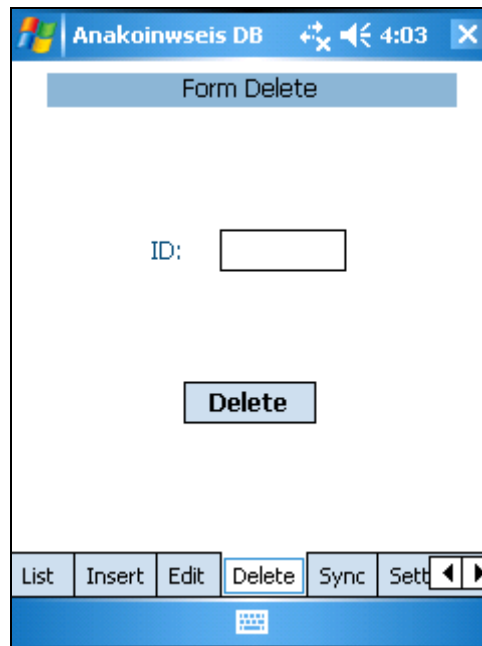
3.3.3 Παρουσίαση και επεξεργασία μιας υπάρχουσας ανακοίνωσης

Μια από τις δυνατότητες που διαθέτει επίσης αυτή η εφαρμογή είναι η παρουσίαση και επεξεργασία μιας υπάρχουσας ανακοίνωσης στην τοπική βάση δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στην τρίτη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή και είναι η «Edit» (Σχήμα 3.3). Η καρτέλα αυτή αποτελείται συγκεντρωτικά από πέντε πεδία κειμένου και από δύο κουμπιά, το ένα είναι το «OK» (πάνω δεξιά) και το άλλο είναι το «Update» (κάτω δεξιά). Στην καρτέλα αυτή μπορούμε να δούμε τα δεδομένα μίας ήδη υπάρχουσας ανακοίνωσης με βάση το κωδικό της, καθώς επίσης και να την επεξεργαστούμε, αποθηκεύοντας κιόλας τις αλλαγές. Στο πρώτο πεδίο κειμένου εισάγουμε τον κωδικό της ανακοίνωσης, που πιθανόν έχουμε δει πριν από την καρτέλα «List» ή γνωρίζουμε, και στην συνέχεια πατάμε το κουμπί «OK». Ύστερα από αυτό στα πεδία κειμένου 2, 3, 4, και 5 (δηλ. του συντάκτη, της ημερομηνίας, του τίτλου και του κείμενου) εμφανίζονται τα δεδομένα της ανακοίνωσης με τον συγκεκριμένο κωδικό. Τέλος, μπορούμε να κάνουμε οποιεσδήποτε αλλαγές θέλουμε στα δεδομένα που εμφανίστηκαν και να τις αποθηκεύσουμε στην τοπική βάση δεδομένων πατώντας πολύ απλά το κουμπί «Update».

Σχήμα 3.3: Καρτέλα «Edit» της εφαρμογής «Anakoinwseis_DB».

3.3.4 Διαγραφή μιας υπάρχουσας ανακοίνωσης

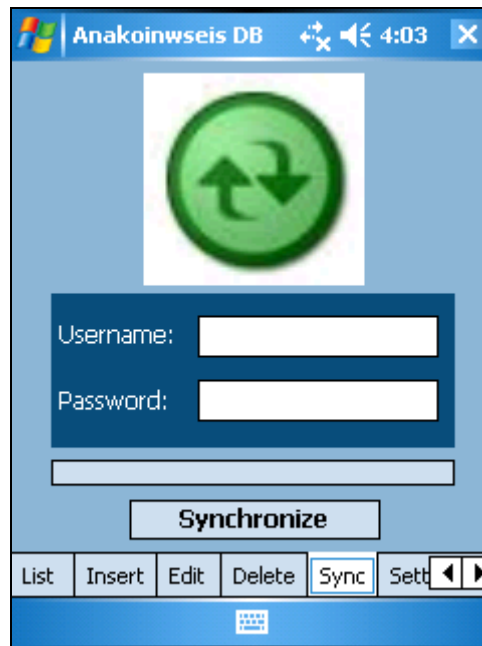
Μια από τις δυνατότητες που διαθέτει επίσης αυτή η εφαρμογή είναι η διαγραφή μιας υπάρχουσας ανακοίνωσης στη τοπική βάση δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στη τέταρτη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή και είναι η «Delete» (Σχήμα 3.4). Η καρτέλα αυτή αποτελείται από ένα πεδίο κειμένου και από ένα κουμπί το «Delete» (κάτω κέντρο). Στην καρτέλα αυτή μπορούμε να διαγράψουμε τα δεδομένα μίας ήδη υπάρχουσας ανακοίνωσης με βάση το κωδικό της. Σε αυτό το πεδίο κειμένου εισάγουμε τον κωδικό της ανακοίνωσης, που πιθανόν έχουμε δει πριν από την καρτέλα «List» ή γνωρίζουμε, και στην συνέχεια πατάμε το κουμπί «Delete». Ύστερά από αυτό θα έχει διαγραφεί από την τοπική βάση δεδομένων η ανακοίνωση με το συγκεκριμένο κωδικό.



Σχήμα 3.4: Καρτέλα «Delete» της εφαρμογής «Anakoinwseis_DB».

3.3.5 Συγχρονισμός με την κεντρική βάση δεδομένων

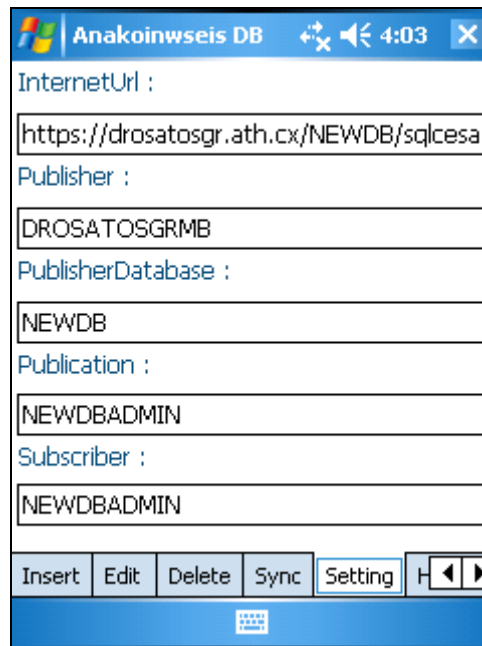
Μια από τις δυνατότητες που διαθέτει επίσης αυτή η εφαρμογή είναι ο συγχρονισμός της τοπικής βάσης δεδομένων με την απομακρυσμένη κεντρική βάση δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στην πέμπτη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή και είναι η «Sync» (Σχήμα 3.5). Η καρτέλα αυτή αποτελείται από δύο πεδία κειμένου και από ένα κουμπί το «Synchronize» (κάτω κέντρο). Στην καρτέλα αυτή μπορούμε να πραγματοποιήσουμε τον συγχρονισμό της τοπικής βάσης δεδομένων με την απομακρυσμένη κεντρική βάση, εφόσον όμως πρώτα έχουν γίνει οι κατάλληλες ρυθμίσεις στην επόμενη καρτέλα «Setting» (Σχήμα 3.6). Στο πρώτο και στο δεύτερο πεδίο κειμένου συμπληρώνουμε αντίστοιχα το «Username» και το «Password», τα οποία θα πρέπει να μας έχουν δοθεί από τον διαχειριστή της κεντρικής βάσης δεδομένων, και στη συνέχεια πατάμε το κουμπί «Synchronize». Η διαδικασία του συγχρονισμού πιθανόν να πάρει μερικά λεπτά και όταν ολοκληρωθεί θα έχει γεμίσει η μπάρα που βρίσκεται πάνω από το κουμπί.



Σχήμα 3.5: Καρτέλα «Sync» της εφαρμογής «Anakoinwseis_DB».

3.3.6 Ρυθμίσεις του συγχρονισμού

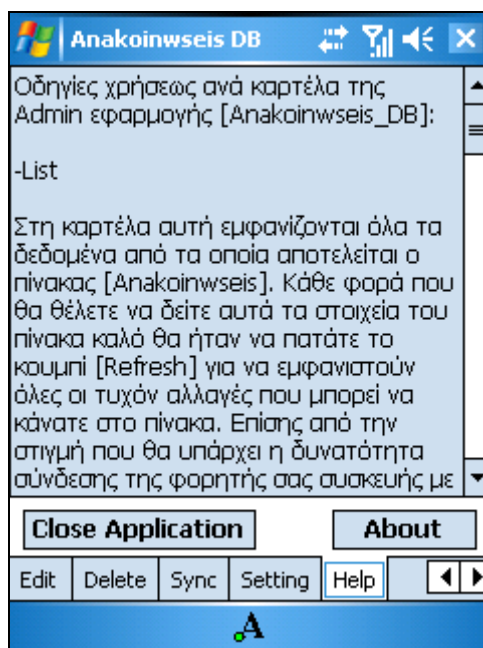
Η έκτη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή είναι η «Setting» (Σχήμα 3.6). Η καρτέλα αυτή αποτελείται συνολικά από πέντε πεδία κειμένου, χωρίς να διαθέτει τίποτα άλλο. Στην καρτέλα αυτή συμπληρώνουμε τις απαραίτητες ρυθμίσεις που χρειάζονται για την πραγματοποίηση του συγχρονισμού. Το πρώτο πεδίο κειμένου που εμφανίζεται είναι το «InternetUrl» και αποτελεί την διεύθυνση διαδικτύου στην οποία βρίσκεται η κεντρική βάση δεδομένων. Το επόμενο είναι το «Publisher» και αποτελεί το όνομα του server στον οποίο βρίσκεται η βάση δεδομένων. Το πεδίο κειμένου με το όνομα «PublisherDatabase» αποτελεί το όνομα της κεντρικής βάσης δεδομένων. Επίσης, το πεδίο με όνομα «Publication» αποτελεί το όνομα του Merge Replication που υπάρχει στην κεντρική βάση για τη διαδικασία συγχρονισμού του πίνακα «Anakoinwseis». Το τελευταίο πεδίο κειμένου που εμφανίζεται σε αυτή την καρτέλα είναι το «Subscriber» και είναι το όνομα με το οποίο συνδέεται η εφαρμογή στο Replication της βάσης.



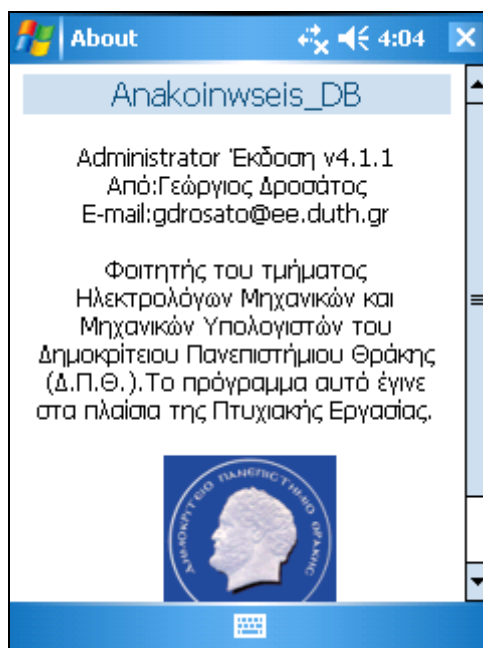
Σχήμα 3.6: Καρτέλα «Setting» της εφαρμογής «Anakoinwseis_DB».

3.3.7 Οδηγίες της εφαρμογής

Η έβδομή και τελευταία κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή είναι η «Help» (Σχήμα 3.7). Η καρτέλα αυτή αποτελείται από ένα μεγάλο πεδίο κειμένου και από δύο κουμπιά, το ένα είναι το «Close Application» (κάτω αριστερά) και το άλλο είναι το «About» (κάτω δεξιά). Στο μοναδικό πεδίο κειμένου που διαθέτει αναγράφονται όλες οι απαραίτητες οδηγίες χρήσεως της εφαρμογής. Το κουμπί «Close Application», αυτό που κάνει είναι να τερματίζει την εφαρμογή, λόγω της ιδιοτροπίας που παρουσιάζουν τα Windows Mobile 5.0, πατώντας το «X» να μην κλείνει η εφαρμογή και απλά να γίνεται «Hide». Τέλος, το κουμπί «About» εμφανίζει την φόρμα του σχήματος 3.8 με πληροφορίες για τον κατασκευαστή.



Σχήμα 3.7: Καρτέλα «Help» της εφαρμογής «Anakoinwseis_DB».



Σχήμα 3.8: Φόρμα «About» της εφαρμογής «Anakoinwseis_DB».

3.4 Διαδικασία κατασκευής

Στην ενότητα αυτή θα γίνει μία προσπάθεια να περιγραφούν τα βήματα που πρέπει να ακολουθήσει κάποιος για την κατασκευή αυτής της εφαρμογής (για την βασική υπόδειξη αυτών των βημάτων, βλέπε στο Παράρτημα Α). Έτσι λοιπόν θα ακολουθήσει μία σειρά από βήματα τα οποία απαιτούνται για τη σωστή κατασκευή και λειτουργία της εφαρμογής.

3.4.1 Κατασκευή κεντρικής βάσης δεδομένων και πίνακα

Σαν πρώτο βήμα είναι η κατασκευή της κεντρικής βάσης δεδομένων «NEWDB», η οποία θα φιλοξενήσει τον πίνακα «Anakoinwseis». Αυτό πολύ απλά μπορεί να γίνει με τις εξής SQL εντολές:

```
CREATE DATABASE NEWDB;
```

Από την στιγμή που υπάρχει η βάση δεδομένων μπορούμε πλέον να κατασκευάσουμε τον πίνακα «Anakoinwseis» μέσα σε αυτή. Η κατασκευή του πίνακα γίνεται επίσης πολύ απλά με τις εξής SQL εντολές:

```
CREATE TABLE Anakoinwseis (ID INTEGER IDENTITY (1,1) CONSTRAINT  
pkItemID PRIMARY KEY, Suntakths NVarChar (50), Hmeromhnia datetime,  
Titlos NVarChar (100), Keimeno NText);
```

Στο σημείο αυτό θα πρέπει να επισημάνουμε ότι όλες αυτές οι SQL εντολές εκτελούνται σε ένα εργαλείο του SQL SERVER 2005 που ονομάζεται SQL Server Management Studio και στο οποίο γενικά θα πραγματοποιήσουμε όλα όσα έχουν γενικά σχέση με βάσεις δεδομένων. Στην συγκεκριμένη περίπτωση για να εκτελέσουμε τις SQL εντολές απλά δημιουργούμε ένα «New Query».

3.4.2 Δημιουργία Publication

Από τη στιγμή που η βάση δεδομένων και ο πίνακας έχουν κατασκευαστεί το επόμενο βήμα είναι να κατασκευαστεί ένα Publication στην κεντρική βάση. Πριν από τη δημιουργία ενός καινούργιου Publication, πρέπει να σιγουρευτούμε ότι βρίσκεται σε λειτουργία ο SQL Server Agent. Κατά την δημιουργία του καινούριου Publication επιλέγουμε τη βάση δεδομένων που θα χρησιμοποιήσουμε, στη συγκεκριμένη περίπτωση την «NEWDB», στη συνέχεια επιλέγουμε ως τύπο Publication το Merge Replication και ως τύπο Subscriber τον SQL Server 2005 Mobile Edition.

Τώρα σειρά έχει η επιλογή του πίνακα και των στηλών που θα χρησιμοποιηθούν, στη συγκεκριμένη περίπτωση ο πίνακας είναι ο «Anakoinwseis» και όλες οι στήλες του. Τέλος, είναι πολύ σημαντικό να ρυθμίσουμε ποια θα είναι η ασφάλεια του Agent, που στην συγκεκριμένη περίπτωση επιλέγουμε να είναι SQL Server Agent service Account.

3.4.3 Ρύθμιση του δικτυακού συγχρονισμού

Σε αυτό το σημείο τώρα έχει σειρά η ρύθμιση του δικτυακού συγχρονισμού, η οποία εάν υπάρχει ήδη δεν χρειάζεται να ξαναδημιουργηθεί. Για να πραγματοποιηθεί όμως θα πρέπει πρώτα να βρίσκεται σε λειτουργία ο IIS (πρόγραμμα του λειτουργικού συστήματος Windows XP Professional SP2 και όχι μόνο) και να είναι κατάλληλα ρυθμισμένος ώστε να δουλεύει σε ασφαλή λειτουργία με το πρωτόκολλο HTTPS. Αυτό το πετυχαίνουμε με την κατασκευή Server Certificates με την βοήθεια του OpenSSL (βλέπε Παράρτημα C.1). Για την δική μας ευκολία, θα μπορούσαμε να δώσουμε στον IIS κάποιο συγκεκριμένο domain name, αυτό μπορούμε να το επιτύχουμε με την βοήθεια του DynDNS (βλέπε Παράρτημα C.2).

Οπότε τώρα μπορούμε πλέον να ξεκινήσουμε τις ρυθμίσεις, έτσι λοιπόν επιλέγουμε σαν τύπο Subscriber, τον SQL Server Mobile Edition και δημιουργούμε ένα καινούργιο «Virtual Directory» στο «Default Web Site» του IIS. Στη συνέχεια επιλέγουμε σαν ασφαλές κανάλι επικοινωνίας το SSL χωρίς όμως να απαιτείται Client Certificate. Εδώ θα πρέπει να σημειωθεί ότι έγιναν προσπάθειες για χρήση Client Certificates χωρίς όμως να έχει αποτελέσματα, ο λόγος για τον οποίο δεν δούλεψαν δεν έγινε αντιληπτός, πολύ πιθανόν να υπήρχε κάποια ασυμβατότητα ανάμεσα στον IIS και στην εφαρμογή. Στον Browser όμως των Windows Mobile 5.0 του Pocket PC, τα Client Certificates δούλεψαν αμέσως. Τέλος, επιλέγουμε σαν Client Authentication, ο «client» να συνδέεται ανώνυμα στο IIS χωρίς να χρειάζεται κάποιο Username και Password.

Τέλος, εάν κάποιος επιθυμεί να δοκιμάσει κατά πόσο αυτό το κανάλι επικοινωνίας (HTTPS) είναι ασφαλές, έτσι ώστε να χρησιμοποιηθεί για αυτή την δουλειά, μπορεί να το δοκιμάσει με κάποιο πρόγραμμα sniffer που θα ελέγχει και θα παρακολουθεί τα πακέτα αποστολής και λήψης του SQL Server μέσω του IIS. Ένα τέτοιο πρόγραμμα sniffer είναι το Ethereal (βλέπε Παράρτημα C.3).

3.4.4 Κατασκευή τοπικής βάσης δεδομένων και δημιουργία Subscription

Από τη στιγμή που έχει κατασκευαστεί το τμήμα αυτό που αφορά την κεντρική βάση δεδομένων, τώρα έχει σειρά το τμήμα που αφορά την τοπική βάση δεδομένων του Pocket PC. Σε αυτό το σημείο ξεκινάμε με την κατασκευή της τοπικής βάσης δεδομένων, δημιουργώντας ένα άδειο «*.sdf » αρχείο, το οποίο δεν περιέχει καν τον πίνακα «Anakoinwseis». Το επόμενο βήμα είναι η δημιουργία του Subscription μέσα στην τοπική βάση, η οποία με τη σειρά της θα φτιάξει τοπικά τον πίνακα «Anakoinwseis» καθώς επίσης και τις απαραίτητες ρυθμίσεις για την πραγματοποίηση του συγχρονισμού. Έτσι λοιπόν δημιουργούμε ένα καινούργιο Subscription, στο οποίο χρησιμοποιούμε το Publication (ενότητα 3.4.2) και τη δικτυακή διεύθυνση (ενότητα 3.4.3) του «Virtual Directory» του Publication στον IIS που φτιάξαμε προηγουμένως. Τέλος, ρυθμίζουμε ο client να συνδέεται ανώνυμα στον Web Server και να απαιτείται SQL Server Authentication για σύνδεση στην κεντρική βάση δεδομένων, δηλαδή κάποιος λογαριασμός χρήστη με Username και Password.

3.4.5 Κατασκευή Εφαρμογής.

Από τη στιγμή που έχουν πραγματοποιηθεί όλα τα παραπάνω βήματα, τώρα έχει σειρά η ανάπτυξη της εφαρμογής. Η εφαρμογή αναπτύχθηκε σε γλώσσα προγραμματισμού C# για το παραθυρικό περιβάλλον των Windows Mobile 5.0 του Pocket PC, χρησιμοποιώντας το Visual Studio 2005. Λόγω της έλλειψης ενός πραγματικού Pocket PC για την πραγματοποίηση των διάφορων δοκιμών, χρησιμοποιήθηκε ο προσομοιωτής Windows Mobile 5.0 Pocket PC SDK, ο οποίος ενσωματώνεται σαν εργαλείο του Visual Studio 2005.

Σε αυτό το σημείο δεν έχει νόημα να παρουσιάσουμε ολόκληρο τον κώδικα της εφαρμογής (δίνεται στο CD που βρίσκεται στο τέλος αυτής της διπλωματικής εργασίας) και να εξηγήσουμε τι κάνει, αλλά θα παρουσιάσουμε τα πιο σημαντικά σημεία αυτής της ανάπτυξης. Ξεκινώντας, θα δημιουργήσουμε ένα καινούργιο Project σε C# για «Smart Device» και συγκεκριμένα για Windows Mobile 5.0 Pocket PC με την χρήση του .NET Compact Framework 2.0. Στη συνέχεια έχει σειρά η

κατασκευή του γραφικού περιβάλλοντος και η ανάπτυξη του κώδικα. Από τη στιγμή που έχουμε τελειώσει με όλα αυτά, προχωρούμε στην εκτέλεση της εφαρμογής στον προσομοιωτή. Ένα πολύ σημαντικό σημείο είναι πριν πραγματοποιήσουμε το συγχρονισμό με την κεντρική βάση δεδομένων να έχουμε συνδέσει τον προσομοιωτή με το διαδίκτυο. Αυτό γίνεται πηγαίνοντας στα «Tools» του Visual Studio και ανοίγοντας τον «Device Emulator Manager», στον οποίο πηγαίνουμε στον συγκεκριμένο προσομοιωτή που έχουμε κάνει «Connect» και τον κάνουμε επίσης «Cradle». Εκείνη την στιγμή θα πραγματοποιηθεί σύνδεση του προσομοιωτή με το «ActiveSync» και κατεπέκταση με το διαδίκτυο.

