



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Ανάλυση και ένταξη του Λειτουργικού Συστήματος
“Android” σε περιβάλλον υπολογιστικού πλέγματος**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΧΟΝΔΡΟΓΙΑΝΝΗ ΕΥΘΥΜΙΟΥ

Επιβλέπων : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2008



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάλυση και ένταξη του Λειτουργικού Συστήματος “Android” σε περιβάλλον υπολογιστικού πλέγματος

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΧΟΝΔΡΟΓΙΑΝΝΗ ΕΥΘΥΜΙΟΥ

Επιβλέπων : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18^η Φεβρουαρίου 1999.

(Υπογραφή)

.....
Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

(Υπογραφή)

.....
Ελευθέριος Καγιάφας
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Βασίλειος Λούμος
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2008

Περίληψη

Ο σκοπός της διπλωματικής εργασίας ήταν η δημιουργία ενός συστήματος που θα μας επιτρέπει την ανακάλυψη και εκτέλεση υπηρεσιών από το κινητό μας τηλέφωνο. Για την υλοποίηση του συστήματός μας χρειάστηκε η δημιουργία μιας εφαρμογής που θα τρέχει στο κινητό τηλέφωνο πάνω από το “Android” καθώς και οι δημιουργία ορισμένων απαραίτητων services για την άσκησή μας. Το “Android” είναι ένα Λειτουργικό Σύστημα για κινητά τηλέφωνα το οποίο μας επιτρέπει να δημιουργούμε εύκολα εφαρμογές γι’ αυτά.

Σκόπός της υλοποίησης ήταν να στέλνουμε όλα τα ερωτήματα από το κινητό μας τηλέφωνο σε έναν server, καλώντας τα κατάλληλα services που δημιουργήσαμε, και αυτός με την σειρά του να αναλαμβάνει την ολοκλήρωση των ερωτημάτων μας.

Για να έχουμε πρόσβαση στις υπηρεσίες που υπάρχουν, χρησιμοποιήσαμε το Gria το οποίο μας δίνει την δυνατότητα να ανακαλύπτουμε με εύκολο τρόπο τις υπάρχουσες υπηρεσίες καθώς και να ανεβάζουμε ή κατεβάζουμε δεδομένα από τα gria resources και να εκτλούμε τις υπηρεσίες σε αυτά. Έτσι με την βοήθεια του Gria έχουμε την δυνατότητα να εντοπίσουμε τον server που μας παρέχει μια υπηρεσία και να ζητήσουμε κατόπιν την εκτέλεσή της. Για την δημιουργία πολύπλοκων υπηρεσιών που απαρτίζονται από τον συνδιασμό δύο ή περισσότερων υπηρεσιών, χρησιμοποιήσαμε το πρόγραμμα Taverna το οποίο μας επιτρέπει να δημιουργούμε εύκολα ροές εργασιών ή όπως αλλιώς λέγεται workflows.

Εκτός όμως από την δυνατότητα εκτέλεσης πολύπλοκων εργασιών από το κινητό μας τηλέφωνο, με το σύστημά μας έχουμε την δυνατότητα να αναζητήσουμε πληροφορίες στον σκληρό δίσκο όλων των κινητών τηλεφώνων, είτε από το κινητό μας τηλέφωνο είτε από κάποιον υπολογιστή. Για να το πετύχουμε αυτό δημιουργήσαμε στο κινητό τηλέφωνο έναν server ο οποίος μπορεί να απαντά στα ερωτήματά μας και στην συνέχεια, ανάλογα τι χρειαζόμαστε στέλνουμε το κατάλληλο ερώτημα σε ένα ή περισσότερα κινητά τηλέφωνα.

Λέξεις Κλειδιά: λογισμικό, κινητό τηλέφωνο, εξυπηρετής, υπηρεσία, ροή εργασιών

Abstract

The purpose of my diplomatic work was to create a system that will give us the possibility to execute services from mobile phone. In order to implement our system, it was needed to create an application which will run in mobile phone over Android stack. Also, it was needed to create some services which were important for our system. Android is a software stack and contains an operation system. It allows us to create easily applications for mobile phones.