Τα πιο σημαντικά σημεία του κώδικα τα οποία αξίζει να αναφέρουμε επιγραμματικά χωρίς πολλές λεπτομέρειες είναι:

α) Τα σημεία του κώδικα που πραγματοποιούν την εκτέλεση των εντολών «Insert, Update και Delete» στην τοπική βάση δεδομένων:

```
string strPath = @"Program Files\Anakoinwseis_DB\NEWDB.sdf";
. . .
SqlConnection cn = new SqlConnection("Data Source=" + strPath);
cn.Open();

string sSQL = "SQL εντολές INSERT, UPDATE και DELETE";
SqlCeCommand cmd = new SqlCeCommand(sSQL, cn);
cmd.CommandType = CommandType.Text;
cmd.ExecuteNonQuery();
cn.Close();
```

β) Ο κώδικας που πραγματοποιεί SQL ερωτήματα στην τοπική βάση δεδομένων:

```
SqlConnection cn = new SqlConnection("Data Source=" + strPath);
cn.Open();

string sSQL = "SELECT . . . FROM . . . WHERE . . . ";
SqlCeCommand cmd = new SqlCeCommand(sSQL, cn);
SqlCeDataReader reader = cmd.ExecuteReader();
reader.Read();
string x = reader.GetString(0);
. . .
reader.Close();
cn.Close();
```

γ) Και το κομμάτι του κώδικα που πραγματοποιεί το συγχρονισμό της τοπικής βάσης δεδομένων με την κεντρική βάση δεδομένων:

```
SqlCeReplication repl = new SqlCeReplication();

repl.InternetUrl =
@"http://[name_iis_server]/[virtual_directory]/sqlcesa30.dll";
repl.Publisher = @"[name_sql_server]";
repl.PublisherDatabase = @"[name_database]";
repl.PublisherSecurityMode = SecurityType.DBAuthentication;
repl.PublisherLogin = @"[username]";
repl.PublisherPassword = @"[password]";
repl.Publication = @"[name_publication]";
repl.Subscriber = @"[name_subscriber]";

repl.SubscriberConnectionString = "Data Source=" + strPath;
repl.Synchronize();
```

3.4.6 Τεχνικά χαρακτηριστικά

Η έκδοση της εφαρμογής αυτής είναι η «Anakoinwseis_DB» v4.1.1 και δουλεύει σε λειτουργικό σύστημα Windows Mobile 5.0 και συγκεκριμένα για Pocket PC. Για την ανάπτυξη της εφαρμογής αυτής χρειάζονται τα παρακάτω εργαλεία-προγράμματα:

1. Ο Web Server IIS v5.1 των Windows XP Professional SP2.
2. Ο SQL Server 2005 και ο SQL Server 2005 Mobile Edition.
3. Το Visual Studio 2005.
4. Το .NET Compact Framework version 2.0.
5. Το ActiveSync v4.1.
6. Τα Windows Mobile 5.0 SDK για Pocket PC.

3.5 Συμπεράσματα

Ύστερα από την παρουσίαση του τι κάνει, πως δουλεύει και πως κατασκευάστηκε αυτή η εφαρμογή τώρα έχει σειρά να αναπτύξουμε κάποια συμπεράσματα για τις δυνατότητες της. Ξεκινώντας, θα αναφερθούμε για τη μεγάλη σημασία που έχει η ύπαρξη της τοπικής βάσης δεδομένων, η οποία εάν δεν υπήρχε θα έπρεπε η εφαρμογή μας να δουλεύει σαν «client» και να είναι σε διαρκή επικοινωνία με την κεντρική βάση δεδομένων. Αυτό θα είχε συνέπειες τόσο στο κόστος της διαρκούς επικοινωνίας με την κεντρική βάση, όσο και στην κατανάλωση ισχύος που είναι εξαιρετικά σημαντική για φορητές συσκευές όπως είναι το Pocket PC. Έτσι λοιπόν με αυτό τον τρόπο και συνεχώς δε χρειάζεται να είμαστε συνδεδεμένοι, αλλά και όποτε είναι διαθέσιμο κάποιο δίκτυο μπορούμε να πραγματοποιήσουμε το συγχρονισμό. Πρακτικά, λόγω της θέσης στην οποία βρισκόμαστε μπορεί να έχουμε δίκτυο μόνο κατά διαστήματα, οπότε και πάλι μπορούμε να εξυπηρετηθούμε.

Από την άλλη η χρήση πρωτόκολλου υψηλού επιπέδου (HTTPS), κάνει την επικοινωνία πιο εύκολη και ασφαλή. Χωρίς να γίνει χρήση κάποιων πιο πολύπλοκων πρωτόκολλων χαμηλού επιπέδου που πιθανόν να μην δούλευαν κιόλας στις φορητές αυτές συσκευές λόγω των περιορισμένων δυνατοτήτων που διαθέτουν, αλλά και σε κάποιους περιορισμούς που πιθανόν να υπήρχαν στο δίκτυο επικοινωνίας (π.χ.: κυψελοειδή δίκτυα κινητής τηλεφωνίας (GPRS)). Τέλος, ο προγραμματισμός με τη χρήση της γλώσσας C# για το «.NET Compact Framework 2.0» αποδείχθηκε ένας πολύ εύκολος τρόπος για την κατασκευή της εφαρμογής σε σύντομο χρονικό διάστημα και με εξαιρετικές μάλιστα δυνατότητες.

4. Εφαρμογή «Anakoinwseis_RDB»

Περίληψη κεφαλαίου

Στο κεφάλαιο αυτό θα αναφερθούμε στην εφαρμογή «Anakoinwseis_RDB» που ουσιαστικά είναι όμοια με την εφαρμογή που παρουσιάστηκε στο κεφάλαιο 3, με την διαφορά ότι αυτή η εφαρμογή είναι μόνο για ανάγνωση (read-only). Πιο συγκεκριμένα θα αναφερθούμε στο τι κάνει, με ποιο τρόπο δουλεύει, πως κατασκευάστηκε και στα συμπεράσματα στα οποία καταλήξαμε από την ανάπτυξη αυτής της εφαρμογής.

4.1 Εισαγωγή

Η εφαρμογή αυτή βρίσκεται σε άμεση συνεργασία με την εφαρμογή που παρουσιάστηκε στο κεφάλαιο 3 και είναι στενά συνδεδεμένες. Το κύριο χαρακτηριστικό της είναι ότι η εφαρμογή αυτή είναι μόνο για ανάγνωση των ανακοινώσεων που βρίσκονται στην κεντρική βάση δεδομένων. Δηλαδή, ο χρήστης έχει την δυνατότητα να συνδεθεί με την κεντρική βάση δεδομένων, όταν βέβαια είναι διαθέσιμο κάποιο δίκτυο επικοινωνίας, και να φέρει στην τοπική-φορητή βάση δεδομένων της φορητής πλατφόρμας (του Pocket PC) όλες τις ανακοινώσεις. Βέβαια, τις ανακοινώσεις αυτές ούτε μπορεί να τις διαγράψει και να τις επεξεργαστεί, αλλά ούτε και να εισαγάγει μία καινούργια ανακοίνωση, το μόνο που μπορεί να κάνει είναι να τις διαβάσει.

Αυτή η εφαρμογή θα μπορούσε να χαρακτηριστεί, σε σύγκριση με την εφαρμογή του προηγούμενου κεφαλαίου, ως μόνο για ανάγνωση (Read Only). Από ότι γίνεται αντιληπτό, οι χρήστες που έχουν την άδεια να χειρίζονται αυτή την εφαρμογή έχουν δικαιώματα ενός απλού χρήστη (Guest), γεγονός που εξηγείται από τη στιγμή που δεν μπορούν να κάνουν τίποτα το ιδιαίτερο.

4.2 Τι κάνει η εφαρμογή;

Ο χρήστης που χρησιμοποιεί αυτή την εφαρμογή έχει τη δυνατότητα να βλέπει τις ανακοινώσεις που βρίσκονται στην τοπική βάση, είτε σαν γραμμές σε ένα πίνακα περιεχομένων, είτε να τις δει αναλυτικά χρησιμοποιώντας τον κωδικό της ανακοίνωσης. Επίσης, η εφαρμογή του παρέχει τη δυνατότητα συγχρονισμού της τοπικής βάσης δεδομένων (του πίνακα «RAnakoinwseis») με την απομακρυσμένη κεντρική βάση δεδομένων. Η διαδικασία αυτή είναι παρόμοια με αυτή που περιγράφηκε στο προηγούμενο κεφάλαιο (Ενότητα 3.2), με τη μόνη διαφορά ότι οι αλλαγές που έγιναν στην κεντρική βάση περνάνε και στην τοπική, αλλά όχι το αντίθετο.

Τώρα όμως, θα πρέπει να αναφέρουμε και στο υπόβαθρο που χρειάζεται για να δουλέψει αυτή η εφαρμογή. Συγκεκριμένα, θα πρέπει να υπάρχει στην κεντρική βάση δεδομένων ο πίνακας «Anakoinwseis» που χρησιμοποιήσαμε στην εφαρμογή του κεφαλαίου 3 και επιπλέον ένας πίνακας με το όνομα «RAnakoinwseis», ο οποίος περιέχει επίσης τις στήλες «ID, Suntakths, Hmeromhnia, Titlos, Keimeno», με τη

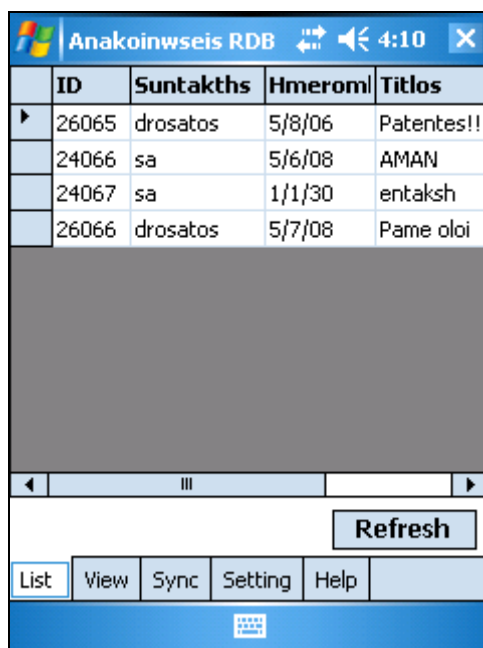
μόνη διαφορά ότι ο κωδικός (ID) της ανακοίνωσης δεν είναι αυτόματος. Εδώ θα πρέπει να πούμε ότι ο πίνακας «RAnakoimwseis» είναι πιστό αντίγραφο του πίνακα «Anakoimwseis» και ότι αλλαγές γίνονται στο δεύτερο γίνονται ταυτόχρονα και στον πρώτο, χωρίς όμως να γίνεται και το ανάποδο. Επίσης θα πρέπει να υπάρχει και μία τοπική βάση δεδομένων (είναι κάποιο αρχείο με κατάληξη *.sdf) μέσα στο Pocket PC, που θα περιέχει τον ίδιο ακριβώς πίνακα με την κεντρική βάση, χωρίς όμως απαραίτητα να περιέχει και τα ίδια δεδομένα. Τέλος, θα πρέπει να επισημάνουμε ότι εδώ ο χρήστης που χειρίζεται την εφαρμογή έχει δικαιώματα μόνο στην ανάγνωση των ανακοινώσεων και τίποτα παραπάνω.

4.3 Πώς δουλεύει η εφαρμογή;

Στην ενότητα αυτή θα παρουσιάσουμε πως δουλεύει η εφαρμογή αυτή χωρίς να μπορούμε σε τεχνικά θέματα που δεν αφορούν τον απλό χρήστη. Ο τρόπος με τον οποίο αυτό μπορεί να γίνει, είναι εξηγώντας τι ακριβώς κάνει η κάθε καρτέλα (tab page) της εφαρμογής, ακολουθώντας τη σειρά με την οποία εμφανίζονται στην εφαρμογή.

4.3.1 Παρουσίαση των περιεχομένων της τοπικής βάσης δεδομένων

Μια από τις δυνατότητες που διαθέτει αυτή η εφαρμογή είναι η παρουσίαση των περιεχομένων της τοπικής βάσης δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στην πρώτη καρτέλα την «List» (Σχήμα 4.1) και στην οποία βλέπουμε τα περιεχόμενα της τοπικής βάσης δεδομένων σε ένα πίνακα με στήλες «ID, Suntakths, Hmeromhnia, Titlos, Keimeno». Στην καρτέλα αυτή επίσης βλέπουμε ένα κουμπί (Button) «Refresh» κάτω δεξιά, το οποίο πρέπει να πατάμε κάθε φορά που κάνουμε συγχρονισμό με την κεντρική βάση δεδομένων και θέλουμε να δούμε τις αλλαγές στα περιεχόμενά της.



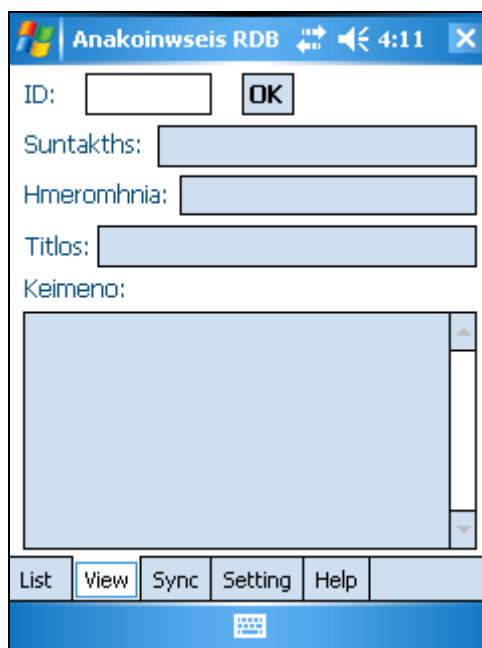
The screenshot shows a window titled 'Anakoinwseis RDB' with a standard Windows XP-style title bar. Inside the window, there is a table with four columns: 'ID', 'Suntakths', 'Hmeromi', and 'Titlos'. The table contains four rows of data. Below the table, there is a large grey rectangular area, likely a placeholder for an image or detailed view. At the bottom of the window, there is a toolbar with buttons labeled 'List', 'View', 'Sync', 'Setting', and 'Help'. A 'Refresh' button is also present above the toolbar. The 'List' button is currently selected.

ID	Suntakths	Hmeromi	Titlos
26065	drosatos	5/8/06	Patentes!!
24066	sa	5/6/08	AMAN
24067	sa	1/1/30	entaksh
26066	drosatos	5/7/08	Pame oloi

Σχήμα 4.1: Καρτέλα «List» της εφαρμογής «Anakoinwseis_RDB».

4.3.2 Παρουσίαση μιας υπάρχουσας ανακοίνωσης

Μια από τις δυνατότητες που διαθέτει επίσης αυτή η εφαρμογή είναι η παρουσίαση μιας υπάρχουσας ανακοίνωσης από την τοπική βάση δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στη δεύτερη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή και είναι η «View» (Σχήμα 4.2). Η καρτέλα αυτή αποτελείται συγκεντρωτικά από πέντε πεδία κειμένου και από ένα κουμπί «OK» (πάνω δεξιά). Στην καρτέλα αυτή μπορούμε να δούμε τα δεδομένα μίας ανακοίνωσης με βάση τον κωδικό της. Στο πρώτο πεδίο κειμένου εισάγουμε τον κωδικό της ανακοίνωσης, που πιθανόν έχουμε δει πριν από την καρτέλα «List» ή γνωρίζουμε, και στη συνέχεια πατάμε το κουμπί «OK». Ύστερα από αυτό στα πεδία κειμένου 2, 3, 4, και 5 (δηλ. του συντάκτη, της ημερομηνίας, του τίτλου και του κείμενου) εμφανίζονται τα δεδομένα της ανακοίνωσης με το συγκεκριμένο κωδικό.



Σχήμα 4.2: Καρτέλα «View» της εφαρμογής «Anakoinwseis_RDB».

4.3.3 Συγχρονισμός με την κεντρική βάση δεδομένων

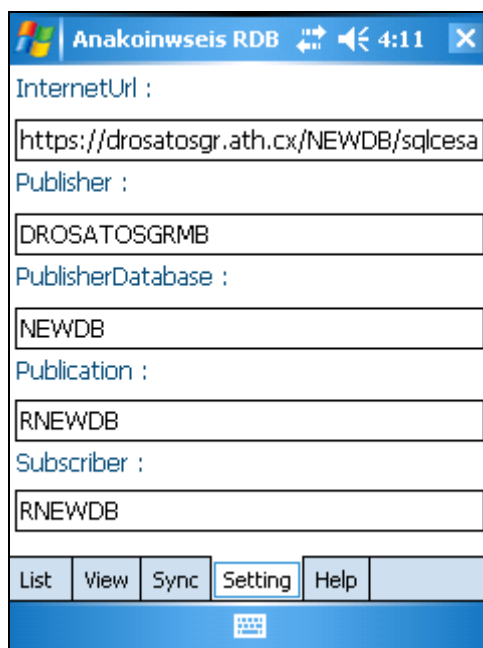
Μια από τις δυνατότητες που διαθέτει επίσης αυτή η εφαρμογή είναι ο συγχρονισμός της τοπικής βάσης δεδομένων με την απομακρυσμένη κεντρική βάση δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στη τρίτη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή και είναι η «Sync» (Σχήμα 4.3). Η καρτέλα αυτή αποτελείται από δύο πεδία κειμένου και από ένα κουμπί το «Synchronize» (κάτω κέντρο). Στην καρτέλα αυτή μπορούμε να πραγματοποιήσουμε το συγχρονισμό της τοπικής βάσης δεδομένων με την απομακρυσμένη κεντρική βάση, εφόσον όμως πρώτα έχουν γίνει οι κατάλληλες ρυθμίσεις στην επόμενη καρτέλα «Setting» (Σχήμα 4.4). Στο πρώτο και στο δεύτερο πεδίο κειμένου συμπληρώνουμε αντίστοιχα το «Username» και το «Password», τα οποία θα πρέπει να μας έχουν δοθεί από το διαχειριστή της κεντρικής βάσης δεδομένων, και στην συνέχεια πατάμε το κουμπί «Synchronize». Η διαδικασία του συγχρονισμού πιθανόν να πάρει μερικά λεπτά και όταν ολοκληρωθεί θα έχει γεμίσει η μπάρα που βρίσκεται πάνω από το κουμπί.



Σχήμα 4.3: Καρτέλα «Sync» της εφαρμογής «Anakoinwseis_RDB».

4.3.4 Ρυθμίσεις του συγχρονισμού

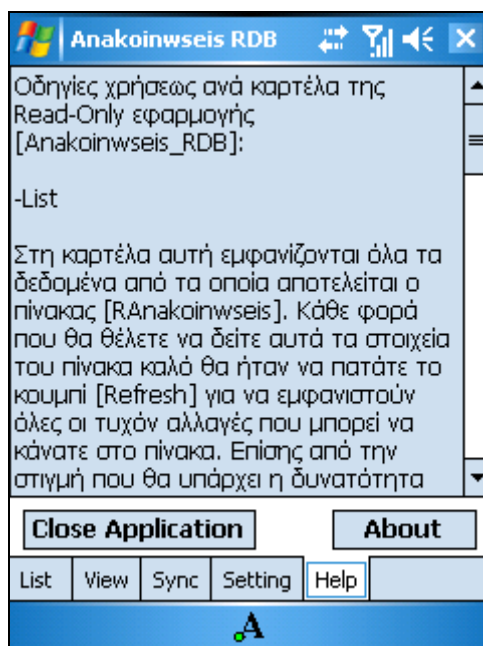
Η τέταρτη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή είναι η «Setting» (Σχήμα 4.4). Η καρτέλα αυτή αποτελείται συνολικά από πέντε πεδία κειμένου, χωρίς να διαθέτει τίποτα άλλο. Στην καρτέλα αυτή συμπληρώνουμε τις απαραίτητες ρυθμίσεις που χρειάζονται για την πραγματοποίηση του συγχρονισμού. Το πρώτο πεδίο κειμένου που εμφανίζεται είναι το «InternetUrl» και αποτελεί την διεύθυνση διαδικτύου στην οποία βρίσκεται η κεντρική βάση δεδομένων. Το επόμενο είναι το «Publisher» και αποτελεί το όνομα του server στον οποίο βρίσκεται η βάση δεδομένων. Το πεδίο κειμένου με το όνομα «PublisherDatabase» αποτελεί το όνομα της κεντρικής βάσης δεδομένων. Επίσης, το πεδίο με όνομα «Publication» αποτελεί το όνομα του Merge Replication που υπάρχει στην κεντρική βάση για τη διαδικασία του συγχρονισμού του πίνακα «RAnakoinwseis». Το τελευταίο πεδίο κειμένου που εμφανίζεται σε αυτή την καρτέλα είναι το «Subscriber» και είναι το όνομα με το οποίο συνδέεται η εφαρμογή στο Replication της βάσης.



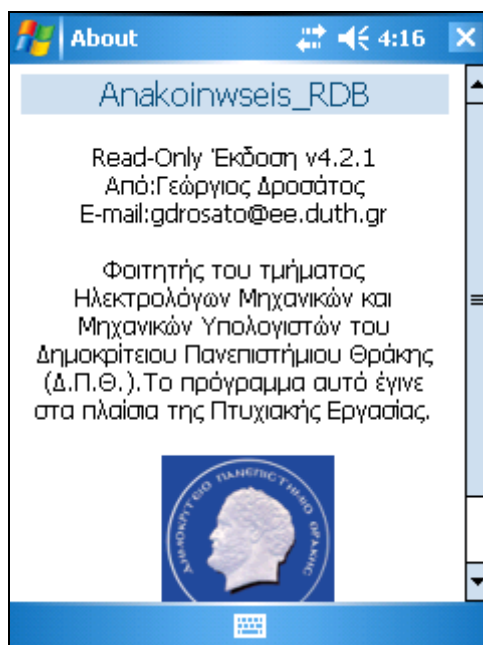
Σχήμα 4.4: Καρτέλα «Setting» της εφαρμογής «Anakoinwseis_RDB».