The purpose of our implementation was to send our data from mobile phone in a server, calling the suitable services that we created. The server, after undertake to complete what we asked. In order to have access in existing services, we used Gria , which gives us the possibility to find easily the existing services and to upload or download data from gria resources or to execute services there. So, using Gria we have the possibility to find the server which offers a service and then execute the service there. In order to implement more complex services that they constituted from two or more services, we used the program Taverna, which allow us to create easily workflows.

Except from executing complex services from mobile phone, with our system we can search for a specific information in hard disk of all visible mobile phones. Also we can ask from mobile phones to return us some specify data that we want; such us were mobile phone is located. We can ask this from our mobile phone or from a computer. In order to do this we create a server in mobile phone, so mobile phone can answer in our questions.

Keywords: software, android, google, mobile phone, server, services, middleware, gria, axis, taverna, workflow, xml, java, apache tomcat

Πίνακας περιεχομένων

1	Εισαγωγή.....	3
1.1	Πρόλογος.....	3
1.2	Σύντομη ιστορική αναδρομή	4
2	Σκοπός της εργασίας - Απαραίτητα προγράμματα	6
2.1	Σκοπός της εργασίας.....	6
2.2	Απαραίτητα προγράμματα.....	8
3	Περιγραφή του συστήματος.....	9
3.1	Γενικά χαρακτηριστικά του συστήματος.....	9
3.2	Αναλυτική περιγραφή του συστήματος.....	12
4	Mobile Grid.....	28
4.1	Mobile Server	28
4.2	Mobile Grid	29
5	Τα workflows που δημιουργήσαμε.....	31
5.1	Υπάρχουσες υπηρεσίες.....	31
5.2	Workflows	32
6	Παραδείγματα εκτέλεσης - Αξιολόγηση του συστήματος.....	33
6.1	Παραδείγματα εκτέλεσης	33
6.2	Αξιολόγηση του συστήματός μας.....	37
7	Επίλογος.....	39
7.1	Σύνοψη και συμπεράσματα	39
8	Σχετικά Links.....	40
9	Παράρτημα Α.....	41
9.1	Workflow - Taverna	41
9.2	Gria.....	42
10	Παράρτημα Α.....	43
10.1	Κώδικας εφαρμογής κινητού τηλεφώνου	44
10.2	Κώδικας των web services.....	137
10.3	Ask Mobile Phone	180

1

Εισαγωγή

Στην ενότητα αυτή θα κάνουμε μια εισαγωγή στα κινητά τηλέφωνα και θα δούμε πως φτάσαμε στα κινητά τηλέφωνα που χρησιμοποιούμε σήμερα κάνοντας μια σύντομη ιστορική αναδρομή.

1.1 Πρόλογος

Κινητό τηλέφωνο, ένας όρος άγνωστος πριν τρεις δεκαετίες, πλέον ακούγεται τόσο κοινός όσο η λέξη άνθρωπος. Το κινητό τηλέφωνο έχει μπει για τα καλά στην ζωή μας και έχει αλλάξει τον τρόπο που επικοινωνούμε. Οπουδήποτε πηγαίνουμε το έχουμε μαζί μας για να μπορούμε να επικοινωνούμε με τους άλλους ανθρώπους. Όμως, παρόλο που δημιουργήθηκε – όπως λέει και το όνομά του – για να μας δίνει την δυνατότητα να καλούμε άλλους ανθρώπους, στις μέρες μας, μας παρέχει πολύ περισσότερες δυνατότητες. Μερικές από αυτές είναι:

- ✚ Η δυνατότητα εγγραφής και αναπαραγωγής ήχου.
- ✚ Η δυνατότητα λήψης φωτογραφιών και βίντεο.
- ✚ Η δυνατότητα επικοινωνίας μεταξύ κινητών τηλεφώνων αλλά και με τον υπολογιστή με υπέρυθρες και Bluetooth.
- ✚ Η δυνατότητα πρόσβασης στο ιντερνέτ.

Το κινητό τηλέφωνο έχει αλλάξει αρκετά στο πέρασμα των χρόνων. Για να κατανοήσουμε καλύτερα πως φτάσαμε στα κινητά τηλέφωνα που χρησιμοποιούμε σήμερα, θα δούμε στην επόμενη υποενότητα μια σύντομη ιστορική αναδρομή. Αυτό θα μας βοηθήσει να κατανοήσουμε όχι μόνο πως φτάσαμε εδώ , αλλά και να αντιληφθούμε τι αλλαγές πρόκειται να γίνουν .