4.3.5 Οδηγίες της εφαρμογής

Η πέμπτη και τελευταία κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή είναι η «Help» (Σχήμα 4.5). Η καρτέλα αυτή αποτελείται από ένα μεγάλο πεδίο κειμένου και από δύο κουμπιά, το ένα είναι το «Close Application» (κάτω αριστερά) και το άλλο είναι το «About» (κάτω δεξιά). Στο μοναδικό πεδίο κειμένου που διαθέτει αναγράφονται όλες οι απαραίτητες οδηγίες χρήσεως της εφαρμογής. Το κουμπί «Close Application», αυτό που κάνει είναι να τερματίζει την εφαρμογή, λόγω της ιδιοτροπίας που παρουσιάζουν τα Windows Mobile 5.0, πατώντας το «X» να μην κλείνει η εφαρμογή και απλά να γίνεται «Hide». Τέλος, το κουμπί «About» εμφανίζει την φόρμα του σχήματος 4.6 με πληροφορίες για τον κατασκευαστή.



Σχήμα 4.5: Καρτέλα «Help» της εφαρμογής «Anakoinwseis_RDB».



Σχήμα 4.6: Φόρμα «About» της εφαρμογής «Anakoinwseis_RDB».

4.4 Διαδικασία κατασκευής

Στην ενότητα αυτή θα γίνει μία προσπάθεια να περιγραφούν τα βήματα που πρέπει να ακολουθήσει κάποιος για την κατασκευή αυτής της εφαρμογής. Δε θα ξαναπούμε όμως όσα έχουν αναφερθεί ήδη στην εφαρμογή του κεφαλαίου 3 και δεν παρουσιάζουν καμία ιδιαίτερη αλλαγή. Έτσι λοιπόν θα μιλήσουμε μόνο για τα δύο πρώτα βήματα δηλαδή, για την δημιουργία του πίνακα και των triggers, και την

δημιουργία του Publication. Τα επόμενα βήματα παραμένουν τα ίδια με μερικές μόνο διευκρινήσεις:

- Όσον αφορά τη ρύθμιση του δικτυακού συγχρονισμού, από τη στιγμή που την έχουμε φτιάξει ήδη μία φορά δεν χρειάζεται να την ξαναδημιουργήσουμε.
- Στην κατασκευή της τοπικής βάσης δεδομένων και τη δημιουργία Subscription δεν παρουσιάζεται καμία ιδιαίτερη αλλαγή.
- Τέλος, όσον αφορά την κατασκευή της εφαρμογής, η διαδικασία παραμένει ίδια, μόνο που σαν πιο σημαντικά σημεία του κώδικα είναι μόνο το (β) και το (γ), γιατί το (α) δε χρησιμοποιείται σε αυτή την εφαρμογή. Όλος ο κώδικας αυτής της εφαρμογής μαζί με το Project του Visual Studio βρίσκονται στο CD που περιλαμβάνεται στο τέλος αυτής της διπλωματικής εργασίας.

4.4.1 Δημιουργία του πίνακα και των triggers στην κεντρική βάση δεδομένων

Από τη στιγμή που υπάρχει η βάση δεδομένων «NEWDB» και ο πίνακας «Anakoinwseis» (που δημιουργήσαμε στην ενότητα 3.4.1 της προηγούμενης εφαρμογής) μπορούμε πλέον να κατασκευάσουμε τον πίνακα «RAnakoinwseis». Η κατασκευή του πίνακα γίνεται πολύ απλά με τις εξής SQL εντολές:

```
CREATE TABLE RAnakoinwseis (ID INTEGER CONSTRAINT pkID PRIMARY KEY,
Suntakths NVarChar (50), Hmeromhnia datetime, Titlos NVarChar (100),
Keimeno NText);
```

Τώρα πλέον μπορούμε να φτιάξουμε τους triggers, οι οποίοι θα παρακολουθούν τις αλλαγές που γίνονται στον πίνακα «Anakoinwseis» και θα τις περνούν στον πίνακα «RAnakoinwseis», χωρίς όμως να γίνεται και το αντίθετο. Και αυτό γιατί, εάν για κάποιο λόγο αλλοιωθούν δεδομένα στον πίνακα «RAnakoinwseis», δε θα επηρεάσουν καθόλου τον πρωτότυπο πίνακα «Anakoinwseis». Οι triggers που χρησιμοποιήθηκαν είναι τρεις:

α) Ο trigger «Anakoinwseis_Insert», που όταν γίνεται μια εισαγωγή (Insert) στον πίνακα «Anakoinwseis», κάνει ταυτόχρονα την ίδια εισαγωγή στον πίνακα «RAnakoinwseis». Οι SQL εντολές δημιουργίας του είναι:

```
CREATE TRIGGER Anakoinwseis_Insert ON Anakoinwseis FOR INSERT
AS
BEGIN
INSERT RAnakoinwseis SELECT ID, Suntakths, Hmeromhnia, Titlos,
Keimeno, rowguid FROM Anakoinwseis WHERE ID IN
(SELECT ID FROM inserted)
END
```

β) Ο trigger «Anakoinwseis_Update», που όταν γίνεται μια ενημέρωση (Update) στον πίνακα «Anakoinwseis», κάνει ταυτόχρονα την ίδια ενημέρωση στον πίνακα «RAnakoinwseis». Οι SQL εντολές δημιουργίας του είναι:

```
CREATE TRIGGER Anakoinwseis_Update ON Anakoinwseis AFTER Update
AS
BEGIN
DELETE From RAnakoinwseis Where RAnakoinwseis.ID in (SELECT ID FROM
deleted)
```

```
INSERT RAnakoinwseis SELECT ID, Suntakths, Hmeromhnia, Titlos,  
Keimeno, rowguid FROM Anakoinwseis WHERE ID IN  
(SELECT ID FROM inserted)  
END
```

γ) Και ο trigger «Anakoinwseis_Delete», που όταν γίνεται μια διαγραφή (Delete) στον πίνακα «Anakoinwseis», κάνει ταυτόχρονα την ίδια διαγραφή στον πίνακα «RAnakoinwseis». Οι SQL εντολές δημιουργίας του είναι:

```
CREATE TRIGGER Anakoinwseis_Delete ON Anakoinwseis FOR DELETE  
AS  
BEGIN  
DELETE From RAnakoinwseis Where RAnakoinwseis.ID in (SELECT ID FROM  
deleted)  
END
```

Πρέπει να επισημάνουμε ότι όλες αυτές οι SQL εντολές εκτελέστηκαν στο SQL Server Management Studio και για να εκτελέσουμε τις SQL εντολές απλά δημιουργούμε ένα «New Query» στη βάση δεδομένων.

4.4.2 Δημιουργία Publication

Από τη στιγμή που ο πίνακας και οι triggers έχουν κατασκευαστεί το επόμενο βήμα είναι να κατασκευαστεί ένα Publication στην κεντρική βάση. Πριν από τη δημιουργία ενός καινούργιου Publication, πρέπει να σιγουρευτούμε ότι βρίσκεται σε λειτουργία ο SQL Server Agent. Κατά τη δημιουργία του καινούριου Publication επιλέγουμε τη βάση δεδομένων που θα χρησιμοποιήσουμε, στη συγκεκριμένη περίπτωση την «NEWDB», στη συνέχεια επιλέγουμε ως τύπο Publication το Merge Replication και ως τύπο Subscriber τον SQL Server 2005 Mobile Edition.

Τώρα σειρά έχει η επιλογή του πίνακα και των στηλών που θα χρησιμοποιηθούν, στη συγκεκριμένη περίπτωση ο πίνακας είναι ο «RAnakoinwseis» και όλες οι στήλες του. Εδώ θα πρέπει να επισημάνουμε ότι τον πίνακα που επιλέξαμε, τον ρυθμίζουμε να είναι μόνο για κατέβασμα (table download-only) στη φορητή βάση δεδομένων, έτσι λοιπόν αποφεύγεται οποιοσδήποτε κίνδυνος αλλοίωσης των δεδομένων από τους χρήστες. Τέλος, είναι πολύ σημαντικό να ρυθμίσουμε ποια θα είναι η ασφάλεια του Agent, που στη συγκεκριμένη περίπτωση επιλέγουμε να είναι SQL Server Agent service Account.

4.4.3 Τεχνικά χαρακτηριστικά

Η έκδοση της εφαρμογής αυτής είναι η «Anakoinwseis_RDB» v4.2.1 και δουλεύει σε λειτουργικό σύστημα Windows Mobile 5.0 και συγκεκριμένα για Pocket PC. Για την ανάπτυξη της εφαρμογής αυτής χρειάζονται τα παρακάτω εργαλεία-προγράμματα:

1. Ο Web Server IIS v5.1 των Windows XP Professional SP2.
2. Ο SQL Server 2005 και ο SQL Server 2005 Mobile Edition.
3. Το Visual Studio 2005.
4. Το .NET Compact Framework version 2.0.
5. Το ActiveSync v4.1.

6. Τα Windows Mobile 5.0 SDK για Pocket PC.

4.5 Συμπεράσματα

Υστέρα από την παρουσίαση της εφαρμογής «Anakoinwseis_RDB», μπορούμε πλέον τώρα να βγάλουμε τα συμπεράσματά μας από την ανάπτυξη αυτή. Το σίγουρο πάντως είναι ότι και εδώ ισχύουν τα συμπεράσματα που είχαμε βγάλει για την εφαρμογή του προηγούμενου κεφαλαίου (Κεφ. 3, ενότητα 3.5), με τη διαφορά ότι έχουν προκύψει και μερικά άλλα.

Το επιπλέον χαρακτηριστικό της εφαρμογής αυτής είναι η ασφάλεια δεδομένων που παρέχει, όταν αυτή χρησιμοποιείται από τους πολλούς και διάφορους χρήστες. Η ασφάλεια αυτή είναι αποτέλεσμα των ρυθμίσεων download-only στο Publication, αλλά και των triggers που επιτρέπουν μόνο τη μονόδρομη αντιγραφή δεδομένων. Έτσι λοιπόν εάν κάποιος επιχειρήσει να κάνει κάποια αλλαγή στα δεδομένα της τοπικής βάσης, τότε αυτή δεν θα τον αφήσει καν να την κάνει, λέγοντας του πολύ απλά ότι δεν έχει δικαιώματα. Από την άλλη, ακόμη και εάν κάποιος καταφέρει να αλλάξει τα δεδομένα στην τοπική βάση και αυτά καταφέρουν να περάσουν μέσω του συγχρονισμού στην κεντρική βάση δεδομένων και πάλι δεν θα επηρεάσουν τα αρχικά δεδομένα, γιατί αυτά βρίσκονται σε διαφορετικό πίνακα (στον πίνακα «Anakoinwseis»). Όπως βλέπουμε λοιπόν, η εφαρμογή αυτή παρέχει αξιόλογη ασφάλεια στα πρωτότυπα δεδομένα της κεντρικής βάσης.

5. Εφαρμογή «Anakoinwseis_DB_ENCR»

Περίληψη κεφαλαίου

Στο κεφάλαιο αυτό θα αναφερθούμε στην εφαρμογή «Anakoinwseis_DB_ENCR», που ουσιαστικά είναι και αυτή μια εφαρμογή για τη διαχείριση ανακοινώσεων, όπως και η εφαρμογή του 3 κεφαλαίου. Με την διαφορά όμως εδώ ότι ο κάθε χρήστης μπορεί να διαχειρίζεται μόνο τις ανακοινώσεις που έχει γράψει ο ίδιος, χωρίς όμως να μπορεί να δει καθόλου τις ανακοινώσεις των άλλων χρηστών. Επιπλέον, παρέχει ασφάλεια που αφορά την ακεραιότητα των δεδομένων στην κεντρική βάση, αλλά και στο απόρρητο των ίδιων των δεδομένων. Πιο συγκεκριμένα θα αναφερθούμε στο τι κάνει, με ποιο τρόπο δουλεύει, πως κατασκευάστηκε και στα συμπεράσματα στα οποία καταλήξαμε από την ανάπτυξη αυτής της εφαρμογής.

5.1 Εισαγωγή

Η εφαρμογή αυτή είναι αποτέλεσμα εξέλιξης της εφαρμογής που παρουσιάστηκε στο κεφάλαιο 3. Πιο συγκεκριμένα, αυτή η εφαρμογή αποτελεί επίσης ένα εργαλείο για την διαχείριση των ανακοινώσεων, με την διαφορά όμως ότι ο κάθε χρήστης που τη χρησιμοποιεί, έχει δικαίωμα να διαχειρίζεται μόνο όσες ανακοινώσεις έχει συντάξει ο ίδιος, χωρίς να μπορεί να δει τίποτα περισσότερο. Έτσι λοιπόν με αυτό τον τρόπο, ενώ υπάρχουν στην κεντρική βάση δεδομένων πάρα πολλές ανακοινώσεις από διάφορους συντάκτες, ο καθένας από τους χρήστες φέρνει κατά την διάρκεια του συγχρονισμού στην τοπική βάση δεδομένων μόνο όσες ανακοινώσεις έχει γράψει ο ίδιος. Επιπλέον, εάν επιχειρήσει ο χρήστης να εισάγει ή να τροποποιήσει μία ανακοίνωση βάζοντας ως όνομα συντάκτη ένα διαφορετικό από αυτό με το οποίο πραγματοποιεί το συγχρονισμό, τότε όταν πραγματοποιήσει συγχρονισμό ό,τι παράτυπες αλλαγές και να είχε κάνει, διαγράφονται.

Μία επιπλέον δικλείδα ασφαλείας που χρησιμοποιείται σε αυτή την εφαρμογή είναι ότι το κείμενο κάθε ανακοίνωσης αποθηκεύεται στην τοπική αλλά και κατεπέκταση στην κεντρική βάση δεδομένων κρυπτογραφημένο (με χρήση «Password»). Αυτό σημαίνει ότι κανείς άλλος δεν μπορεί να διαβάσει το κείμενο αυτό, ακόμη και εάν αυτός είναι ο διαχειριστής (Administrator) της κεντρικής βάσης δεδομένων. Επίσης, για να έχει κάποιος πρόσβαση στην τοπική βάση δεδομένων (που είναι και αυτή κρυπτογραφημένη) χρειάζεται να γνωρίζει το «Password», και αυτό για περισσότερη ακόμη ασφάλεια. Τέλος, κάτι άλλο το οποίο έχει ιδιαίτερη σημασία για τη σωστή λειτουργία της εφαρμογής είναι ότι σε περίπτωση που η εφαρμογή πρέπει να χρησιμοποιηθεί από κάποιον άλλον χρήστη, εκτός από αυτόν που την χρησιμοποιεί ήδη, θα πρέπει να διαγράψουμε πρώτα την τοπική βάση δεδομένων μας και να δημιουργήσουμε μία καινούργια. Και αυτό γιατί στην τοπική βάση δεδομένων αποθηκεύονται πληροφορίες για το Subscription, με αποτέλεσμα να δημιουργηθεί πρόβλημα στην κεντρική βάση δεδομένων όταν πραγματοποιηθεί συγχρονισμός από διαφορετικό χρήστη με το ίδιο Subscription.

5.2 Τι κάνει η εφαρμογή;

Ο χρήστης που χρησιμοποιεί αυτή την εφαρμογή έχει την δυνατότητα να βλέπει μόνο τις δικές του ανακοινώσεις που βρίσκονται στην τοπική βάση, είτε σαν γραμμές σε ένα πίνακα περιεχομένων, είτε να τις δει αναλυτικά χρησιμοποιώντας τον κωδικό της ανακοίνωσης. Επίσης, μπορεί να εισάγει μία καινούρια ανακοίνωση στον πίνακα της τοπικής βάσης, συμπληρώνοντας ως όνομα συντάκτη αυτό με το οποίο πραγματοποιεί το συγχρονισμό, καθώς επίσης και τα υπόλοιπα στοιχεία που απαιτούνται. Εκτός βέβαια από την δυνατότητα, του να εισάγει μια καινούρια ανακοίνωση, έχει ακόμη την δυνατότητα να επεξεργαστεί τις ήδη υπάρχουσες ανακοινώσεις που έχει συντάξει ο ίδιος (έτσι και αλλιώς δεν βλέπει άλλες ανακοινώσεις εκτός από τις δικές του). Εκτός βέβαια από τα παραπάνω έχει και τη δυνατότητα να σβήσει μία ανακοίνωση από την τοπική βάση δεδομένων, την οποία την είχε εισάγει και πάλι ο ίδιος. Εδώ θα πρέπει να σημειωθεί ότι πριν κάνει τίποτα από τα παραπάνω, ο χρήστης θα πρέπει πρώτα να έχει συμπληρώσει το «Password» της τοπικής βάσης δεδομένων αλλά και το «Password» που χρησιμοποιείται για την κρυπτογραφία του κειμένου της ανακοίνωσης.

Επίσης, η εφαρμογή του παρέχει τη δυνατότητα συγχρονισμού της τοπικής βάσης δεδομένων (του πίνακα «EAnakoinwseis») με την απομακρυσμένη κεντρική βάση δεδομένων. Η διαδικασία αυτή είναι παρόμοια με αυτή που περιγράφηκε σε προηγούμενο κεφάλαιο (Ενότητα 3.2), απλά και εδώ ο χρήστης θα πρέπει να διαθέτει κάποιο Username και Password για την κεντρική βάση δεδομένων, που θα του έχουν δοθεί από το διαχειριστή της.

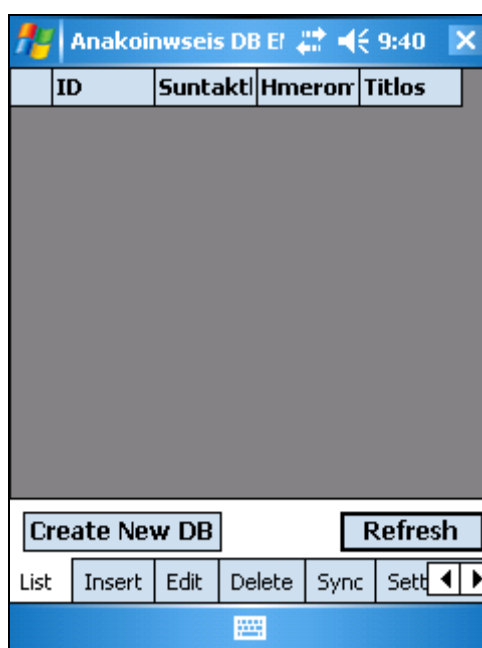
Τώρα όμως, θα πρέπει να αναφέρουμε και τι υπόβαθρο χρειάζεται για να δουλέψει αυτή η εφαρμογή. Συγκεκριμένα, θα πρέπει να υπάρχει σε μία κεντρική βάση δεδομένων ένας πίνακας με το όνομα «EAnakoinwseis», ο οποίος περιέχει τις στήλες «ID, Suntakths, Hmeromhnia, Titlos, Keimeno» και στον οποίο αποθηκεύονται διάφορες ανακοινώσεις με χαρακτηριστικά όπως: έναν αυτόματο κωδικό, το όνομα του συντάκτη, την ημερομηνία, τον τίτλο και το κείμενο της ανακοίνωσης. Επίσης θα πρέπει να υπάρχει και μία τοπική βάση δεδομένων (είναι κάποιο αρχείο με κατάληξη *.sdf) μέσα στο Pocket PC, που θα περιέχει τον ίδιο ακριβώς πίνακα με την κεντρική βάση, χωρίς όμως απαραίτητα να περιέχει και τα ίδια δεδομένα. Τέλος, θα πρέπει να επισημάνουμε ότι εδώ ο χρήστης που χειρίζεται την εφαρμογή έχει πλήρη δικαιώματα μόνο στις ανακοινώσεις που έχει συντάξει ο ίδιος.

5.3 Πώς δουλεύει η εφαρμογή;

Στην ενότητα αυτή θα παρουσιάσουμε πως δουλεύει η εφαρμογή αυτή χωρίς να μπούμε σε τεχνικά θέματα που δεν αφορούν τον απλό χρήστη. Ο τρόπος με τον οποίο αυτό μπορεί να γίνει, είναι εξηγώντας τι ακριβώς κάνει η κάθε καρτέλα (tab page) της εφαρμογής, ακολουθώντας τη σειρά με την οποία εμφανίζονται στην εφαρμογή.

5.3.1 Παρουσίαση των περιεχομένων της τοπικής βάσης δεδομένων

Μια από τις δυνατότητες που διαθέτει αυτή η εφαρμογή είναι η παρουσίαση των περιεχομένων της τοπικής βάσης δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στην καρτέλα «List» (Σχήμα 5.1) που εμφανίζεται πρώτη στην εφαρμογή και στην οποία βλέπουμε τα περιεχόμενα της τοπικής βάσης δεδομένων σε ένα πίνακα με στήλες «ID, Sunktakths, Hmeromhnia, Titlos». Στην καρτέλα αυτή επίσης βλέπουμε δύο κουμπιά, το ένα είναι το «Create New DB» (κάνω αριστερά) και το άλλο είναι το «Refresh» (κάτω δεξιά). Το κουμπί «Create New DB» το πατάμε μόνο όταν χρειαστεί να χρησιμοποιήσει την εφαρμογή κάποιος διαφορετικός χρήστης, ενώ το κουμπί «Refresh» πρέπει να πατάμε κάθε φορά που πραγματοποιείται μία αλλαγή στην τοπική βάση και θέλουμε να δούμε τα περιεχόμενά της.



Σχήμα 5.1: Καρτέλα «List» της εφαρμογής «Anakoinwseis_DB_ENCR».

5.3.2 Εισαγωγή μιας καινούριας ανακοίνωσης

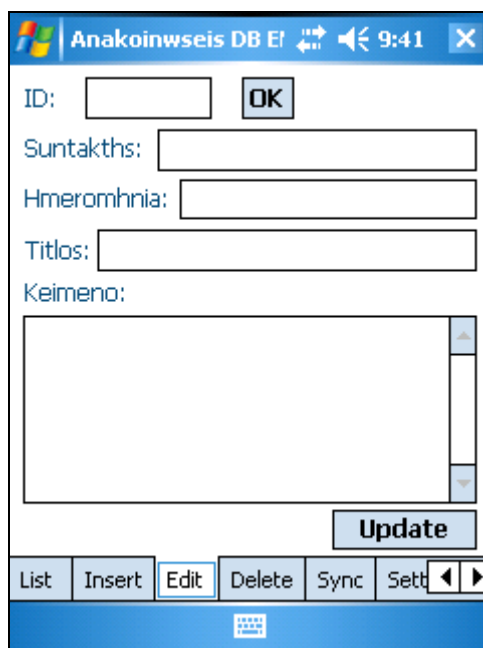
Μια από τις δυνατότητες που διαθέτει επίσης αυτή η εφαρμογή είναι η εισαγωγή μίας καινούριας ανακοίνωσης στην τοπική βάση δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στη δεύτερη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή και είναι η «Insert» (Σχήμα 5.2). Η καρτέλα αυτή αποτελείται από τέσσερα πεδία κειμένου και ένα κουμπί «Insert» (κάτω δεξιά) για την εισαγωγή μίας καινούργιας ανακοίνωσης. Πιο συγκεκριμένα στο πρώτο πεδίο κειμένου, από πάνω προς τα κάτω, μπορούμε να εισάγουμε το όνομα-Username (περισσότερες πληροφορίες για αυτό δίνονται εάν κάνουμε κλικ πάνω στο link «Sunktakths») με το οποίο κάνουμε συγχρονισμό (το όνομα αυτό δεν πρέπει να υπερβαίνει τους 50 χαρακτήρες). Στο δεύτερο πεδίο κειμένου εισάγουμε την ημερομηνία που έγινε η σύνταξη σε συγκεκριμένη όμως μορφή, όπως υποδηλώνεται, δηλαδή στην μορφή «YYYY-MM-DD hh:mm», χωρίς όμως να είναι απαραίτητο να συμπληρώσουμε όλα

αυτά τα στοιχεία, π.χ. «2006-07-20», «2006». Επίσης, στο τρίτο και το τέταρτο πεδίο κειμένου μπορούμε να εισάγουμε τον τίτλο (μέχρι 100 χαρακτήρες) και το κείμενο της ανακοίνωσης, αντίστοιχα. Τέλος, για να εισάγουμε την καινούργια αυτή ανακοίνωση στην τοπική βάση δεδομένων, πρέπει να πατήσουμε το κουμπί «Insert».

Σχήμα 5.2: Καρτέλα «Insert» της εφαρμογής «Anakoinwseis_DB_ENCR».

5.3.3 Παρουσίαση και επεξεργασία μιας υπάρχουσας ανακοίνωσης

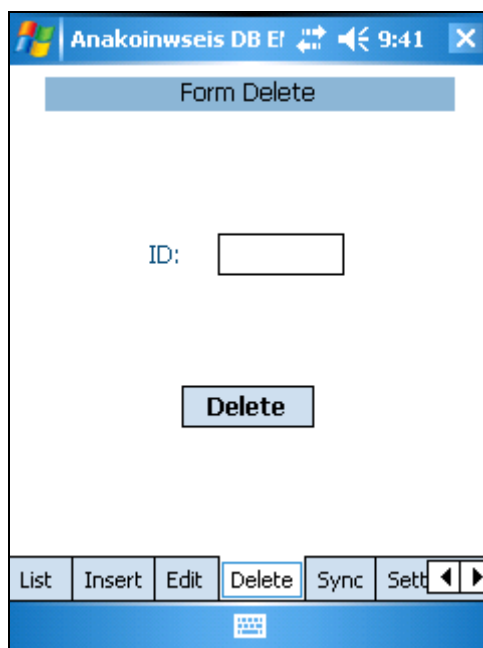
Μια από τις δυνατότητες που διαθέτει επίσης αυτή η εφαρμογή είναι η παρουσίαση και επεξεργασία μιας υπάρχουσας ανακοίνωσης στη τοπική βάση δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στη τρίτη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή και είναι η «Edit» (Σχήμα 5.3). Η καρτέλα αυτή αποτελείται συγκεντρωτικά από πέντε πεδία κειμένου και από δύο κουμπιά, το ένα είναι το «OK» (πάνω δεξιά) και το άλλο είναι το «Update» (κάτω δεξιά). Στην καρτέλα αυτή μπορούμε να δούμε τα δεδομένα μίας ήδη υπάρχουσας ανακοίνωσης με βάση το κωδικό της, καθώς επίσης και να την επεξεργαστούμε, αποθηκεύοντας κιόλας τις αλλαγές. Στο πρώτο πεδίο κειμένου εισάγουμε τον κωδικό της ανακοίνωσης, που πιθανόν έχουμε δει πριν από την καρτέλα «List» ή γνωρίζουμε, και στην συνέχεια πατάμε το κουμπί «OK». Ύστερα από αυτό στα πεδία κειμένου 2, 3, 4, και 5 (δηλ. στο συντάκτη, στην ημερομηνία, στον τίτλο και στο κείμενο) εμφανίζονται τα δεδομένα της ανακοίνωσης με το συγκεκριμένο κωδικό. Τέλος, μπορούμε να κάνουμε οποιεσδήποτε αλλαγές θέλουμε στα δεδομένα που εμφανίστηκαν και να τις αποθηκεύσουμε στην τοπική βάση δεδομένων πατώντας πολύ απλά το κουμπί «Update».



Σχήμα 5.3: Καρτέλα «Edit» της εφαρμογής «Anakoinwseis_DB_ENCR».

5.3.4 Διαγραφή μιας υπάρχουσας ανακοίνωσης

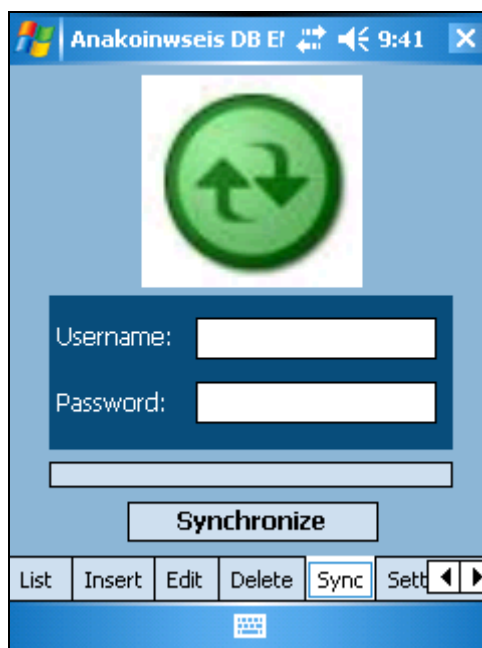
Μια από τις δυνατότητες που διαθέτει επίσης αυτή η εφαρμογή είναι η διαγραφή μιας υπάρχουσας ανακοίνωσης στην τοπική βάση δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στη τέταρτη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή και είναι η «Delete» (Σχήμα 5.4). Η καρτέλα αυτή αποτελείται από ένα πεδίο κειμένου και από ένα κουμπί το «Delete» (κάτω κέντρο). Στην καρτέλα αυτή μπορούμε να διαγράψουμε τα δεδομένα μίας ήδη υπάρχουσας ανακοίνωσης με βάση τον κωδικό της. Σε αυτό το πεδίο κειμένου εισάγουμε τον κωδικό της ανακοίνωσης, που πιθανόν έχουμε δει πριν από την καρτέλα «List» ή γνωρίζουμε, και στη συνέχεια πατάμε το κουμπί «Delete». Ύστερά από αυτό θα έχει διαγραφεί από την τοπική βάση δεδομένων η ανακοίνωση με το συγκεκριμένο κωδικό.



Σχήμα 5.4: Καρτέλα «Delete» της εφαρμογής «Anakoinwseis_DB_ENCR».

5.3.5 Συγχρονισμός με την κεντρική βάση δεδομένων

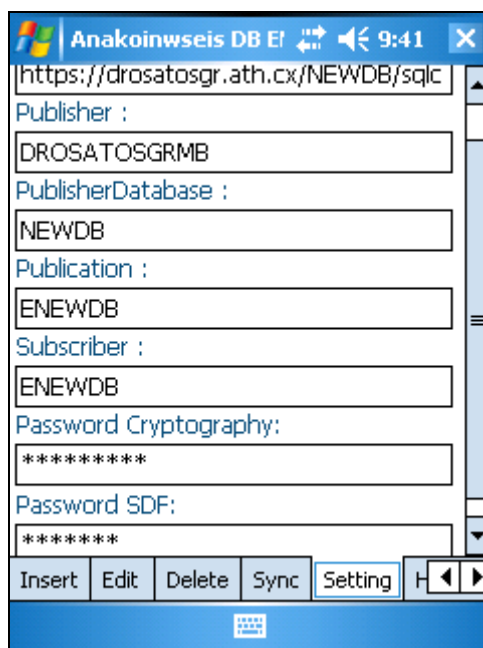
Μια από τις δυνατότητες που διαθέτει επίσης αυτή η εφαρμογή είναι ο συγχρονισμός της τοπικής βάσης δεδομένων με την απομακρυσμένη κεντρική βάση δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στην πέμπτη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή και είναι η «Sync» (Σχήμα 5.5). Η καρτέλα αυτή αποτελείται από δύο πεδία κειμένου και από ένα κουμπί το «Synchronize» (κάτω κέντρο). Στην καρτέλα αυτή μπορούμε να πραγματοποιήσουμε το συγχρονισμό της τοπικής βάσης δεδομένων με την απομακρυσμένη κεντρική βάση, εφόσον όμως πρώτα έχουν γίνει οι κατάλληλες ρυθμίσεις στην επόμενη καρτέλα «Setting» (Σχήμα 5.6). Στο πρώτο και στο δεύτερο πεδίο κειμένου συμπληρώνουμε αντίστοιχα το «Username» και το «Password», τα οποία θα πρέπει να μας έχουν δοθεί από τον διαχειριστή της κεντρικής βάσης δεδομένων, και στη συνέχεια πατάμε το κουμπί «Synchronize». Η διαδικασία του συγχρονισμού πιθανόν να πάρει μερικά λεπτά και όταν ολοκληρωθεί θα έχει γεμίσει η μπάρα που βρίσκεται πάνω από το κουμπί.



Σχήμα 5.5: Καρτέλα «Sync» της εφαρμογής «Anakoinwseis_DB_ENCR».

5.3.6 Ρυθμίσεις του συγχρονισμού

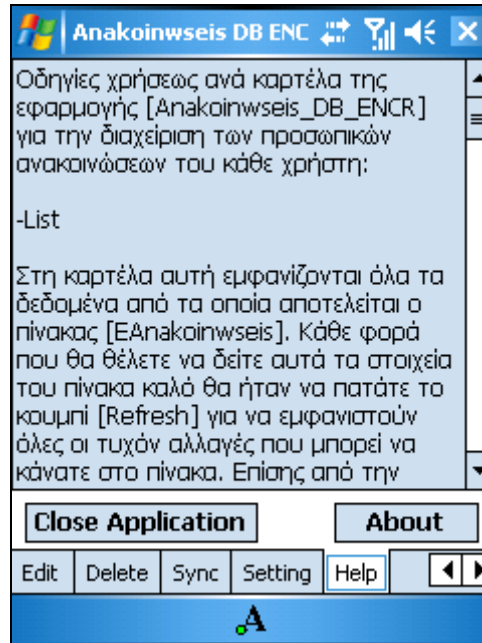
Η έκτη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή είναι η «Setting» (Σχήμα 5.6). Η καρτέλα αυτή αποτελείται συνολικά από επτά πεδία κειμένου, χωρίς να διαθέτει τίποτα άλλο. Στην καρτέλα αυτή συμπληρώνουμε τις απαραίτητες ρυθμίσεις που χρειάζονται για τη σωστή λειτουργία της εφαρμογής. Το πρώτο πεδίο κειμένου που εμφανίζεται είναι το «InternetUrl» και αποτελεί τη διεύθυνση διαδικτύου στην οποία βρίσκεται η κεντρική βάση δεδομένων. Το επόμενο είναι το «Publisher» και αποτελεί το όνομα του server στον οποίο βρίσκεται η βάση δεδομένων. Το πεδίο κειμένου με το όνομα «PublisherDatabase» αποτελεί το όνομα της κεντρικής βάσης δεδομένων. Επίσης, το πεδίο με όνομα «Publication» αποτελεί το όνομα του Merge Replication που υπάρχει στην κεντρική βάση για τη διαδικασία του συγχρονισμού του πίνακα «EAnakoinwseis». Το πεδίο κειμένου με το όνομα «Subscriber» και είναι το όνομα με το οποίο συνδέεται η εφαρμογή στο Replication της βάσης. Επιπλέον, το πεδίο κειμένου «Password Cryptography» αποτελεί το Password με το οποίο γίνεται η κρυπτογραφία-αποκρυπτογραφία του κειμένου της ανακοίνωσης με την χρήση του συμμετρικού αλγόριθμου Rijndael. Το τελευταίο πεδίο κειμένου που εμφανίζεται σε αυτή την καρτέλα είναι το «Password SDF», που αποτελεί το Password με το οποίο η εφαρμογή ανοίγει την τοπική βάση δεδομένων.



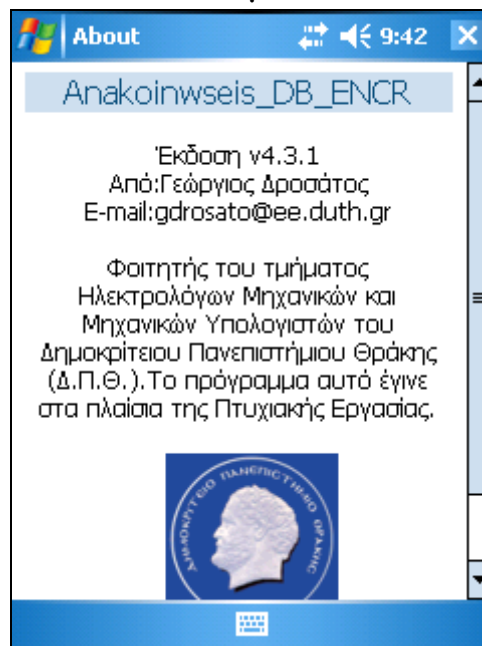
Σχήμα 5.6: Καρτέλα «Setting» της εφαρμογής «Anakoinwseis_DB_ENCR».

5.3.7 Οδηγίες της εφαρμογής

Η έβδομή και τελευταία κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή είναι η «Help» (Σχήμα 5.7). Η καρτέλα αυτή αποτελείται από ένα μεγάλο πεδίο κειμένου και από δύο κουμπιά, το ένα είναι το «Close Application» (κάτω αριστερά) και το άλλο είναι το «About» (κάτω δεξιά). Στο μοναδικό πεδίο κειμένου που διαθέτει αναγράφονται όλες οι απαραίτητες οδηγίες χρήσεως της εφαρμογής. Το κουμπί «Close Application», αυτό που κάνει είναι να τερματίζει την εφαρμογή, λόγω της ιδιοτροπίας που παρουσιάζουν τα Windows Mobile 5.0, πατώντας το «X» να μην κλείνει η εφαρμογή και απλά να γίνεται «Hide». Τέλος, το κουμπί «About» εμφανίζει την φόρμα του σχήματος 5.8 με πληροφορίες για τον κατασκευαστή.



Σχήμα 5.7: Καρτέλα «Help» της εφαρμογής «Anakoinwseis_DB_ENCR»



Σχήμα 5.8: Φόρμα «About» της εφαρμογής «Anakoinwseis_DB_ENCR».

5.4 Διαδικασία κατασκευής

Στην ενότητα αυτή θα γίνει μία προσπάθεια να περιγραφούν τα βήματα που πρέπει να ακολουθήσει κάποιος για την κατασκευή αυτής της εφαρμογής. Δε θα ξαναπούμε όμως όσα έχουν αναφερθεί ήδη στην εφαρμογή του κεφαλαίου 3 και δεν παρουσιάζουν καμία ιδιαίτερη αλλαγή. Έτσι λοιπόν θα μιλήσουμε μόνο για τα δύο πρώτα βήματα και το τελευταίο δηλαδή, για τη δημιουργία του πίνακα και των

triggers, τη δημιουργία του Publication, και την κατασκευή της εφαρμογής. Τα επόμενα βήματα παραμένουν τα ίδια με μερικές μόνο διευκρινήσεις:

- Όσον αφορά την ρύθμιση του δικτυακού συγχρονισμού, από τη στιγμή που την έχουμε φτιάξει ήδη μία φορά δε χρειάζεται να την ξαναδημιουργήσουμε.
- Στην κατασκευή της τοπικής βάσης δεδομένων και την δημιουργία Subscription δεν παρουσιάζεται καμία ιδιαίτερη αλλαγή.

5.4.1 Δημιουργία του πίνακα και των triggers στην κεντρική βάση δεδομένων

Από την στιγμή που υπάρχει η βάση δεδομένων «NEWDB» (που δημιουργήσαμε στο κεφ. 3, ενότητα 3.4.1) μπορούμε πλέον να κατασκευάσουμε τον πίνακα «EAnakoinwseis». Η κατασκευή του πίνακα γίνεται πολύ απλά με τις εξής SQL εντολές:

```
CREATE TABLE EAnakoinwseis (ID INTEGER IDENTITY (1,1) CONSTRAINT
EpkID PRIMARY KEY, Suntakths NVarChar (50), Hmeromhnia datetime,
Titlos NVarChar (100), Keimeno NText);
```

Τώρα πλέον μπορούμε να φτιάξουμε τους triggers, οι οποίοι ελέγχουν εάν το όνομα του συντάκτη της ανακοίνωσης είναι το ίδιο με αυτό του χρήστη που κάνει τις αλλαγές στον πίνακα «EAnakoinwseis». Εάν είναι το ίδιο, τότε επιτρέπει τις αλλαγές (εισαγωγής και τροποποίησης), ενώ εάν είναι διαφορετικό δεν τις αφήνει να γίνουν. Αυτό γίνεται για να μην μπορεί κάποιος να εισάγει ή να τροποποιήσει μια ανακοίνωση, όπου ο συντάκτης δεν είναι ο ίδιος. Οι triggers που χρησιμοποιήθηκαν είναι δύο:

α) Ο trigger «EAnakoinwseis_Insert_USERS», που όταν γίνεται μια εισαγωγή (Insert) στον πίνακα «EAnakoinwseis», ελέγχει εάν ο συντάκτης (Suntakths) της ανακοίνωσης είναι ο ίδιος με αυτόν που πραγματοποιεί την εισαγωγή (εάν είναι ο ίδιος την αφήνει, εάν όχι τη διαγράφει). Οι SQL εντολές για τη δημιουργία του είναι:

```
CREATE TRIGGER EAnakoinwseis_Insert_USERS ON EAnakoinwseis AFTER
INSERT
AS
BEGIN
DELETE From EAnakoinwseis Where EAnakoinwseis.ID in (SELECT ID FROM
inserted)AND NOT(EAnakoinwseis.Suntakths = SUSER_SNAME())
END
```

β) Ο trigger «EAnakoinwseis_Update_USERS», που όταν γίνεται μια ενημέρωση (Update) στον πίνακα «EAnakoinwseis», ελέγχει εάν ο συντάκτης (Suntakths) της ανακοίνωσης είναι ο ίδιος με αυτόν που πραγματοποιεί την ενημέρωση (εάν είναι ο ίδιος την αφήνει, εάν όχι την επαναφέρει). Οι SQL εντολές για την δημιουργία του είναι:

```
CREATE TRIGGER EAnakoinwseis_Update_USERS ON EAnakoinwseis instead of
Update
AS
BEGIN
IF NOT(SUSER_SNAME() IN (SELECT Suntakths FROM inserted))
```

```

UPDATE EAnakoinwseis SET Suntakths = (SELECT Suntakths FROM
deleted),Hmeromhnia = (SELECT Hmeromhnia FROM deleted),Titlos =
(SELECT Titlos FROM deleted),Keimeno = (SELECT Keimeno FROM deleted)
WHERE EAnakoinwseis.ID in (SELECT ID FROM deleted)
ELSE
UPDATE EAnakoinwseis SET Hmeromhnia = (SELECT Hmeromhnia FROM
inserted),Titlos = (SELECT Titlos FROM inserted),Keimeno = (SELECT
Keimeno FROM inserted) WHERE EAnakoinwseis.ID in (SELECT ID FROM
inserted)
END

```

5.4.2 Δημιουργία Publication

Από την στιγμή που ο πίνακας και οι triggers έχουν κατασκευαστεί, μπορούμε πλέον να κατασκευάσουμε το Publication στην κεντρική βάση δεδομένων. Πριν από τη δημιουργία ενός καινούργιου Publication, πρέπει να σιγουρευτούμε ότι βρίσκεται σε λειτουργία ο SQL Server Agent. Κατά την δημιουργία του καινούριου Publication επιλέγουμε την βάση δεδομένων που θα χρησιμοποιήσουμε, στη συγκεκριμένη περίπτωση την «NEWDB», στη συνέχεια επιλέγουμε ως τύπο Publication το Merge Replication και ως τύπο Subscriber τον SQL Server 2005 Mobile Edition.

Τώρα σειρά έχει η επιλογή του πίνακα και των στηλών του που θα χρησιμοποιηθούν, στη συγκεκριμένη περίπτωση ο πίνακας είναι ο «EAnakoinwseis» και όλες οι στήλες του. Επιπλέον, επειδή θέλουμε οι χρήστες να έχουν πρόσβαση μόνο στις ανακοινώσεις που έχουν εισάγει οι ίδιοι, χρησιμοποιούμε ένα παραμετρικό-δυναμικό φίλτρο στον πίνακα «EAnakoinwseis», το οποίο είναι:

```

SELECT <published_columns> FROM [dbo].[EAnakoinwseis] WHERE
[Suntakths] = SUSER_SNAME()

```

Τέλος, είναι πολύ σημαντικό να ρυθμίσουμε και την ασφάλεια του Agent, που στην συγκεκριμένη περίπτωση επιλέγουμε να είναι SQL Server Agent service Account.

5.4.3 Κατασκευή Εφαρμογής.

Όσον αφορά την κατασκευή της εφαρμογής, η διαδικασία παραμένει ίδια με αυτή που παρουσιάστηκε στο κεφάλαιο 3 και συγκεκριμένα στην ενότητα 3.4.5, μόνο που εδώ θα παρουσιάσουμε και κάποια επιπλέον σημεία του κώδικα που έχουν εξαιρετικό ενδιαφέρον. Πρέπει να σημειωθεί ότι ολόκληρος ο κώδικας αυτής της εφαρμογής μαζί με το Project του Visual Studio βρίσκονται στο CD που περιλαμβάνεται στο τέλος αυτής της διπλωματικής εργασίας.