1.2 Σύντομη ιστορική αναδρομή

Τα κινητά τηλέφωνα έκαναν την εμφάνισή τους στα μέσα της δεκαετίας του '80 (η ημερομηνία γέννησής τους θεωρείται η 3η Απριλίου 1973). Είχαν μεγάλο μέγεθος, μεταφέρονταν με δυσκολία και παρείχαν την δυνατότητα πραγματοποίησης τηλεφωνικών κλήσεων. Αυτά ήταν τα *κινητά τηλέφωνα πρώτης γενιάς 1G*.

Στις αρχές της δεκαετίας του '90, με την ψηφιοποίηση δικτύων (GSM) και συσκευών, άρχισε η απογείωση των κινητών τηλεφώνων. Τα κινητά τηλέφωνα έγιναν μικρότερα και έδιναν την δυνατότητα για την πραγματοποίηση τηλεφωνικών κλήσεων και αποστολή μηνυμάτων καθώς και πολύ περιορισμένα παιχνίδια, για να τραβήξουν την προσοχή των μικρότερων σε ηλικία. Αυτά ήταν τα *κινητά τηλέφωνα δεύτερης γενιάς 2G*.

Με το πέρασμα των χρόνων οι εταιρίες παραγωγής τους έριξαν βάρος στην μείωση του μεγέθους του κινητού τηλεφώνου και μέχρι το τέλος της προηγούμενης δεκαετίας το κινητό είχε γίνει τόσο μικρό και λεπτό όσο ένα κομπιουτεράκι τσέπης. Το κινητό δεν μπορούσε να μικρύνει άλλο.



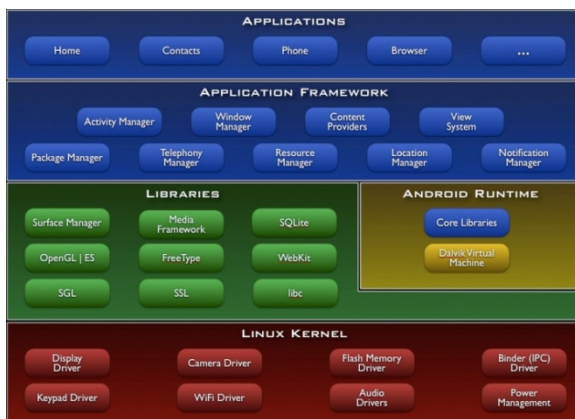
Στα χρόνια που ακολούθησαν οι εταιρίες παραγωγής τους ακολούθησαν άλλη πορεία. Μην μπορώντας να μικρύνουν άλλο το κινητό τηλέφωνο, άρχισαν να προσθέτουν σε αυτό επιπλέον δυνατότητες όπως δυνατότητα λήψης φωτογραφιών και βίντεο, περισσότερα παιχνίδια και πολλά άλλα. Αυτά ήταν τα *κινητά τηλέφωνα τρίτης γενιάς 3G*.

Μέσα σε λίγα χρόνια τα κινητά τηλέφωνα είχαν σχετικά μεγάλες οθόνες με πολύ καλή ανάλυση, είχαν την δυνατότητα λήψης φωτογραφιών και βίντεο καθώς και την δυνατότητα να επικοινωνούν μεταξύ τους και να μπαίνουν στο ιντερνέτ.



Στις μέρες μας, τα κινητά τηλέφωνα αλλάζουν και πάλι πορεία. Αυτό οφείλεται στην προσπάθεια δημιουργίας ενός λειτουργικού συστήματος που θα δίνει την δυνατότητα στους προγραμματιστές να δημιουργούν εφαρμογές για κινητά. Ένα τέτοιο λειτουργικό σύστημα είναι το android της Google.

(Τα κινητά που χρησιμοποιούσαμε μέχρι σήμερα είχαν λειτουργικό σύστημα αλλά η δομή του δεν επέτρεπε σε προγραμματιστές εκτός της εταιρία να δημιουργήσουν εφαρμογές για αυτά.)



Το android όπως φαίνεται και στην διπλανή εικόνα είναι μια στοίβα πακέτων λογισμικού.

Περιλαμβάνει ένα λειτουργικό σύστημα καθώς και όλες τις απαραίτητες κλάσεις για την δημιουργία εφαρμογών για κινητά.