Τα επιπλέον αυτά σημεία του κώδικα που παρουσιάζουν ενδιαφέρον, αφορούν τη δημιουργία μίας καινούργιας τοπικής βάσης δεδομένων, αλλά και την κρυπτογραφία-αποκρυπτογραφία του κειμένου της ανακοίνωσης. Πιο συγκεκριμένα τα σημεία αυτά είναι:

α) Ο κώδικας που χρησιμοποιείται για τη δημιουργία της τοπικής βάσης δεδομένων, αλλά και του πίνακα «EAnakoinwseis» είναι ο εξής:

```

SqlCeEngine m_cEngine = new SqlCeEngine("Data Source=" + strPath +
";Password =\"*****\"; Encrypt = TRUE ;");

m_cEngine.CreateDatabase();
SqlCeConnection m_cConn = new SqlCeConnection("Data Source=" +
strPath + "; Password =\"*****\";");

m_cConn.Open();
SqlCeCommand sqlCreateTable = m_cConn.CreateCommand();

sqlCreateTable.CommandText = "CREATE TABLE EAnakoinwseis (ID INTEGER
IDENTITY (1,1) CONSTRAINT EpkID PRIMARY KEY, Suntakths NVarChar (50),
Hmeromhnia datetime, Titlos NVarChar (100), Keimeno NText)";

sqlCreateTable.ExecuteNonQuery();
m_cConn.Close();

```

β) Ο κώδικας που χρησιμοποιείται για την κρυπτογραφία του κείμενου της ανακοίνωσης, με τη χρήση του συμμετρικού κρυπτογραφικού αλγόριθμου Rijndael είναι ο εξής:

```

public static string Encrypt(string clearText, string Password)
{

byte[] clearBytes = System.Text.Encoding.Unicode.GetBytes(clearText);

MemoryStream ms = new MemoryStream();
Rijndael alg = Rijndael.Create();

alg.Key = Createkey(Password, 32);
alg.IV = CreateIV(Password, 16);

CryptoStream cs = new CryptoStream(ms, alg.CreateEncryptor(),
CryptoStreamMode.Write);

cs.Write(clearBytes, 0, clearBytes.Length);
cs.Close();

byte[] encryptedData = ms.ToArray();

return Convert.ToBase64String(encryptedData);

}

```

γ) Και ο κώδικας που χρησιμοποιείται για την αποκρυπτογραφία του κείμενου της ανακοίνωσης, με τη χρήση του συμμετρικού κρυπτογραφικού αλγόριθμου Rijndael είναι ο εξής:

```

public static string Decrypt(string cipherText, string Password)
{

byte[] cipherBytes = Convert.FromBase64String(cipherText);

MemoryStream ms = new MemoryStream();
Rijndael alg = Rijndael.Create();

alg.Key = Createkey(Password, 32);
alg.IV = CreateIV(Password, 16);

```

```

CryptoStream cs = new CryptoStream(ms, alg.CreateDecryptor(),
CryptoStreamMode.Write);

cs.Write(cipherBytes, 0, cipherBytes.Length);
cs.Close();

byte[] decryptedData = ms.ToArray();

return System.Text.Encoding.Unicode.GetString(decryptedData, 0,
decryptedData.Length);

}

```

5.4.4 Τεχνικά χαρακτηριστικά

Η έκδοση της εφαρμογής αυτής είναι η «Anakoinwseis_DB_ENCR» v4.3.1 και δουλεύει σε λειτουργικό σύστημα Windows Mobile 5.0 και συγκεκριμένα για Pocket PC. Για την ανάπτυξη της εφαρμογής αυτής χρειάζονται τα παρακάτω εργαλεία-προγράμματα:

1. O Web Server IIS v5.1 των Windows XP Professional SP2.
2. O SQL Server 2005 και o SQL Server 2005 Mobile Edition.
3. To Visual Studio 2005.
4. To .NET Compact Framework version 2.0.
5. To ActiveSync v4.1.
6. Τα Windows Mobile 5.0 SDK για Pocket PC.

5.5 Συμπεράσματα

Υστέρα από την παρουσίαση της εφαρμογής «Anakoinwseis_DB_ENCR», μπορούμε πλέον τώρα να βγάλουμε τα συμπεράσματά μας από την ανάπτυξη αυτή. Το σίγουρο πάντως είναι ότι και εδώ ισχύουν τα συμπεράσματα που είχαμε βγάλει για την εφαρμογή του κεφαλαίου 3 (Ενότητα 3.5), με τη διαφορά ότι έχουν προκύψει και μερικά άλλα.

Το επιπλέον χαρακτηριστικό της εφαρμογής αυτής είναι η ασφάλεια, όπως άλλωστε και στην εφαρμογή του κεφαλαίου 4, με την διαφορά όμως εδώ ότι η ασφάλεια σχετίζεται περισσότερο με την έννοια του προσωπικού απορρήτου, αλλά και τη σωστή λειτουργία της κεντρικής βάσης δεδομένων. Αυτή η ασφάλεια χωρίζεται σε αυτή που οφείλεται στο δυναμικό φίλτρο του πίνακα στις ρυθμίσεις του Publication, σε αυτή λόγω των triggers και σε αυτή που οφείλεται στην κρυπτογραφία του κειμένου της ανακοίνωσης. Η χρήση του φίλτρου επιτρέπει στους διάφορες χρήστες που χρησιμοποιούν την εφαρμογή να βλέπουν και να διαχειρίζονται μόνο τις ανακοινώσεις που έχουν συντάξει οι ίδιοι, χωρίς να μπορούν ούτε καν να δουν τις ανακοινώσεις των άλλων χρηστών. Από την άλλη οι triggers εξυπηρετούν την αντιμετώπιση διάφορων παραβιάσεων που μπορεί να γίνονται από τους χρήστες, όπως είναι η εισαγωγή ή επεξεργασία ανακοινώσεων όπου ο συντάκτης δεν είναι οι ίδιοι. Τέλος, η χρήση της κρυπτογραφίας εξασφαλίζει ότι κανείς δεν θα μπορέσει να διαβάσει το περιεχόμενο του κειμένου της ανακοίνωσης, εάν δεν γνωρίζει το Password με το οποίο έχει κρυπτογραφηθεί. Αξιοσημείωτο είναι ότι ακόμη και ο διαχειριστής της κεντρικής βάσης δεδομένων δεν μπορεί να διαβάσει τα κείμενα των

ανακοινώσεων των διαφόρων χρηστών που βρίσκονται στον πίνακα «EAnakoinwseis».

6. Εφαρμογή «LogBookDive»

Περίληψη κεφαλαίου

Στο κεφάλαιο αυτό θα αναφερθούμε στην εφαρμογή «LogBookDive», που αποτελεί ουσιαστικά μια πρακτική εφαρμογή των δυνατοτήτων που μας παρέχει ο SQL Server σε συνδυασμό με την C# για φορητές πλατφόρμες (Pocket PC). Πιο συγκεκριμένα, είναι μια βοηθητική εφαρμογή για την πραγματοποίηση ασφαλών καταδύσεων στο βυθό της θάλασσας. Η εφαρμογή αυτή έχει την δυνατότητα να υπολογίζει τα επίπεδα του αζώτου στο αίμα των δυτών, αλλά και να αποθηκεύει όλες εκείνες τις πληροφορίες που χαρακτηρίζουν μία κατάδυση. Από την άλλη, η εφαρμογή αυτή παρουσιάζει παρόμοιες ιδιότητες, όσον αφορά το συγχρονισμό (όχι όμως την κρυπτογραφία των δεδομένων), με την εφαρμογή που παρουσιάστηκε στο κεφάλαιο 5. Δηλαδή και εδώ ο κάθε χρήστης (δύτης) μπορεί να βλέπει μόνο τις καταδύσεις που έχει κάνει ο ίδιος, χωρίς όμως να μπορεί να δει καθόλου τις καταδύσεις των άλλων χρηστών (δυτών). Έτσι λοιπόν, με αυτό τον τρόπο διασφαλίζεται το απόρρητο των προσωπικών του δεδομένων. Πιο αναλυτικά θα αναφερθούμε στο τι κάνει, με ποιο τρόπο δουλεύει, πως κατασκευάστηκε και στα συμπεράσματα στα οποία καταλήξαμε από την ανάπτυξη αυτής της εφαρμογής.

6.1 Εισαγωγή

Η εφαρμογή αυτή είναι αποτέλεσμα εξέλιξης της εφαρμογής που παρουσιάστηκε στο κεφάλαιο 5, και θα μπορούσε να αποτελέσει ουσιαστικά μια πρακτική-εμπορική εφαρμογή όλων όσων γνωρίσαμε σε αυτή την διπλωματική εργασία. Πιο συγκεκριμένα, αποτελεί μια βοηθητική εφαρμογή για την πραγματοποίηση ασφαλών καταδύσεων στο βυθό της θάλασσας. Η εφαρμογή αυτή έχει την δυνατότητα να υπολογίζει τα επίπεδα του αζώτου (σε μη επιτρεπτά επίπεδα προκαλεί δηλητηρίαση) στο αίμα των δυτών, με βάση το βάθος στο οποίο βρίσκονται, το χρονικό διάστημα στο οποίο παραμένουν σε αυτό, αλλά και εάν έχουν πραγματοποιήσει δεύτερη κατάδυση την ίδια μέρα. Ο υπολογισμός αυτός προκύπτει ουσιαστικά από αναζήτηση σε κάποιους εμπειρικούς πίνακες που έχουν προκύψει πρακτικά για αυτή τη δουλειά. Από την άλλη παρέχει την δυνατότητα αποθήκευσης όλων εκείνων των πληροφοριών που χαρακτηρίζουν μία κατάδυση. Αυτή η αποθήκευση γίνεται είτε τοπικά, στην βάση δεδομένων της φορητής πλατφόρμας (Pocket PC), είτε και με την διαδικασία του συγχρονισμού στην κεντρική βάση δεδομένων, όπου γενικά υπάρχουν όλες οι καταδύσεις των χρηστών (δυτών) της εφαρμογής. Όσον αφορά το συγχρονισμό, η εφαρμογή αυτή παρουσιάζει παρόμοιες ιδιότητες (όχι όμως την κρυπτογραφία των δεδομένων), με την εφαρμογή που παρουσιάστηκε στο κεφάλαιο 5. Δηλαδή και εδώ ο κάθε χρήστης (δύτης) μπορεί να βλέπει (διαχειρίζεται) μόνο τις καταδύσεις που έχει κάνει ο ίδιος, χωρίς όμως να μπορεί να δει καθόλου τις καταδύσεις των άλλων χρηστών (δυτών). Έτσι λοιπόν, με αυτό τον τρόπο διασφαλίζεται το απόρρητο των προσωπικών του δεδομένων.

Για να γίνει πιο αντιληπτό αυτό, ενώ υπάρχουν στην κεντρική βάση δεδομένων πάρα πολλές καταδύσεις από διάφορους χρήστες (δύτες), ο καθένας από τους χρήστες (δύτες) φέρνει κατά την διάρκεια του συγχρονισμού στην τοπική βάση

δεδομένων μόνο όσες καταδύσεις έχει κάνει ο ίδιος. Επιπλέον, εάν επιχειρήσει ο χρήστης (δύτης) να εισάγει ή να τροποποιήσει μία κατάδυση, εάν και αυτό δεν παρέχεται από την εφαρμογή, αλλά θα μπορούσε όμως μέσω του Query Analyzer, βάζοντας ως όνομα δύτη (χρήστη) ένα διαφορετικό από αυτό με το οποίο πραγματοποιεί το συγχρονισμό, τότε όταν πραγματοποιήσει συγχρονισμό, ό,τι παράτυπες αλλαγές και να είχε κάνει, διαγράφονται. Τέλος, κάτι άλλο το οποίο έχει ιδιαίτερη σημασία για την σωστή λειτουργία της εφαρμογής είναι ότι σε περίπτωση που η εφαρμογή πρέπει να χρησιμοποιηθεί από κάποιον άλλον χρήστη (δύτη), εκτός από αυτόν που την χρησιμοποιεί ήδη, θα πρέπει να διαγράψουμε πρώτα την τοπική βάση δεδομένων μας και να δημιουργήσουμε μία καινούργια. Και αυτό γιατί στην τοπική βάση δεδομένων αποθηκεύονται πληροφορίες για το Subscription, με αποτέλεσμα να δημιουργηθεί πρόβλημα στην κεντρική βάση δεδομένων όταν πραγματοποιηθεί συγχρονισμός από διαφορετικό χρήστη (δύτη) με το ίδιο Subscription.

6.2 Τι κάνει η εφαρμογή;

Ο χρήστης (δύτης) που χρησιμοποιεί αυτή την εφαρμογή έχει τη δυνατότητα να βλέπει μόνο τις δικές του καταδύσεις που βρίσκονται στην τοπική βάση, είτε σαν γραμμές σε ένα πίνακα περιεχομένων, είτε να τις δει αναλυτικά χρησιμοποιώντας τον κωδικό της κατάδυσης. Επίσης, μπορεί με την εισαγωγή κατάλληλων τιμών (όπως το βάθος και το χρόνο της κατάδυσης), να υπολογίσει το επίπεδο αζώτου (A, B, C, D, κ.τ.λ.) στο αίμα και στην συνέχεια μαζί με τα υπόλοιπα στοιχεία που απαιτούνται να τα εισάγει ως μία καινούρια κατάδυση στον πίνακα της τοπικής βάσης. Εκτός βέβαια από την δυνατότητα, του να εισάγει μια καινούρια κατάδυση, έχει ακόμη την δυνατότητα να σβήσει μία κατάδυση από την τοπική βάση δεδομένων, την οποία βέβαια την είχε εισάγει ο ίδιος.

Επίσης, η εφαρμογή του παρέχει τη δυνατότητα συγχρονισμού της τοπικής βάσης δεδομένων (του πίνακα «History») με την απομακρυσμένη κεντρική βάση δεδομένων. Η διαδικασία αυτή είναι παρόμοια με αυτή που περιγράφηκε σε προηγούμενο κεφάλαιο (Ενότητα 3.2), απλά και εδώ ο χρήστης (δύτης) θα πρέπει να διαθέτει κάποιο Username και Password για την κεντρική βάση δεδομένων, που θα του έχουν δοθεί από τον διαχειριστή της. Θα πρέπει να σημειωθεί, ότι πουθενά στην εφαρμογή δεν συμπληρώνουμε το όνομα του δύτη, και αυτό γιατί η εισαγωγή γίνεται αυτόματα όταν θα πραγματοποιήσει τον συγχρονισμό. Οπότε λοιπόν θέλει ιδιαίτερη προσοχή ώστε το Username που γράφουμε στον συγχρονισμό να είναι σωστό, γιατί σε μια διαφορετική περίπτωση, ό,τι είχε εισάγει ο χρήστης μετά τον προηγούμενο συγχρονισμό να διαγραφεί την επόμενη φορά που θα δώσει το σωστό Username.

Τώρα όμως, θα πρέπει να αναφέρουμε και τι υπόβαθρο χρειάζεται για να δουλέψει αυτή η εφαρμογή. Συγκεκριμένα, θα πρέπει να υπάρχει σε μία κεντρική βάση δεδομένων ένας πίνακας με το όνομα «History», ο οποίος περιέχει τις στήλες «Kwdikos, OnomaDuth, Hmeromhnia, Dive_Site, Bottle, X, Bar_Start, Time_In, Time_Out, Bar_End, PGO, SI, PGN, PG, Depth, Bot_Time, RNT, ABT, TBT» και στον οποίο αποθηκεύονται διάφορες καταδύσεις με χαρακτηριστικά όπως: έναν αυτόματο κωδικό, το όνομα του δύτη, την ημερομηνία, την περιοχή της κατάδυσης, τον τύπο της μπουκάλας, το ποσοστό της εάν είναι «EAN», τα bar στην αρχή, την ώρα κατάδυσης, την ώρα ανάδυσης, τα bar στο τέλος, το επίπεδο αζώτου από την προηγούμενη κατάδυση, το χρόνο ανάμεσα στην 1^η και τη 2^η κατάδυση, το επίπεδο αζώτου του χρόνου από την προηγούμενη κατάδυση, το επίπεδο αζώτου από τη νέα

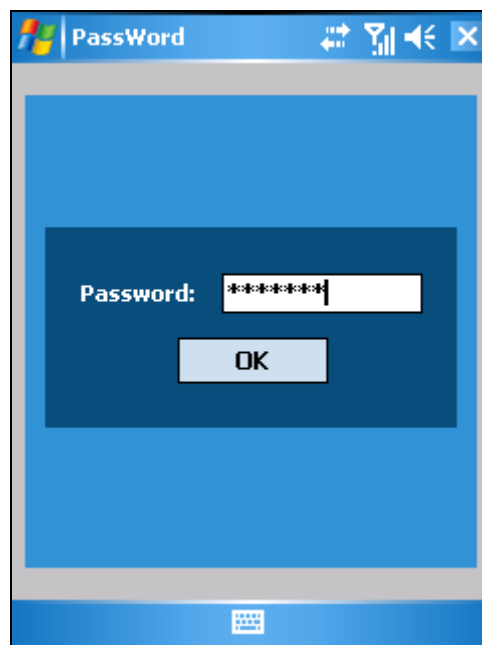
κατάδυση, το βάθος της κατάδυσης, το χρόνο κατάδυσης και άλλους τρεις χαρακτηριστικούς χρόνους της κατάδυσης. Επίσης θα πρέπει να υπάρχει και μία τοπική βάση δεδομένων (είναι κάποιο αρχείο με κατάληξη *.sdf) μέσα στο Pocket PC, που θα περιέχει τον ίδιο ακριβώς πίνακα με την κεντρική βάση, χωρίς όμως απαραίτητα να περιέχει και τα ίδια δεδομένα. Τέλος, θα πρέπει να επισημάνουμε ότι εδώ ο χρήστης (δύτης) που χειρίζεται την εφαρμογή έχει πλήρη δικαιώματα μόνο στις καταδύσεις που έχει κάνει ο ίδιος.

6.3 Πώς δουλεύει η εφαρμογή;

Στην ενότητα αυτή θα παρουσιάσουμε πως δουλεύει η εφαρμογή αυτή χωρίς να μπούμε σε τεχνικά θέματα που δεν αφορούν τον απλό χρήστη. Ο τρόπος με τον οποίο αυτό μπορεί να γίνει, είναι εξηγώντας τι ακριβώς κάνει η κάθε καρτέλα (tab page) της εφαρμογής, ακολουθώντας τη σειρά με την οποία εμφανίζονται στην εφαρμογή.

6.3.1 Έλεγχος εισόδου στην εφαρμογή

Πριν ξεκινήσουμε με την παρουσίαση των καρτελών (tabpages) του κυρίως μενού της εφαρμογής, θα πρέπει να αναφέρουμε ότι κατά την εκκίνηση ανοίγει μια φόρμα (Σχήμα 6.1) για να εισάγουμε κάποιο Password. Εάν δεν καταφέρει ο χρήστης να πληκτρολογήσει το σωστό Password που του έχει δοθεί, σε τρεις προσπάθειες τότε η εφαρμογή κλείνει αυτόματα και θα πρέπει να ξαναπροσπαθήσει εκτελώντας πάλι την εφαρμογή.



Σχήμα 6.1: Φόρμα «Password» της εφαρμογής «LogBookDive».

6.3.2 Υπολογισμός και αποθήκευση μιας κατάδυσης

Μια από τις δυνατότητες που διαθέτει αυτή εφαρμογή είναι ο υπολογισμός αλλά και η αποθήκευση των αποτελεσμάτων της κατάδυσης στη τοπική βάση δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στην καρτέλα «Dive1-2» (Σχήμα 6.2) που εμφανίζεται πρώτη στην εφαρμογή. Η καρτέλα αυτή αποτελείται συνολικά από 17 πεδία κειμένου, από τα οποία τα 7 (ή 8 ανάλογα με το τύπο της μπουκάλας) (με χρώμα άσπρο) απαιτείται να τα συμπληρώνουμε εμείς και τα υπόλοιπα (με χρώμα γαλάζιο) είναι μόνο για ανάγνωση. Επίσης, αποτελείται από 2 Checkbox και από τρία κουμπιά, το ένα είναι το «Find», δίπλα του είναι το «Run» (και τα δύο πάνω δεξιά) και το άλλο είναι το «Save» (πάνω δεξιά κάτω από το «Run»). Πιο συγκεκριμένα, τώρα θα εξηγήσουμε τι είναι το κάθε πεδίο κειμένου πάνω σε αυτή την καρτέλα: το «Date» είναι η ημερομηνία της κατάδυσης, το «Dive Site» είναι η περιοχή, το «X» είναι το ποσοστό της μπουκάλας «EAN», το «Psi/Bar START» και το «Psi/Bar END» είναι η πίεση στις μπουκάλες πριν και μετά την κατάδυση, το «Time IN» και το «Time OUT» είναι η ώρα κατάδυσης και ανάδυσης, το «PGO» είναι το επίπεδο αζώτου εάν έχουμε κάνει ήδη μια κατάδυση, το «SI» είναι ο χρόνος ανάμεσα στην 1^η και στη 2^η κατάδυση, το «PGN» είναι το επίπεδο αζώτου λόγω του χρόνου «SI», το «PG» είναι το επίπεδο αζώτου της κατάδυσης, το «DEPTH» είναι το βάθος, το «BOTTOM TIME» είναι ο χρόνος παραμονής στο βυθό, τα «RNT», «ABT» και «TBT» είναι οι χαρακτηριστικοί χρόνοι κατάδυσης και το «Message» είναι οι βασικοί κανόνες για την κατάδυση. Από τα Checkbox είναι οι δύο διαφορετικοί τύποι μπουκάλας και μπορούμε να επιλέξουμε μόνο έναν από τους δύο (δηλαδή, την «AIR» ή την «EAN»). Στην περίπτωση που έχουμε επιλέξει την «EAN» ελευθερώνεται και μπορούμε να γράψουμε το ποσοστό της, το πεδίο κειμένου «X».

Η διαδικασία που ακολουθούμε έχει ως εξής: Γράφουμε την ημερομηνία και πατάμε το κουμπί «Find», εάν δεν έχουμε κάνει άλλη κατάδυση εμφανίζεται στην περιοχή του «DIVE PROFILE» ο αριθμός '1', ενώ εάν είναι η δεύτερη κατάδυση την ίδια μέρα τότε εμφανίζεται ο αριθμός '2' (παραπάνω από 2 καταδύσεις δεν μας αφήνει να κάνουμε). Ύστερα, μπορούμε να συμπληρώσουμε τα υπόλοιπα πεδία κειμένου και να πατήσουμε το κουμπί «Run». Οπότε, εάν έχουμε δώσει όλες τις σωστές τιμές, συμπληρώνονται αυτόματα όλα τα υπόλοιπα πεδία κειμένου που είναι μόνο για ανάγνωση. Τέλος, για να αποθηκεύσουμε όλα αυτά τα δεδομένα της κατάδυσης πατάμε το κουμπί «Save», και μπορούμε να επαναλάβουμε τη διαδικασία που περιγράψαμε.

The screenshot shows the 'LogBookDive' application window. At the top, there's a title bar with the application name and standard window controls. Below the title bar, there are input fields for 'Date' (with a calendar icon), 'Dive Site', and buttons for 'Find', 'Run', and 'Save'. A section titled 'DIVE PROFILE' contains checkboxes for 'AIR' and 'EAN x', and a field for 'X=' with a value of '0'. Below this, there are fields for 'Psi/Bar START', 'Time IN', 'Time OUT', and 'Psi/Bar END'. A graph area shows a depth profile with labels 'PGO', 'SI', 'PGN', and 'PG'. To the right of the graph, there are buttons for 'RNT', '+ABT', and 'TBT'. A 'Message:' label is followed by a large text area. At the bottom, there's a navigation bar with buttons for 'Dive1-2', 'History', 'View', 'Delete', and 'Sync', along with left and right arrow buttons.

Σχήμα 6.2: Καρτέλα «Dive1-2» της εφαρμογής «LogBookDive».

6.3.3 Παρουσίαση των περιεχομένων της τοπικής βάσης δεδομένων

Μια από τις δυνατότητες που διαθέτει επίσης αυτή η εφαρμογή είναι η παρουσίαση των περιεχομένων της τοπικής βάσης δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στη δεύτερη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή και είναι η «History» (Σχήμα 6.3). Σε αυτήν βλέπουμε τα περιεχόμενα της τοπικής βάσης δεδομένων σε ένα πίνακα με στήλες «Kwdikos, OnomaDuth, Hmeromhnia, Dive_Site, Bottle, X, Bar_Start, Time_In, Time_Out, Bar_End, PGO, SI, PGN, PG, Depth, Bot_Time, RNT, ABT, TBT». Στην καρτέλα αυτή επίσης βλέπουμε δύο κουμπιά, το ένα είναι το «Create New DB» (κάνω αριστερά) και το άλλο είναι το «Refresh» (κάτω δεξιά). Το κουμπί «Create New DB» το πατάμε μόνο όταν χρειαστεί να χρησιμοποιήσει την εφαρμογή κάποιος διαφορετικός χρήστης, ενώ το κουμπί «Refresh» πρέπει να πατάμε κάθε φορά που πραγματοποιείται μία αλλαγή στην τοπική βάση και θέλουμε να δούμε τα περιεχόμενά της.

	Kwdikos	Hmeron	Dive_Sit	Bottle	X
▶	6003	9/7/06	Xalkida	AIR	0
	8002	10/7/06	Kalamata	AIR	0
	6002	9/7/06	Xios	EAN	20

Σχήμα 6.3: Καρτέλα «History» της εφαρμογής «LogBookDive».

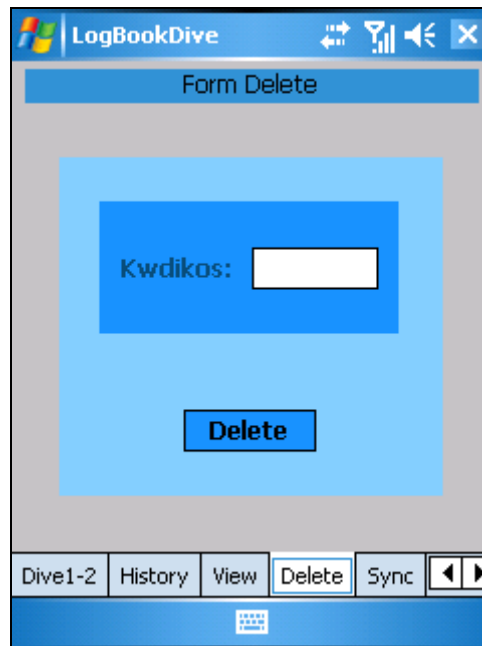
6.3.4 Παρουσίαση μιας υπάρχουσας κατάδυσης

Μια από τις δυνατότητες που διαθέτει επίσης αυτή η εφαρμογή είναι η παρουσίαση μιας υπάρχουσας κατάδυσης από την τοπική βάση δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στη τρίτη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή και είναι η «View» (Σχήμα 6.4). Η καρτέλα αυτή αποτελείται συγκεντρωτικά από 17 πεδία κειμένου, 2 Checkbox και από δύο κουμπιά, το ένα είναι το «OK» και το άλλο δίπλα του είναι το «Clear» (και τα δύο πάνω δεξιά). Στην καρτέλα αυτή μπορούμε να δούμε τα δεδομένα μίας κατάδυσης με βάση το κωδικό της. Στο πεδίο κειμένου «Kwdikos» εισάγουμε τον κωδικό της κατάδυσης, που πιθανόν έχουμε δει πριν από την καρτέλα «History» ή γνωρίζουμε, και στη συνέχεια πατάμε το κουμπί «OK». Ύστερα από αυτό σε όλα τα πεδία κειμένου εμφανίζονται τα δεδομένα της κατάδυσης με το συγκεκριμένο κωδικό. Το τι είναι το κάθε πεδίο κειμένου το περιγράψαμε στην καρτέλα «Dive1-2». Τέλος, το κουμπί «Clear» καθαρίζει όλα τα πεδία κειμένου της καρτέλας «View».

Σχήμα 6.4: Καρτέλα «View» της εφαρμογής «LogBookDive».

6.3.5 Διαγραφή μιας υπάρχουσας κατάδυσης

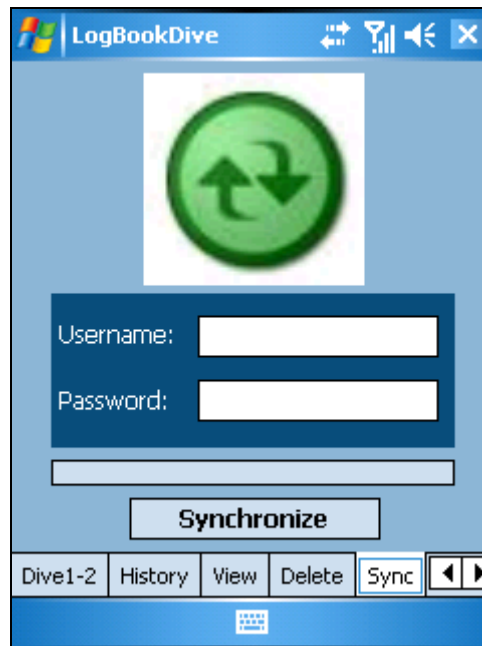
Μια από τις δυνατότητες που διαθέτει επίσης αυτή η εφαρμογή είναι η διαγραφή μιας υπάρχουσας κατάδυσης στην τοπική βάση δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στην τέταρτη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή και είναι η «Delete» (Σχήμα 6.5). Η καρτέλα αυτή αποτελείται από ένα πεδίο κειμένου και από ένα κουμπί το «Delete» (κάτω κέντρο). Στην καρτέλα αυτή μπορούμε να διαγράψουμε τα δεδομένα μίας ήδη υπάρχουσας κατάδυσης με βάση τον κωδικό της. Σε αυτό το πεδίο κειμένου εισάγουμε τον κωδικό της κατάδυσης, που πιθανόν έχουμε δει πριν από την καρτέλα «History» ή γνωρίζουμε, και στη συνέχεια πατάμε το κουμπί «Delete». Ύστερά από αυτό θα έχει διαγραφεί από την τοπική βάση δεδομένων η κατάδυση με το συγκεκριμένο κωδικό.



Σχήμα 6.5: Καρτέλα «Delete» της εφαρμογής «LogBookDive».

6.3.6 Συγχρονισμός με την κεντρική βάση δεδομένων

Μια από τις δυνατότητες που διαθέτει επίσης αυτή η εφαρμογή είναι ο συγχρονισμός της τοπικής βάσης δεδομένων με την απομακρυσμένη κεντρική βάση δεδομένων. Αυτή η δυνατότητα παρουσιάζεται στην πέμπτη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή και είναι η «Sync» (Σχήμα 6.6). Η καρτέλα αυτή αποτελείται από δύο πεδία κειμένου και από ένα κουμπί το «Synchronize» (κάτω κέντρο). Στην καρτέλα αυτή μπορούμε να πραγματοποιήσουμε το συγχρονισμό της τοπικής βάσης δεδομένων με την απομακρυσμένη κεντρική βάση, εφόσον όμως πρώτα έχουν γίνει οι κατάλληλες ρυθμίσεις στην επόμενη καρτέλα «Setting» (Σχήμα 6.7). Στο πρώτο και στο δεύτερο πεδίο κειμένου συμπληρώνουμε αντίστοιχα το «Username» και το «Password», τα οποία θα πρέπει να μας έχουν δοθεί από το διαχειριστή της κεντρικής βάσης δεδομένων, και στη συνέχεια πατάμε το κουμπί «Synchronize». Η διαδικασία του συγχρονισμού πιθανόν να πάρει μερικά λεπτά και όταν ολοκληρωθεί θα έχει γεμίσει η μπάρα που βρίσκεται πάνω από το κουμπί.



Σχήμα 6.6: Καρτέλα «Sync» της εφαρμογής «LogBookDive».

6.3.7 Ρυθμίσεις του συγχρονισμού

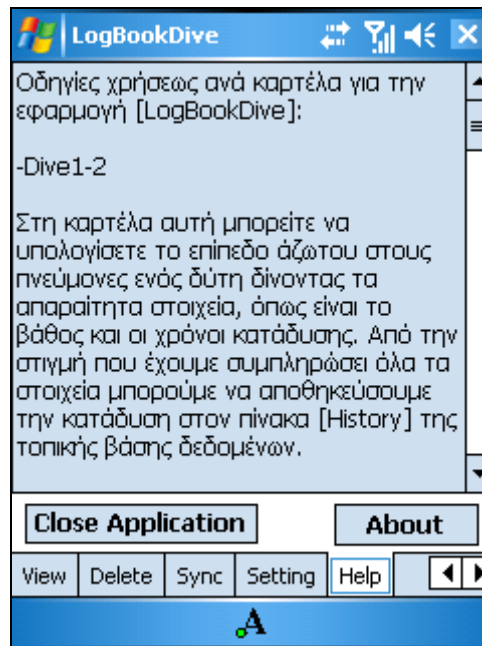
Η έκτη κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή είναι η «Setting» (Σχήμα 6.7). Η καρτέλα αυτή αποτελείται συνολικά από πέντε πεδία κειμένου, χωρίς να διαθέτει τίποτα άλλο. Στην καρτέλα αυτή συμπληρώνουμε τις απαραίτητες ρυθμίσεις που χρειάζονται για την πραγματοποίηση του συγχρονισμού. Το πρώτο πεδίο κειμένου που εμφανίζεται είναι το «InternetUrl» και αποτελεί τη διεύθυνση διαδικτύου στην οποία βρίσκεται η κεντρική βάση δεδομένων. Το επόμενο είναι το «Publisher» και αποτελεί το όνομα του server στον οποίο βρίσκεται η βάση δεδομένων. Το πεδίο κειμένου με το όνομα «PublisherDatabase» αποτελεί το όνομα της κεντρικής βάσης δεδομένων. Επίσης, το πεδίο με όνομα «Publication» αποτελεί το όνομα του Merge Replication που υπάρχει στην κεντρική βάση για τη διαδικασία του συγχρονισμού του πίνακα «RAnakoinwseis». Το τελευταίο πεδίο κειμένου που εμφανίζεται σε αυτή την καρτέλα είναι το «Subscriber» και είναι το όνομα με το οποίο συνδέεται η εφαρμογή στο Replication της βάσης.



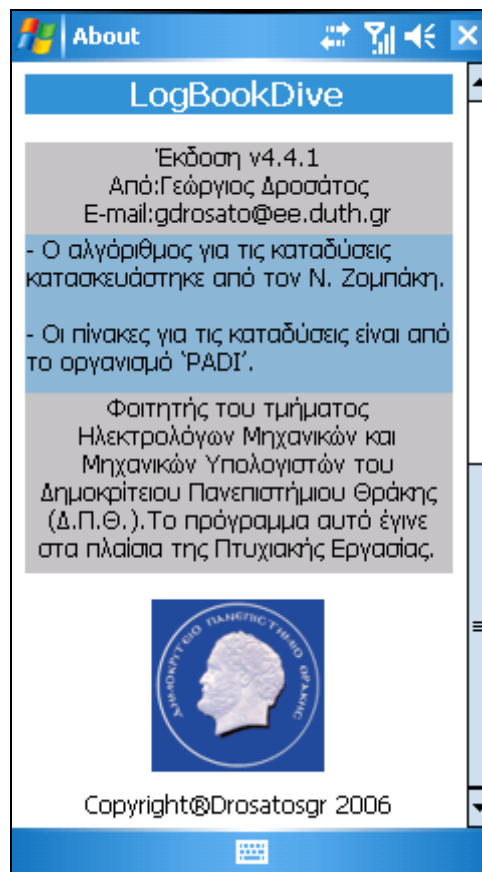
Σχήμα 6.7: Καρτέλα «Setting» της εφαρμογής «LogBookDive».

6.3.8 Οδηγίες της εφαρμογής

Η έβδομη και τελευταία κατά σειρά καρτέλα που εμφανίζεται στην εφαρμογή είναι η «Help» (Σχήμα 6.8). Η καρτέλα αυτή αποτελείται από ένα μεγάλο πεδίο κειμένου και από δύο κουμπιά, το ένα είναι το «Close Application» (κάτω αριστερά) και το άλλο είναι το «About» (κάτω δεξιά). Στο μοναδικό πεδίο κειμένου που διαθέτει αναγράφονται όλες οι απαραίτητες οδηγίες χρήσεως της εφαρμογής. Το κουμπί «Close Application», αυτό που κάνει είναι να τερματίζει την εφαρμογή, λόγω της ιδιοτροπίας που παρουσιάζουν τα Windows Mobile 5.0, πατώντας το «X» να μην κλείνει η εφαρμογή και απλά να γίνεται «Hide». Τέλος, το κουμπί «About» εμφανίζει τη φόρμα του σχήματος 6.9 με πληροφορίες για τον κατασκευαστή.



Σχήμα 6.8: Καρτέλα «Help» της εφαρμογής «LogBookDive».



Σχήμα 6.9: Φόρμα «About» της εφαρμογής «LogBookDive».

6.4 Διαδικασία κατασκευής

Στην ενότητα αυτή θα γίνει μία προσπάθεια να περιγραφούν τα βήματα που πρέπει να ακολουθήσει κάποιος για την κατασκευή αυτής της εφαρμογής. Δεν θα ξαναπούμε όμως όσα έχουν αναφερθεί ήδη στις εφαρμογές των κεφαλαίων 3 & 5, και δεν παρουσιάζουν καμία ιδιαίτερη αλλαγή. Έτσι λοιπόν θα μιλήσουμε μόνο για το πρώτο και το τελευταίο βήμα, δηλαδή για τη δημιουργία του πίνακα και των triggers, αλλά και για την κατασκευή της εφαρμογής. Τα επόμενα βήματα παραμένουν τα ίδια με μερικές μόνο διευκρινίσεις:

- Η δημιουργία του Publication είναι ίδια με αυτή που παρουσιάστηκε στο κεφάλαιο 5 (Ενότητα 5.4.2), με την διαφορά όμως εδώ ότι το φίλτρο χρησιμοποιεί το όνομα του δύτε «OnomaDuth»:

```
SELECT <published_columns> FROM [dbo].[History] WHERE
[OnomaDuth] = SUSER_SNAME()
```

- Όσον αφορά τη ρύθμιση του δικτυακού συγχρονισμού, από τη στιγμή που την έχουμε φτιάξει ήδη μία φορά δεν χρειάζεται να την ξαναδημιουργήσουμε (Κεφ. 3, Ενότητα 3.4.3).
- Στην κατασκευή της τοπικής βάσης δεδομένων και τη δημιουργία Subscription δεν παρουσιάζεται καμία ιδιαίτερη αλλαγή (Κεφ. 3, Ενότητα 3.4.4).

6.4.1 Δημιουργία του πίνακα και των triggers στην κεντρική βάση δεδομένων

Από την στιγμή που υπάρχει η βάση δεδομένων «NEWDB» (που δημιουργήσαμε στο κεφ. 3, ενότητα 3.4.1) μπορούμε πλέον να κατασκευάσουμε τον πίνακα «History». Η κατασκευή του πίνακα γίνεται πολύ απλά με τις εξής SQL εντολές:

```
CREATE TABLE History (Kwdikos INTEGER IDENTITY (1,1) CONSTRAINT
pkKwdikos PRIMARY KEY, OnomaDuth NVarChar (50), Hmeromhnia datetime,
Dive_Site NVarChar (100), Bottle NVarChar (3), X int, Bar_Start int,
Time_In NVarChar (5), Time_Out NVarChar (5), Bar_End int, PGO
NVarChar (1), SI int, PGN NVarChar (1), PG NVarChar (1), Depth int,
Bot_Time int, RNT int, ABT int, TBT int);
```

Τώρα πλέον μπορούμε να φτιάξουμε τους triggers, οι οποίοι ελέγχουν εάν το όνομα του δύτε της κατάδυσης είναι το ίδιο με αυτό του χρήστη (δύτε) που κάνει τις αλλαγές στον πίνακα «History». Εάν είναι το ίδιο, τότε επιτρέπει τις αλλαγές (εισαγωγής και τροποποίησης), ενώ εάν είναι διαφορετικό δεν τις αφήνει να γίνουν. Αυτό γίνεται για να μην μπορεί κάποιος να εισάγει ή να τροποποιήσει μια κατάδυση, όπου ο δύτες δεν είναι ο ίδιος. Οι triggers που χρησιμοποιήθηκαν είναι δύο:

α) Ο trigger «History_Insert_USERS», που όταν γίνεται μια εισαγωγή (Insert) στον πίνακα «History», ελέγχει εάν το όνομα του δύτε (OnomaDuth) της κατάδυσης είναι το ίδιο με αυτό που πραγματοποιεί την εισαγωγή (εάν είναι το ίδιο την αφήνει, εάν όχι τη διαγράφει). Οι SQL εντολές για τη δημιουργία του είναι:

```
CREATE TRIGGER History_Insert_USERS ON History AFTER INSERT
AS
BEGIN
DELETE From History Where History.Kwdikos in (SELECT Kwdikos FROM
inserted)AND NOT(History.OnomaDuth = SUSER_SNAME())
END
```

β) Ο trigger «History_Update_USERS», που όταν γίνεται μια ενημέρωση (Update) στον πίνακα «History», ελέγχει εάν το όνομα του δύτη (OnomaDuth) της κατάδυσης είναι το ίδιο με αυτό που πραγματοποιεί την ενημέρωση (εάν είναι το ίδιο την αφήνει, εάν όχι την επαναφέρει). Οι SQL εντολές για τη δημιουργία του είναι:

```
CREATE TRIGGER History_Update_USERS ON History instead of Update
AS
BEGIN
IF NOT(SUSER_SNAME() IN (SELECT OnomaDuth FROM inserted))

UPDATE History SET OnomaDuth = (SELECT OnomaDuth FROM
deleted),Hmeromhnia = (SELECT Hmeromhnia FROM deleted),Dive_Site=
(SELECT Dive_Site FROM deleted), Bottle= (SELECT Bottle FROM
deleted), X= (SELECT X FROM deleted), Bar_Start= (SELECT Bar_Start
FROM deleted), Time_In= (SELECT Time_In FROM deleted), Time_Out=
(SELECT Time_Out FROM deleted), Bar_End= (SELECT Bar_End FROM
deleted), PGO= (SELECT PGO FROM deleted), SI= (SELECT SI FROM
deleted), PGN= (SELECT PGN FROM deleted), PG= (SELECT PG FROM
deleted), Depth= (SELECT Depth FROM deleted), Bot_Time= (SELECT
Bot_Time FROM deleted), RNT= (SELECT RNT FROM deleted), ABT= (SELECT
ABT FROM deleted), TBT= (SELECT TBT FROM deleted) WHERE
History.Kwdikos in (SELECT Kwdikos FROM deleted)

ELSE

UPDATE History SET Hmeromhnia = (SELECT Hmeromhnia FROM
inserted),Dive_Site= (SELECT Dive_Site FROM inserted), Bottle=
(SELECT Bottle FROM inserted), X= (SELECT X FROM inserted),
Bar_Start= (SELECT Bar_Start FROM inserted), Time_In= (SELECT Time_In
FROM inserted), Time_Out= (SELECT Time_Out FROM inserted), Bar_End=
(SELECT Bar_End FROM inserted), PGO= (SELECT PGO FROM inserted), SI=
(SELECT SI FROM inserted), PGN= (SELECT PGN FROM inserted), PG=
(SELECT PG FROM inserted), Depth= (SELECT Depth FROM inserted),
Bot_Time= (SELECT Bot_Time FROM inserted), RNT= (SELECT RNT FROM
inserted), ABT= (SELECT ABT FROM inserted), TBT= (SELECT TBT FROM
inserted) WHERE History.Kwdikos in (SELECT Kwdikos FROM inserted)

END
```

6.4.2 Κατασκευή Εφαρμογής

Όσον αφορά την κατασκευή της εφαρμογής, η διαδικασία παραμένει ίδια με αυτή που παρουσιάστηκε στο κεφάλαιο 3 και συγκεκριμένα στην ενότητα 3.4.5. Με σημαντικά σημεία, όλα όσα αναφέρθηκαν στο κεφάλαιο 3 ((α), (β) και (γ) της ενότητας 3.4.5), το (α) του κεφαλαίου 5 (Ενότητα 5.4.3), καθώς και ένα επιπλέον σημείο κώδικα που θα παρουσιάσουμε εδώ. Πρέπει να σημειωθεί ότι ολόκληρος ο κώδικας αυτής της εφαρμογής μαζί με το Project του Visual Studio βρίσκονται στο CD που περιλαμβάνεται στο τέλος αυτής της διπλωματικής εργασίας.