Εμείς σε αυτήν την εργασία θα ασχοληθούμε με την δημιουργία μιας εφαρμογής για κινητά τηλέφωνα που θα τρέχει πάνω από το android.

2

Σκοπός της εργασίας μας – Απαραίτητα προγράμματα

2.1 Σκοπός της εργασίας μας

Σκοπός της εργασίας μας δεν είναι η δημιουργία μιας απλής εφαρμογής που θα τρέχει στο κινητό μας τηλέφωνο, αλλά η δημιουργία ενός συστήματος που θα μας δίνει την δυνατότητα αφενός μεν να εκτελούμε πολύπλοκες εργασίες μακριά από το κινητό τηλέφωνο αλλά και να μπορούμε να αναζητούμε τις πληροφορίες που θέλουμε στα άλλα κινητά τηλέφωνα.

Ο πρώτος λοιπόν στόχος της εργασίας μας είναι η δημιουργία μιας εφαρμογής που θα μας δίνει την δυνατότητα να εντοπίζουμε τις διαθέσιμες υπηρεσίες και κατόπιν να τις εκτελούμε. Οι υπηρεσίες που θέλουμε να είναι ορατές από το κινητό μας τηλέφωνο δεν παρέχονται από έναν μόνο server, αλλά από ένα σύνολο από servers γεγονός που κάνει δύσκολο τον εντοπισμό και την εκτέλεσή τους. Για τον σκοπό αυτό, προκειμένου να μπορούμε εύκολα να ανακαλύπτουμε τις διαθέσιμες υπηρεσίες και κατόπιν να τις εκτελούμε, χρησιμοποιήσαμε το Gria. Πρόκειται για ένα middleware που μας δίνει την δυνατότητα να διαχειριζόμαστε εύκολα τους διαθέσιμους πόρους και υπηρεσίες, χωρίς να χρειάζεται να γνωρίζουμε πολλές πληροφορίες γι' αυτά. Για να χρησιμοποιήσουμε όμως το Gria πρέπει να κάνουμε import στο project μας ξαποια jar αρχεία προκειμένου να χρησιμοποιήσουμε τα κατάλληλα object που θα χρειαστούμε. Το μέγεθος των jar αρχείων και η πολυπλοκότητα των κλάσεων που αυτά περιέχουν καθυστερούν απαγορευτικό να τα εισάγουμε στην εφαρμογή που θα τρέχει στο κινητό μας τηλέφωνο. Για τον σκοπό αυτό τις εργασίες που απαιτούνται για τον χειρισμό

των πόρων και τον υπηρεσιών θα τις εκτελούμε σε έναν server. Για να έχουμε όμως εμείς πρόσβαση από το κινητό μας τηλέφωνο στις υπηρεσίες που υπάρχουν, θα επικοινωνούμε με τον παραπάνω server ζητώντας του να πραγματοποιήσει τις ενέργειες που του ζητήσαμε από το κινητό μας τηλέφωνο. Για να υλοποιήσουμε την επικοινωνία αυτή, στον παραπάνω server, φτιάξαμε ένα σύνολο από services τα οποία χρησιμοποιούν τον Gria ανάλογα με τι θα θέλαμε να κάνουμε. Έπειτα από το κινητό μας τηλέφωνο, ανάλογα τι θα θέλαμε, καλούμε το κατάλληλο service από αυτά που δημιουργήσαμε και το οποίο αναλαμβάνει την περάτωση της εργασίας που του ζητήσαμε.