Το επιπλέον αυτό σημείο του κώδικα που παρουσιάζει ενδιαφέρον, αφορά την χρήση του κρυπτογραφικού αλγόριθμου SHA, για το έλεγχο των Passwords που χρησιμοποιούνται από τους χρήστες κατά την είσοδό τους στην εφαρμογή. Ο κώδικας αυτός είναι ο εξής:

```
private string[] pass = {"raMdXgB9pMjyViV3HCpz8PhHe6o/QBc7IbUk",
                        "WMDs4IwauPxx4Q1SfFjdMBZ5PZge7tm2OA==",
                        "O62mxtSRvIT9tOtboKpnEHWBfnDgNq9T8kpw",
                        "xQ0kNip2HfO5+sdvVFwKDW9E/LLczJRofA==",
                        "601okHDiPBOYI10S7Nb14kyk4hhYp+Dj8VA=",
                        "197sizas01/oUE19rx6aFO8S7sI5DxpV",
                        "zuvUGak0pDyt4P2xz9W5/PHI+WwCZyh2y01A",
                        "bO7hqPWGjnlyLu6HjPKBjmm+LhFyxqge0w==" };

for (int i = 0; i < pass.Length; i++)
{
    if (SimpleHash.VerifyHash(textBox1.Text, "SHA1", pass[i]))
    {
        .
        .
        .
    }
}
```

6.4.3 Τεχνικά χαρακτηριστικά

Η έκδοση της εφαρμογής αυτής είναι η «LogBookDive» v4.4.1 και δουλεύει σε λειτουργικό σύστημα Windows Mobile 5.0 και συγκεκριμένα για Pocket PC. Για την ανάπτυξη της εφαρμογής αυτής χρειάζονται τα παρακάτω εργαλεία-προγράμματα:

1. Ο Web Server IIS v5.1 των Windows XP Professional SP2.
2. Ο SQL Server 2005 και ο SQL Server 2005 Mobile Edition.
3. Το Visual Studio 2005.
4. Το .NET Compact Framework version 2.0.
5. Το ActiveSync v4.1.
6. Τα Windows Mobile 5.0 SDK για Pocket PC.

Επιπλέον, για την ανάπτυξη αυτής της εφαρμογής έγινε χρήση των ακόλουθων:

- Ο αλγόριθμος των καταδύσεων κατασκευάστηκε από τον Νικόλαο Ζομπάκη (E-mail: nzompaki@ee.duth.gr), φοιτητής του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Δημοκρίτειο Πανεπιστήμιο Θράκης (Δ.Π.Θ.).
- Οι πίνακες για τις καταδύσεις είναι από τον οργανισμό της «PADI».

6.5 Συμπεράσματα

Υστέρα από την παρουσίαση της εφαρμογής «LogBookDive», μπορούμε πλέον τώρα να βγάλουμε τα συμπεράσματά μας από την ανάπτυξη αυτή. Το σίγουρο πάντως είναι ότι και εδώ ισχύουν τα συμπεράσματα που είχαμε βγάλει για την

εφαρμογή του κεφαλαίου 3 (Ενότητα 3.5), με τη διαφορά ότι έχουν προκύψει και μερικά άλλα.

Το επιπλέον χαρακτηριστικό της εφαρμογής αυτής είναι η ασφάλεια (ένα τμήμα της είναι ίδιο με αυτό της εφαρμογής του κεφαλαίου 5), που σχετίζεται, με την ελεγχόμενη χρήση της εφαρμογής, με την έννοια του προσωπικού απορρήτου, αλλά και τη σωστή λειτουργία της κεντρικής βάσης δεδομένων. Η ελεγχόμενη χρήση της εφαρμογής, αφορά τη χρησιμοποίηση του κρυπτογραφικού αλγόριθμου SHA για τον έλεγχο των Passwords κατά την είσοδο στην εφαρμογή, η οποία εξασφαλίζει ότι η εφαρμογή δεν θα μπορέσει να χρησιμοποιηθεί από τον καθένα που θα πέσει στα χέρια του. Από την άλλη η διασφάλιση του προσωπικού απορρήτου επιτυγχάνεται με τη χρήση δυναμικού φίλτρου στον πίνακα «History» (ρύθμιση του Publication). Η χρήση του φίλτρου επιτρέπει στους διάφορες χρήστες (δύτες) που χρησιμοποιούν την εφαρμογή να βλέπουν και να διαχειρίζονται μόνο τις καταδύσεις που έχουν κάνει οι ίδιοι, χωρίς να μπορούν ούτε καν να δουν τις καταδύσεις των άλλων χρηστών (δυτών). Τέλος, η διασφάλιση της σωστής λειτουργίας της κεντρικής βάσης δεδομένων επιτυγχάνεται με την χρήση triggers. Οι triggers αυτοί εξυπηρετούν την αντιμετώπιση διάφορων παραβιάσεων που μπορεί να γίνονται από τους χρήστες, όπως είναι η εισαγωγή ή ενημέρωση καταδύσεων όπου το όνομα του δύτη (OnomaDuth) δεν είναι το ίδιο με αυτό του χρήστη (Username).

7. Γενικά Συμπεράσματα

Στο τελευταίο αυτό κεφάλαιο, θα προσπαθήσουμε να αναπτύξουμε με λίγα λόγια, όλα όσα μπορεί να αποκομίσει κάποιος από την ενασχόληση του με το αντικείμενο αυτής της διπλωματικής εργασίας. Επιπλέον, θα αναφέρουμε τι τελικά καταφέραμε, αλλά και τι δεν καταφέραμε να κάνουμε, ύστερα από την ανάπτυξη αυτής της τεχνολογίας και πιο συγκεκριμένα των τεσσάρων αυτών εφαρμογών. Τέλος, να αναφερθούν και κάποιες πιθανές επεκτάσεις αυτής της τεχνολογίας ακόμη και σε κάποιους άλλους τομείς.

Ύστερα από την ενασχόληση με το αντικείμενο αυτής της διπλωματικής εργασίας, αυτά που μπορεί να αποκομίσει κάποιος είναι πάρα πολλά και σε μεγάλη μάλιστα κλίμακα. Επειδή, για να δουλέψουν αυτές οι εφαρμογές δεν χρειάζεται να ξέρεις μόνο προγραμματισμό, αλλά και να ασχοληθείς και με πολλά άλλα θέματα, όπως είναι οι βάσεις δεδομένων, ο Web Server, αλλά και διάφορα άλλα θέματα που αφορούν την ασφάλεια των δεδομένων. Έτσι λοιπόν, το πρώτο πράγμα που θα γνωρίσει κάποιος, λόγω του γεγονότος ότι οι εφαρμογές αυτές αφορούν φορητές πλατφόρμες (συσκευές), είναι να αποκτήσει μία ιδιαίτερη εξοικείωση με το λειτουργικό σύστημα των συσκευών αυτών, που είναι τα Windows Mobile 5.0.

Από την πλευρά τώρα του προγραμματισμού, εκτός του ότι μπορεί να μάθει να χρησιμοποιεί την γλώσσα προγραμματισμού C#, του δίνεται επιπλέον η δυνατότητα να ασχοληθεί και με το γραφικό περιβάλλον της γλώσσας (δηλαδή, το λεγόμενο GUI των εφαρμογών). Επιπλέον, όλα όσα αφορούν την C#, πραγματοποιούνται στο Visual Studio 2005, που αποτελεί ένα εργαλείο ανάπτυξης. Με αποτέλεσμα, να μάθει να χρησιμοποιεί αυτό το περιβάλλον ανάπτυξης, κάτι το οποίο είναι πολύ σημαντικό γιατί εκτός από την C#, μπορεί να προγραμματίσει και σε άλλες γλώσσες όπως είναι η Basic, η C++, και η J#. Τέλος, με το Visual Studio του παρέχονται ένα πλήθος εργαλείων, όπως είναι και ο Emulator του Pocket PC με τα Windows Mobile 5.0, στον οποίο μπορεί να δοκιμάσει όλες τις εφαρμογές που φτιάχνει για Pocket PC, χωρίς να χρειάζεται την ύπαρξη ενός πραγματικού.

Από την πλευρά τώρα των βάσεων δεδομένων, μπορεί να μάθει τις βασικές εντολές SQL που χρειάζονται για τον ορισμό και τη διαχείριση των δεδομένων, αλλά και πιο υψηλού επιπέδου SQL εντολές (Trigger) που εκτελούνται αυτόματα, κάνοντας κάποια ενέργεια, όταν πραγματοποιούνται συγκεκριμένες αλλαγές στα δεδομένα. Εξαιτίας, της φύσεως των εφαρμογών θα γνωρίσει πως δουλεύουν ο SQL Server 2005, καθώς επίσης και ο SQL Server Mobile 2005. Με αυτό τον τρόπο θα ασχοληθεί με την εγκατάσταση, τις διάφορες ρυθμίσεις, και τη διαχείριση (administration) του SQL Server. Όσο αφορά, την διαχείριση του SQL Server, θα πρέπει να τονίσουμε ότι θα μάθει να χειρίζεται θέματα όπως τη δημιουργία μίας καινούριας βάσης δεδομένων, τη δημιουργία Replication, τη δημιουργία λογαριασμών (με διάφορα δικαιώματα) κ.τ.λ., με αποτέλεσμα δηλαδή, να μπορεί χαρακτηριστεί επάξια ως DBA (Database Administrator). Το πιο σημαντικό από τα παραπάνω θέματα διαχείρισης που αναφέρθηκαν είναι το Replication, γιατί είναι αυτό που παίζει το σημαντικότερο ρόλο και στις τέσσερις εφαρμογές που αναπτύχθηκαν. Με κύριο στόχο του, τον συγχρονισμό της τοπικής βάσης δεδομένων (SQL Server Mobile 2005) που βρίσκεται στο Pocket PC, με την κεντρική βάση δεδομένων (SQL Server 2005) που βρίσκεται σε κάποιον κεντρικό υπολογιστή.

Από την πλευρά τώρα του Web Server, μπορεί να μάθει πως δουλεύει ένας διακομιστής διαδικτύου, αλλά βέβαια και όλες εκείνες οι ρυθμίσεις που χρειάζονται για την σωστή λειτουργία του. Πιο συγκεκριμένα θα γνωρίσει των IIS, που είναι ένας

διακομιστής διαδικτύου, που περιέχεται στις server εκδόσεις του λειτουργικού συστήματος Windows. Επιπλέον, θα γνωρίσει πως ο IIS μεσολαβεί στην επικοινωνία-συγχρονισμό ανάμεσα στην τοπική βάση δεδομένων της φορητής πλατφόρμας και της κεντρικής βάσης δεδομένων. Τέλος, επειδή η επικοινωνία αυτή πρέπει να είναι ασφαλής (HTTPS), θα γνωρίσει επίσης πως μπορεί να φτιάχνει self-sign certificates για αυτή την δουλειά, αλλά και πως να τα χρησιμοποιεί στην πράξη.

Από την πλευρά τώρα της ασφάλειας δεδομένων, θα ασχοληθεί όπως ήδη αναφέρθηκε με certificates στο επίπεδο επικοινωνίας, αλλά και με διάφορα θέματα κρυπτογραφίας των δεδομένων. Όσον αφορά την κρυπτογραφία δεδομένων, θα μάθει κυρίως πώς να χρησιμοποιεί προγραμματιστικά τους κρυπτογραφικούς αλγόριθμους AES (Rijndael) και Hash. Ο AES έχει ως στόχο να κρυπτογραφεί τα δεδομένα που πρόκειται να αποθηκευτούν στην βάση δεδομένων, με την χρήση βέβαια κάποιου password, με αποτέλεσμα να μην μπορεί κανείς να τα διαβάσει, εκτός από εκείνον που γνωρίζει το password. Από την άλλη, η κύρια χρήση του Hash είναι η δημιουργία password, ώστε να μπορεί να ελέγχεται ποιοι μπορούν να χρησιμοποιήσουν την εφαρμογή, χωρίς να υπάρχει κίνδυνος να βρεθούν τα password από τον source code της εφαρμογής.

Ο αρχικός στόχος αυτής της διπλωματικής εργασίας ήταν η κατασκευή κάποιας εφαρμογής που να δουλεύει σε Pocket PC, σε κύριο στόχο το συγχρονισμό της τοπικής βάσης δεδομένων με κάποια άλλη κεντρική βάση δεδομένων, με χρήση Replication. Ο αρχικός αυτός στόχος, όχι απλά επιτεύχθηκε, αλλά και επεκτάθηκε με την κατασκευή όχι μόνο μιας εφαρμογής αλλά συνολικά τεσσάρων εφαρμογών. Επιπλέον, μάλιστα όσο αφορά το Replication, έγιναν και αρκετές δοκιμές με διάφορες ρυθμίσεις του, που παρουσιάζονται άλλωστε και στις εφαρμογές των προηγούμενων κεφαλαίων. Από την άλλη, συναντήσαμε αρκετά προβλήματα όσον αφορά το .NET Compact Framework (βιβλιοθήκες της C#), επειδή δεν υπήρχε μεγάλη συμβατότητα με το κανονικό .NET Framework, με αποτέλεσμα να λείπουν πάρα πολλές συναρτήσεις από το πρώτο. Έτσι λοιπόν, αναγκαστήκαμε να φτιάξουμε, σε κάποιες περιπτώσεις, κάποιες δικές μας συναρτήσεις, που πιθανόν να μην είναι και τόσο καλές με τις κανονικές (βέβαια αυτό επίσημα δεν μπορούμε να το ξέρουμε). Επιπλέον, θέλαμε να χρησιμοποιήσουμε εκτός από το ασφαλές κανάλι επικοινωνίας HTTPS, και πιστοποιητικά (client certificates) που θα πιστοποιούν την αυθεντικότητα των διάφορων χρηστών, κάτι όμως που τελικά δεν καταφέραμε να επιτύχουμε σε συνδυασμό με τις εφαρμογές, χωρίς να μπορούμε να ξέρουμε τον ακριβή λόγο (πιθανόν ακόμη και να μην υποστηρίζεται από την συγκεκριμένη μορφή Replication). Το καταφέραμε όμως να δουλέψει, τουλάχιστον στον browser της φορητής πλατφόρμας. Τέλος, εάν και καταφέραμε συγχρονισμό της τοπικής βάσης δεδομένων με την κεντρική, μέσω wireless δικτύου και ενσύρματα μέσω USB, δεν καταφέραμε ωστόσο συγχρονισμό μέσω GPRS (Cosmote), και πάλι χωρίς να ξέρουμε τον λόγο (πιθανόν να οφείλεται στον παροχέα της κινητής τηλεφωνίας).

Τέλος, κάποιες πιθανές επεκτάσεις που θα μπορούσαν να γίνουν αφορούν κυρίως στο να επεκταθεί αυτή η τεχνολογία και σε κάποιες άλλες φορητές συσκευές. Τέτοιες συσκευές, μπορεί να είναι κάποια κινητά τηλέφωνα, όπως τα Smartphones με λειτουργικό σύστημα Windows Mobile 5.0 και Symbian. Θα πρέπει να σημειωθεί εδώ ότι πολύ εύκολα θα μπορούσαν να φτιαχτούν εφαρμογές και για τα Smartphones με Windows Mobile 5.0, η μόνη ουσιαστικά διαφορά με τις εφαρμογές της παρούσας διπλωματικής εργασίας είναι μόνο το γραφικό τους περιβάλλον και τίποτα άλλο, γιατί έχουν πιο μικρή οθόνη και μάλιστα όχι touch screen. Από την άλλη, τα Smartphones που έχουν λειτουργικό σύστημα Symbian, θα μπορούσε και σε αυτά να φτιαχτούν κάποιες εφαρμογές, είτε χρησιμοποιώντας κάποια αντίστοιχα εργαλεία που υπάρχουν

για Symbian, είτε πιθανόν κάποιες μελλοντικές ενημερώσεις της Microsoft για αυτό το λειτουργικό σύστημα.

Παράρτημα

A. Βήματα κατασκευής μιας SQL φορητής εφαρμογής με την βοήθεια του Visual Studio 2005 και του SQL Server 2005

Σε αυτό το παράρτημα, θα δούμε πως χρησιμοποιείται ο SQL Server 2005 Mobile Edition για το συγχρονισμό δεδομένων μεταξύ μιας φορητής συσκευής με λειτουργικό σύστημα με Windows Mobile (CE) και μιας κεντρικής βάσης δεδομένων του SQL Server 2005. Πιο συγκεκριμένα θα δούμε πως γίνεται η εγκατάσταση αλλά και οι διάφορες ρυθμίσεις του SQL Server 2005, του Web Server IIS (Internet Information Services) και κάποιας εφαρμογής του .NET Compact Framework. Όταν ολοκληρώσετε αυτή την διαδικασία, θα έχετε κατασκευάσει μια φορητή εφαρμογή, η οποία θα συγχρονίζει τα τοπικά – φορητά δεδομένα με μια κεντρική βάση δεδομένων. Το αντικείμενο αυτής της ανάπτυξης είναι η κατασκευή μιας φορητής εφαρμογής, με την χρήση του Visual Studio 2005, που θα κρατά τα τοπικά δεδομένα και θα πραγματοποιεί το συγχρονισμό τους με μια κεντρική βάση δεδομένων του SQL Server 2005 [29]. Τα βήματα που απαιτούνται για την κατασκευή αυτής της φορητής εφαρμογής με την χρήση του Visual Studio 2005 αλλά και το συγχρονισμό των δεδομένων με τον SQL Server είναι:

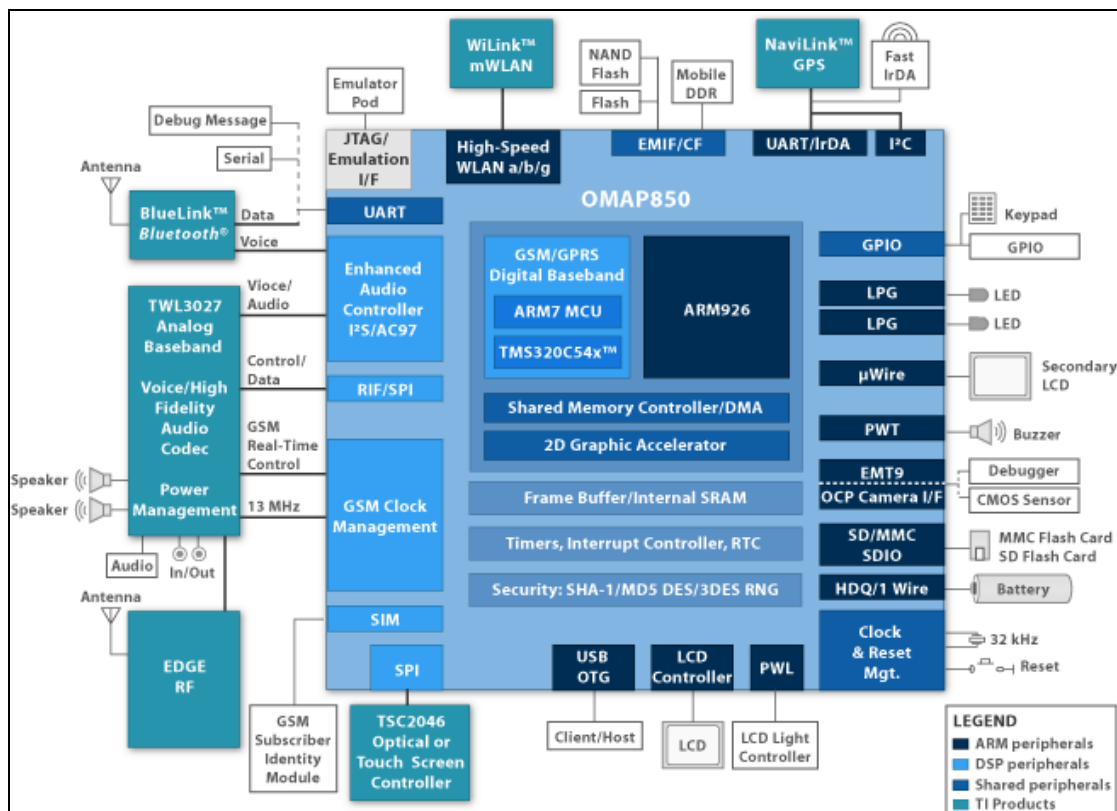
- Εγκατάσταση του SQL Mobile Server.
- Δημιουργία μιας βάσης δεδομένων και ενός Publication στον SQL Server 2005.
- Ρύθμιση του SQL Mobile Server.
- Χρησιμοποίηση του SQL Management Studio.
- Δημιουργία ενός Subscription.
- Χρησιμοποίηση του Visual Studio Server Explorer για σύνδεση αλλά και περιήγηση στα SQL Mobile δεδομένα.
- Ανάπτυξη μιας φορητής εφαρμογής χρησιμοποιώντας το Visual Studio 2005.

Όλα αυτά τα βήματα, αλλά και γενικά όσες πληροφορίες χρειάζονται για την ανάπτυξη αυτής της εφαρμογής βρίσκονται σε ηλεκτρονική μορφή (PDF) με τίτλο «Step_by_Step_Developing_a_SQL_Mobile_Application.pdf» στο CD που περιλαμβάνεται στο τέλος αυτής της διπλωματικής εργασίας. Ουσιαστικά, αυτή η ηλεκτρονική μορφή των βημάτων για την κατασκευή μιας φορητής εφαρμογής, προέρχεται από την online βιβλιοθήκη MSDN της Microsoft, στην οποία όμως γίνονται συχνά αλλαγές και τροποποιήσεις [29]. Η ηλεκτρονική διεύθυνση των βημάτων αυτών στο MSDN είναι:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnppcgen/html/med302_msdn_sql_mobile.asp

B. Επεξεργαστής OMAP850

Ο OMAP850 (Σχήμα B.1) είναι ένα ενιαίο chip της Texas Instruments, που ενσωματώνει τον πυρήνα ARM926EJ-S™ για την εκτέλεση εφαρμογών και το ψηφιακό modem βασικής ζώνης (baseband) TI's EDGE. Ο OMAP850 εκτός από τα βασικά χαρακτηριστικά του, περιλαμβάνει επιπλέον υποστήριξη για 128Mb ή 256Mb stack mobile SDRAM. Αυτό καθιστά τον OMAP850 κατάλληλο για περιορισμένα συστήματα και μικρές φορητές συσκευές. Επίσης, καταναλώνει λιγότερη ενέργεια σε σχέση με τις παραδοσιακές διαμορφώσεις εξωτερικών μνημών. Με αυτό τον τρόπο ευελιξίας, οι κατασκευαστές φορητών συσκευών μπορούν να μειώσουν περαιτέρω το μέγεθος των smartphones της επόμενης γενιάς και των ασύρματων handsets. Επιπλέον, εκτός από την εξοικονόμηση χώρου, η stack SDRAM προσφέρει επιπρόσθετα και μειωμένη κατανάλωση ισχύος (σημαντική παράμετρος για τους σχεδιαστές φορητών συσκευών) [30].



Σχήμα B.1: Μπλοκ διάγραμμα του επεξεργαστή OMAP850.

Χαρακτηριστικά του OMAP850:

Low-Power, High-Performance CMOS Technology

- Low-voltage 130 nm technology
- 1.1 - 1.5V cores, 1.8 - 2.75V IO
- Extremely low power consumption: less than 10 μ A in standby mode
- Split power supplies for application processing, digital baseband and real-time clock enable precise control over power consumption

- Optimized clocking and power management: Only two clocks required at 13 MHz and 32 kHz

ARM926TEJ Core Subsystem

- ARM926EJ-S V5 architecture up to 200 MHz (maximum frequency)
- 16 kB I-cache, 8 kB D-cache
- Java acceleration in hardware
- Multimedia instruction set architecture (ISA) extension

EDGE Digital Baseband Subsystem

- 384 K-bytes internal SRAM
- E-OTD and TTY support
- Quad vocoder with EFR, FR, HR, AMR
- GSM ultra-low power device (ULPD)
- SIM interface

Application Subsystem

- Supports all leading operating systems
- DMA with 4 physical and 17 logical channels and a dedicated 2D graphics engine
- Programmable GPIO keyboard interface
- 54-Mbps WLAN interface
- Security acceleration in hardware:
 - Secure bootloader
 - 48 kB of secure ROM
 - 16 kB of secure RAM
 - Hardware acceleration for security standards and random number generator
 - Unique die ID cell
 - Third-party Security software library
- Enhanced audio controller (EAC)
- Comprehensive memory controller for interfaces to:
 - 128 MB of mobile SDRAM
 - 256 MB Flash
 - NAND Flash controllers
 - 1.6 Mb ISRAM
- SD/MMC/SDIO interface
- Enhanced Trace Module for debug
- LCD controller
- uWire
- SPI
- 1-wire and HDQ interface

- Bluetooth data/audio interface
- USB On-the-Go
- Two high-speed 3.68 MHz UARTs
- Fast IrDA (FIR)
- Two 32-bit timers
- Parallel camera port
- Programmable three-color LED pulse generation
- I²C master/slave controller
- SmartCard interface

289-ball, 12 mm x 12 mm MicroStar BGA™ Package

C. Βοηθητικά εργαλεία

C.1 Δημιουργία Certificates με χρήση του OpenSSL

Το OpenSSL είναι μια υλοποίηση ανοιχτού κώδικα (open source) για τα πρωτόκολλα SSL και TLS. Η βιβλιοθήκη (γραμμένη σε γλώσσα προγραμματισμού C) του υλοποιεί βασικές κρυπτογραφικές συναρτήσεις και παρέχει για την χρήση τους διάφορα εργαλεία. Οι εκδόσεις του OpenSSL είναι περισσότερο διαθέσιμες για λειτουργικά συστήματα Unix (π.χ. Solaris, Linux, MAC OS X, BSD, κ.τ.λ.), και επίσης για τα Windows της Microsoft. Το OpenSSL βασίζεται στο SSLeay, που κατασκευάστηκε από τον Eric Young και Tim Hudson, η ανάπτυξη του οποίου τελείωσε ανεπίσημα γύρω στο Δεκέμβριο του 1998, όταν ο Tim και Eric δούλευαν για το RSA Security [31].

Παρακάτω, ακολουθούν οδηγίες για την κατασκευή certificates (πιστοποιητικών) τόσο για τον Web Server IIS (Server Certificates), όσο και για Client Certificates [32]. Θα πρέπει να σημειωθεί ότι οι οδηγίες αυτές είναι για λειτουργικό σύστημα Windows XP SP2 και για την Windows έκδοση του OpenSSL v0.9.8a. Τα βήματα (περιλαμβάνονται επίσης στο CD που βρίσκεται στο τέλος αυτής της διπλωματικής εργασίας) που πρέπει να γίνουν σε Command Line είναι:

Εγκατάσταση και ρύθμιση του OpenSSL toolkit

1. Κατέβασε το OpenSSL από τη διεύθυνση <http://www.openssl.org> και εγκατέστησε το με τις προεπιλεγμένες ρυθμίσεις. Θα πρέπει το OpenSSL να έχει εγκατασταθεί στο C:\OpenSSL.
2. Πρόσθεσε το C:\OpenSSL\bin στο system path (Control Panel, System, Advanced, Environment Variables, System Variables) του υπολογιστή σου. Αυτό βέβαια δεν είναι απαραίτητο, αλλά γίνεται για λόγους ευκολίας.
3. Δημιούργησε ένα φάκελο για τις διάφορες εργασίες. Στην προκειμένη περίπτωση θα χρησιμοποιήσουμε το C:\ssl ως φάκελο εργασίας.
4. Κατέβασε το αρχείο openssl.conf (Από: <http://www.dylanbeattie.net/docs/openssl.conf>) και βάλτο στο φάκελο εργασίας C:\ssl.
5. Κατασκεύασε τους φακέλους που χρειάζονται για το OpenSSL:

```
C:\ssl>md keys
```

```
C:\ssl>md requests
```

```
C:\ssl>md certs
```

```
C:\ssl>md client
```

6. Δημιούργησε ένα άδειο αρχείο database.txt. Αυτό Command γίνεται ως εξής:

```
c:\ssl>copy con database.txt
^Z
```

7. Δημιούργησε ένα σειριακό αριθμό (serial number) σε ένα αρχείο serial.txt για το Server Certificate. Αυτό γίνεται ως εξής:

```
C:\ssl>copy con serial.txt
01
^Z
```

8. Επίσης, δημιούργησε ένα αρχείο (ca.srl) με τον σειριακό αριθμό του Client Certificate. Αυτό γίνεται ως εξής:

```
C:\ssl>echo 02 > certs/ca.srl
```

Κατασκευή ενός Certificate Authority (CA)

1. Πρώτα, θα δημιουργήσουμε ένα 1024-bit ιδιωτικό κλειδί για τον CA:

```
C:\ssl>C:\OpenSSL\bin\openssl.exe genrsa -des3 -out keys/ca.key 1024

# Loading 'screen' into random state - done
# warning, not much extra random data, consider using the -rand option
# Generating RSA private key, 1024 bit long modulus
# .....++++++
# .....++++++
# e is 65537 (0x10001)
# Enter PEM pass phrase: - choose a memorable pass phrase to use for this
key
# Verifying password - Enter PEM pass phrase: - type your pass phrase
again for verification
```

2. Μετά, θα δημιουργήσουμε ένα master certificate βασισμένο σε αυτό το κλειδί, για να χρησιμοποιείται όταν υπογράφεται ένα άλλο certificates:

```
C:\ssl>C:\OpenSSL\bin\openssl.exe req -config openssl.conf -new -x509 -days
1001 -key keys/ca.key -out certs/ca.cer

# Using configuration from openssl.conf
# Enter PEM pass phrase: - type your passphrase here.
# You are about to be asked to enter information that will be incorporated
# into your certificate request.
# What you are about to enter is what is called a Distinguished Name or a
DN.
# There are quite a few fields but you can leave some blank
# For some fields there will be a default value,
# If you enter '.', the field will be left blank.
# -----
```

```
# Country Name (2 letter code) []:GB
# State or Province Name (full name) []:Hampshire
# Locality Name (eg, city) []:Southampton
# Organization Name (eg, company) []:dylanbeattie.net
# Organizational Unit Name (eg, section) []:
# Common Name (eg, your websites domain name) []:ssl.dylanbeattie.net
# Email Address []:ssl@dylanbeattie.net
```

3. Εάν θέλεις να αντιγράψεις το ca.cer σε ca.pem (για Unix OS), τότε πρέπει να αλλάξεις το όνομα του αρχείου και να επεξεργαστείς το αρχείο αλλάζοντας την φράση "CERTIFICATE" με την "TRUSTED CERTIFICATE".

4. Ακόμη, μπορείς να μετατρέψεις ένα certificate σε μορφή DER ως εξής:

```
C:\ssl>C:\OpenSSL\bin\openssl.exe x509 -in certs/ca.pem -outform DER -out
certs/ca.der
```

5. Τέλος, μπορείς να εισάγεις το CA certificate (ca.der) στο trusted root certificates του υπολογιστή σου, ώστε να είναι αποδεκτή η εκδοτική αρχή των certificates.

Δημιουργία ενός IIS Certificate Request

1. Αντέγραψε το αρχείο certreq.txt στο φάκελο C:\ssl\requests.

2. Υπογραφή του request:

```
C:\ssl>C:\OpenSSL\bin\openssl.exe ca -policy policy_anything -config
openssl.conf -cert certs/ca.cer -in requests/certreq.txt -keyfile keys/ca.key -
days 360 -out certs/iis.cer
```

```
# Using configuration from openssl.conf
# Loading 'screen' into random state - done
# Enter PEM pass phrase:
# Check that the request matches the signature
# Signature ok
# The Subjects Distinguished Name is as follows
# commonName          :PRINTABLE:'myCommonName'
# organizationalUnitName:PRINTABLE:'myOrganisationalUnit'
# organizationName     :PRINTABLE:'myOrganisation'
# localityName         :PRINTABLE:'myLocality'
# stateOrProvinceName  :PRINTABLE:'myProvince'
# countryName          :PRINTABLE:'GB'
# Certificate is to be certified until Feb  2 01:13:14 2004 GMT (360 days)
# Sign the certificate? [y/n]:y
#
#
# 1 out of 1 certificate requests certified, commit? [y/n]y
# Write out database with 1 new entries
# Data Base Updated
```


3. Μετατροπή του υπογεγραμμένου certificate σε μορφή x509 για να χρησιμοποιηθεί στον IIS:

```
C:\ssl>C:\OpenSSL\bin\openssl.exe x509 -in certs/iis.cer -out  
certs/iisx509.cer
```

4. Ύστερα από αυτά το νέο certificate βρίσκεται στο φάκελο `c:\ssl\certs\iisx509.cer` και είναι έτοιμο να εγκατασταθεί στον IIS.

Δημιουργία του Client Certificate

1. Δημιουργία ενός client certificate request (ερωτήματος):

```
C:\ssl>C:\OpenSSL\bin\openssl.exe req -new -newkey rsa:1024 -nodes -out  
client/client1.req -keyout client/client1.key
```

2. Υπογραφή του client certificate από την CA.

```
C:\ssl>C:\OpenSSL\bin\openssl.exe x509 -CA certs/ca.pem -CAkey  
keys/ca.key -CAserial certs/ca.srl -req -in client/client1.req -out  
client/client1.pem -days 365
```

3. Δημιουργία ενός certificate μορφής PKCS12:

```
C:\ssl>C:\OpenSSL\bin\openssl.exe pkcs12 -export -clcerts -in  
client/client1.pem -inkey client/client1.key -out client/client1.p12 -name  
"George Drosatos"
```

4. Τέλος, εισάγουμε το αρχείο PKCS12 στον web browser και μπορούμε πλέον να χρησιμοποιήσουμε το client certificate.
5. Επαναλάβετε τα βήματα 1-3, όσες φορές απαιτούνται.

C.2 DynDNS

Το DynDNS (Dynamic - Δυναμικό DNS) είναι ένα σύστημα που επιτρέπει στα domain name (όνομα περιοχής) που φυλάσσονται σε έναν κεντρικό name server να ενημερώνονται σε πραγματικό χρόνο (real time). Η πιο κοινή χρήση για αυτό είναι να επιτρέπει σε ένα δικτυακό domain name που αντιστοιχεί σε κάποιον υπολογιστή να αλλάζει δυναμικά την IP διεύθυνσή του. Αυτό δίνει την δυνατότητα από πολλές περιοχές (sites) στο διαδίκτυο να συνδέονται σε έναν απομακρυσμένο υπολογιστή, χωρίς να χρειάζεται να γνωρίζουν την IP διεύθυνση του. Η δυνατότητα αυτή επιτυγχάνεται με το να τρέχει ένα πρόγραμμα (λογισμικό) του server στον τοπικό υπολογιστή που έχει την δυναμική IP διεύθυνση [33].

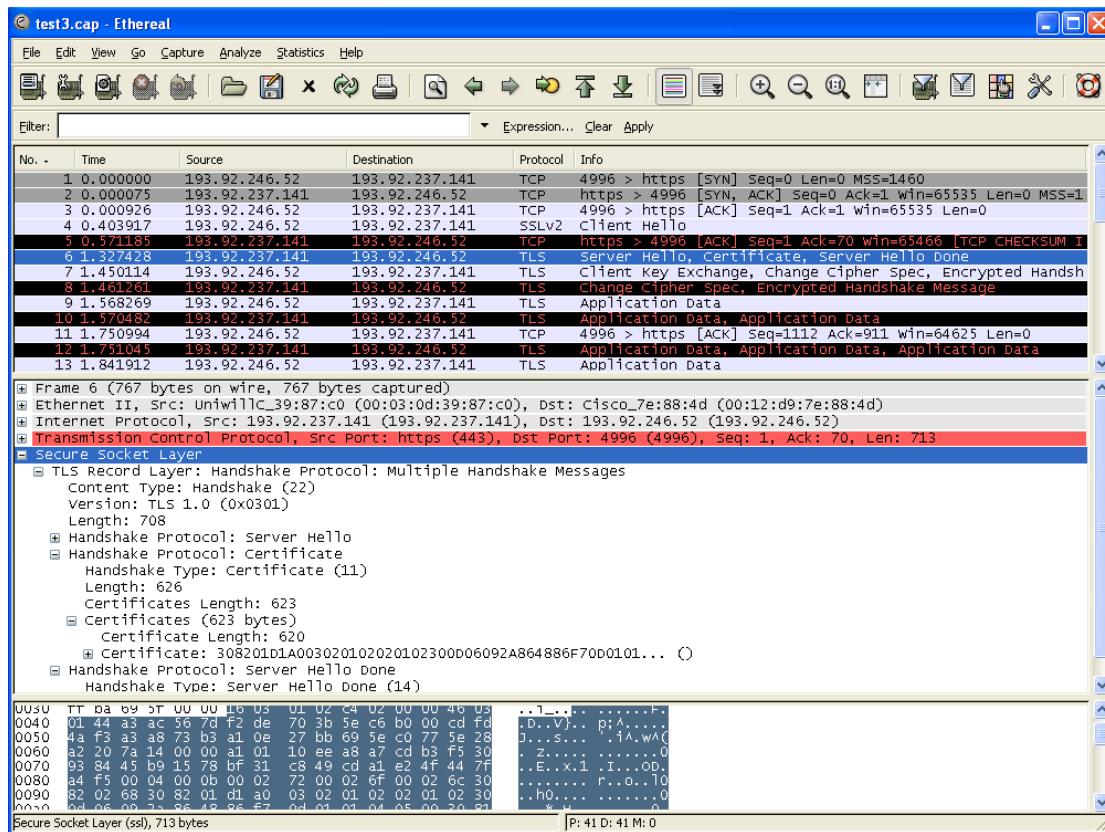
Η υπηρεσία του δυναμικού DNS παρέχεται σε μεγάλη κλίμακα από διάφορες υπηρεσίες DNS hosting, οι οποίες διατηρούν τις τρέχουσες διευθύνσεις σε μια βάση δεδομένων και παρέχουν ένα «client» πρόγραμμα στο χρήστη, που θα στέλνει στον server τις αλλαγές που γίνονται στην διεύθυνση IP. Έτσι λοιπόν, το «client» πρόγραμμα κάθε μερικά λεπτά αποστέλλει στον DNS server τις πιθανές αλλαγές που μπορεί να έχουν συμβεί στην διεύθυνση IP.

Αυτή η δυνατότητα του δυναμικού DNS μπορεί κανείς να την αποκτήσει δημιουργώντας έναν λογαριασμό στην σελίδα www.dyndns.com και χρησιμοποιώντας το «client» πρόγραμμα «DynDNS Updater v3.0» που παρέχεται και πάλι από το ίδιο site. Τέλος, θα πρέπει να σημειώσουμε ότι από την στιγμή που έχουμε κάνει κάποιο λογαριασμό, μπορούμε πλέον να επιλέξουμε και το domain name που θα χρησιμοποιήσουμε για την αντιστοίχιση με τη δυναμική διεύθυνση IP [33].

C.3 Ethereal (Sniffer)

Το Ethereal είναι ένα λογισμικό ανάλυσης πρωτοκόλλων, ή μια εφαρμογή «packet sniffer», που χρησιμοποιείται για την ανίχνευση λαθών σε δίκτυα, την ανάλυση, την ανάπτυξη λογισμικού και πρωτοκόλλων, και την εκπαίδευση. Έχει όλα τα τυποποιημένα χαρακτηριστικά γνωρίσματα ενός λογισμικού ανάλυσης πρωτοκόλλων. Οι λειτουργίες του Ethereal είναι παρόμοιες με του tcpdump, αλλά έχει επιπλέον γραφικό περιβάλλον, με περισσότερες μάλιστα δυνατότητες που αφορούν κυρίως την ταξινόμηση και το φιλτράρισμα των αποτελεσμάτων (πακέτων). Πιο συγκεκριμένα, παρέχει τη δυνατότητα στο χρήστη να βλέπει όλη την κυκλοφορία (traffic) δεδομένων μέσα στο δίκτυο (συνήθως δίκτυο Ethernet), με την τοποθέτηση στην κάρτα δικτύου του υπολογιστή ενός μηχανισμού παρακολούθησης των εισερχόμενων και εξερχόμενων πακέτων (δεδομένων) [34].

Έτσι λοιπόν με αυτό το λογισμικό μας δίνεται η δυνατότητα να παρακολουθήσουμε συγκεκριμένα πρωτόκολλα επικοινωνίας (π.χ. HTTPS, TLS) και να διαπιστώσουμε κατά πόσο η συγκεκριμένη επικοινωνία είναι ασφαλής. Με αυτό τον τρόπο λοιπόν διασφαλίζεται ότι δεν υπάρχει κίνδυνος υποκλοπής των προσωπικών μας δεδομένων. Το περιβάλλον λειτουργίας της εφαρμογής αυτής φαίνεται στο σχήμα C.1.



Σχήμα C.1: Περιβάλλον λειτουργία του Ethereal.

Βιβλιογραφία

- [1] R. Elmasri – S.B. Navathe, Θεμελιώδεις Αρχές Συστημάτων Βάσεων Δεδομένων, Τόμος Α', 3^η Αναθεωρημένη Έκδοση, Εκδόσεις ΔΙΑΥΛΟΣ [2001].
- [2] Abraham Silberschatz – Henry F. Korth – S. Sudarshan, Συστήματα Βάσεων Δεδομένων, 4^η Έκδοση, Εκδόσεις Μ. Γκιούρδας [2004].
- [3] Thomas Connolly – Carolyn Begg, Database Systems, Fourth Edition, Addison Wesley [2005].
- [4] Abraham Silberschatz – Henry F. Korth – S. Sudarshan, Database System Concepts, Fifth Edition, Mc Graw Hill [2006].
- [5] R. Elmasri – S.B. Navathe, Fundamentals Of Database Systems, Fourth Edition, Addison Wesley [2004].
- [6] Welcome to Windows CE 5.0, <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wceintro5/html/wce50oriWelcomeToWindowsCE.asp>, Σεπτέμβριος [2006].
- [7] Pocket PC, http://en.wikipedia.org/wiki/Pocket_PC, Σεπτέμβριος [2006].
- [8] What Is SQL Server 2005, <http://www.microsoft.com/sql/prodinfo/overview/what-is-sql-server.msp>, Σεπτέμβριος [2006].
- [9] SQL Server Database Engine, <http://msdn2.microsoft.com/en-us/library/ms187875.aspx>, Σεπτέμβριος [2006].
- [10] SQL Server Replication, <http://msdn2.microsoft.com/en-us/library/ms151198.aspx>, Σεπτέμβριος [2006].
- [11] SQL Server Mobile Product Overview, <http://www.microsoft.com/sql/editions/sqlmobile/overview.msp>, Ιούλιος [2006].
- [12] Internet Information Services, <http://en.wikipedia.org/wiki/IIS>, Σεπτέμβριος [2006].
- [13] Types of Replication Overview, <http://msdn2.microsoft.com/en-us/library/ms152531.aspx>, Ιούλιος [2006].
- [14] Replicating Data to SQL Server Mobile, <http://msdn2.microsoft.com/en-us/library/ms152568.aspx>, Ιούλιος [2006].
- [15] Merge Replication Overview, <http://msdn2.microsoft.com/en-us/library/ms152746.aspx>, Ιούλιος [2006].

-
- [16] Replication Architecture, <http://msdn2.microsoft.com/en-us/library/ms171927.aspx>, Ιούλιος [2006].
 - [17] How Replication Works, <http://msdn2.microsoft.com/en-us/library/ms172385.aspx>, Ιούλιος [2006].
 - [18] Microsoft Visual Studio, http://en.wikipedia.org/wiki/Visual_Studio_2005, Σεπτέμβριος [2006].
 - [19] .NET Compact Framework, http://en.wikipedia.org/wiki/.NET_Compact_Framework, Ιούλιος [2006].
 - [20] ActiveSync, <http://en.wikipedia.org/wiki/ActiveSync>, Ιούλιος [2006].
 - [21] Bruce Schneier, Applied Cryptography, Second Edition, Wiley [1996].
 - [22] Advanced Encryption Standard, http://en.wikipedia.org/wiki/Advanced_Encryption_Standard, Αύγουστος [2006].
 - [23] Rolf Oppliger, Contemporary Cryptography, Artech House [2005].
 - [24] SHA hash functions, http://en.wikipedia.org/wiki/SHA_hash_functions, Αύγουστος [2006].
 - [25] RSA, <http://en.wikipedia.org/wiki/RSA>, Σεπτέμβριος [2006].
 - [26] Public key certificate, http://en.wikipedia.org/wiki/Public_key_certificate, Αύγουστος [2006].
 - [27] Transport Layer Security, http://en.wikipedia.org/wiki/Transport_Layer_Security, Αύγουστος [2006].
 - [28] Htpps, <http://en.wikipedia.org/wiki/Htpps>, Αύγουστος [2006].
 - [29] Step by Step: Developing a SQL Mobile Application with Visual Studio 2005 and SQL Server 2005, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnppcgen/html/med302_msdn_sql_mobile.asp, Δεκέμβριος [2005].
 - [30] Modem & Applications: OMAP850, <http://focus.ti.com/general/docs/wtbu/wtbuproductcontent.tsp?templateId=6123&navigationId=12000&contentId=4679>, Σεπτέμβριος [2006].
 - [31] OpenSSL, <http://en.wikipedia.org/wiki/OpenSSL>, Σεπτέμβριος [2006].
 - [32] John Viega – Matt Messier – Pravir Chandra, Network Security with OpenSSL, O'Reilly [2002].

- [33] Dynamic DNS, http://en.wikipedia.org/wiki/Dynamic_DNS, Σεπτέμβριος [2006].
- [34] Ethereum (software), [http://en.wikipedia.org/wiki/Ethereum_\(software\)](http://en.wikipedia.org/wiki/Ethereum_(software)), Σεπτέμβριος [2006].