Ο δεύτερος στόχος της εργασίας μας είναι η δυνατότητα επικοινωνία με το κινητό τηλέφωνο. Για να το πούμε λίγο διαφορετικά θα θέλαμε να μπορεί κάποιος να ξεκινήσει αυτός την συνομιλία με το κινητό τηλέφωνο. Για να το πετύχουμε αυτό, δημιουργήσαμε στο κινητό τηλέφωνο έναν server ο οποίος μπορεί να απαντά σε διάφορα ερωτήματα. Με την βοήθεια του server που φτιάξαμε έχουμε την δυνατότητα να ρωτάμε αν ένα κινητό τηλέφωνο έχει κάποιο αρχείο, να του ζητάμε να μας στείλει ένα αρχείο που έχει καθώς και να μπορούμε να του ζητάμε να εκτελέσει μερικές απλές υπηρεσίες, όπως να μας πεί σε ποια περιοχή βρίσκεται. Για να το πετύχουμε όμως αυτό, πρέπει να γνωρίζουμε το ip του κινητού τηλεφώνου για να μπορούμε να απευθυνθούμε σε αυτό και να μπορέσουμε να το ρωτήσουμε αυτό που θέλουμε. Για τον σκοπό αυτό, στον server που λέγαμε λίγο ποιο πριν κρατάμε τα ip από όλα τα κινητά τηλέφωνα. Εάν το ip κάποιου κινητού τηλεφώνου βρίσκεται στον server τότε αυτό είναι ορατό από τα υπόλοιπα κινητά τηλέφωνα. Ο κάθε χρήστης του κινητού τηλεφώνου έχει την δυνατότητα να επιλέξει αν θέλει να είναι ορατός ή όχι, ζητώντας από τον server να προσθέσει ή να αφαιρέσει την ip του από αυτόν. Χρησιμοποιώντας τις ip που υπάρχουν στον server μπορούμε να ψάχνουμε σε όλα τα κινητά τηλέφωνα αυτό που ψάχνουμε είτε πρόκειται για κάποιο αρχείο είτε για μια συγκεκριμένη πληροφορία.

Συνοψίζοντας τα παραπάνω, σκοπός της εργασίας αυτής είναι η ανάπτυξη ενός συστήματος που θα μας προσφέρει τις παρακάτω δυνατότητες:

- ❖ Να ανεβάζουμε δεδομένα στα gria resources.
- ❖ Να κατεβάζουμε τα δεδομένα από τα gria resources.
- ❖ Να αναζητάμε αρχεία ή πληροφορίες στα υπόλοιπα (ορατά) κινητά τηλέφωνα.
- ❖ Να εκτελούμε εργασίες/υπηρεσίες, μακριά από το κινητό μας, σε δεδομένα μας που υπάρχουν είτε στο κινητό είτε στα gria resources.

Τα δεδομένα που ανεβάζουμε από ένα κινητό τηλέφωνο στα gria resources, είναι ορατά μόνο από το κινητό τηλέφωνο που τα ανέβασε και μόνο αυτό έχει πρόσβαση σε αυτά. Η δυνατότητα αυτή, να μπορούμε να ανεβάζουμε δεδομένα από το κινητό μας τηλέφωνο στα gria resources, μας δίνει επιπλέον αποθηκευτικό χώρο. Τον αποθηκευτικό αυτόν χώρο, από εδώ και στο εξή θα τον λέμε Remote Hard Disk. Ο αποθηκευτικός χώρος στην εργασία αυτή, μας παρέχεται δωρεάν.

Οι υπηρεσίες που καλούμε από το κινητό μας τηλέφωνο είναι επίσης δωρεάν.

2.2 Απαραίτητα προγράμματα

Για την υλοποίηση του συστήματός μας χρησιμοποιήσαμε τα παρακάτω “προγράμματα” .

- **Android** :
Το «Android» είναι μια στοίβα λογισμικού που περιλαμβάνει ένα Λειτουργικό Σύστημα και όλες τις απαραίτητες κλάσεις για την δημιουργία εφαρμογών για κινητά τηλέφωνα.
- **Axis** :
Το Axis είναι μια μηχανή επεξεργασίας SOAP μηνυμάτων. Το χρησιμοποιήσαμε για την δημιουργία των service μας.
- **Gria** :
Το Gria είναι ένα πρόγραμμα που μας συνδέει υπηρεσίες και πόρους έτσι ώστε να μπορούμε εύκολα να τα εντοπίσουμε και να τα χρησιμοποιήσουμε. Περισσότερες πληροφορίες για το Gria υπάρχουν στο Παράρτημα Α.
- **Taverna** :
Είναι ένα πρόγραμμα το οποίο χρησιμοποιήσαμε για την δημιουργία ροών εργασιών – workflow. Περισσότερες πληροφορίες για τα workflows και το Taverna υπάρχουν στο παράρτημα Α.

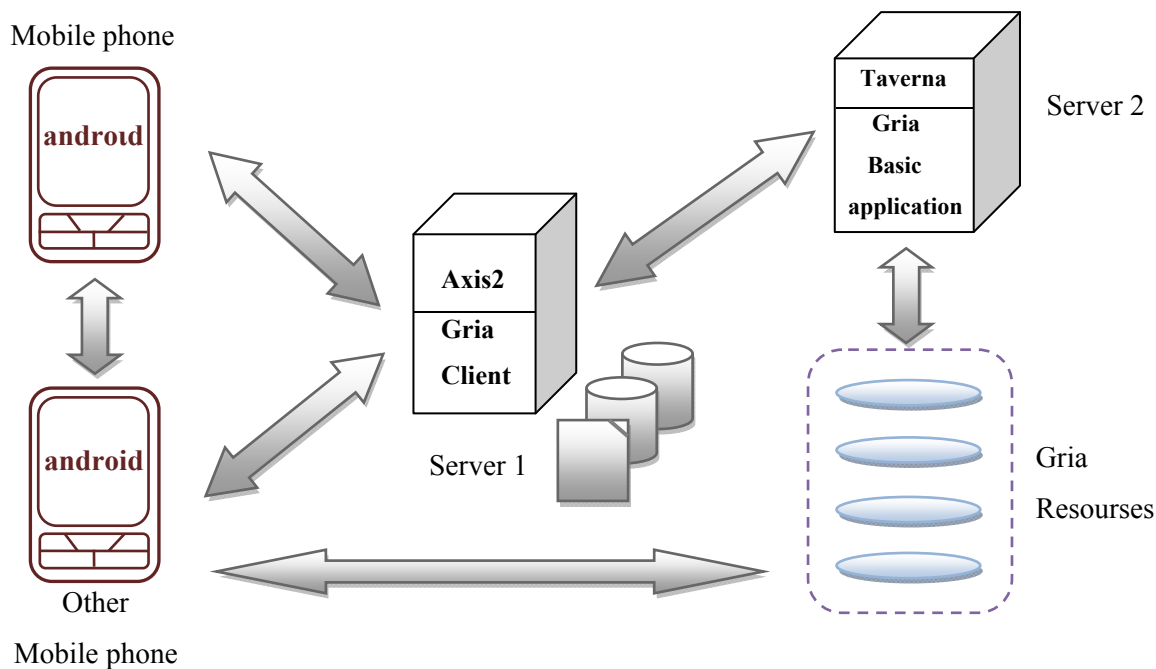
3

Περιγραφή του συστήματος

Στην ενότητα αυτή θα περιγράψουμε το σύστημά μας. Για να είναι πιο κατανοητή η περιγραφή αρχικά, στην υποενότητα 3.1 θα περιγράψουμε ποια είναι τα βασικά συστατικά μέρη του συστήματός και τι θέλουμε να κάνει καθένα από αυτά μας και στην συνέχεια στην υποενότητα 3.2 θα δώσουμε όλες τις απαραίτητες λεπτομέρειες σχετικά με το πώς έχει υλοποιηθεί καθένα από τα συστατικά μέρη του συστήματός μας.

3.1 Γενικά χαρακτηριστικά του συστήματος

Στην ενότητα αυτή, θα δούμε ποια είναι τα βασικά μέρη του συστήματός μας. Το σύστημά μας λοιπόν, θα αποτελείται από τέσσερα μέρη. Το κινητό τηλέφωνο από το οποίο θα καλούμε τις υπηρεσίες που θέλουμε, έναν server ο οποίος θα δέχεται τα δεδομένα που του στέλνουμε και τις ενέργειες που θέλουμε να κάνουμε σε αυτά, και έναν άλλον server ο οποίος θα μας παρέχει την πρόσβαση στα `gria resource` , προκειμένου να υλοποιηθούν οι υπηρεσίες/ενέργειες που ζητήσαμε.



Παρατήρηση:

Τα προγράμματα που έχουν υλοποιηθεί στους server1 , server2 και gria-resources δεν είναι απαραίτητο να βρίσκονται σε διαφορετικούς υπολογιστές.

Όπως βλέπουμε από το παραπάνω σχήμα, από το κινητό μας τηλέφωνο έχουμε την δυνατότητα να επικοινωνήσουμε με κάποιο άλλο κινητό τηλέφωνο με σκοπό να ζητήσουμε κάτι από αυτό όπως ένα αρχείο ή την εκτέλεση μια απλής εργασίας. Επίσης έχουμε την δυνατότητα να επικοινωνήσουμε με έναν server, τον server 1 προκειμένου να ζητήσουμε την υλοποίηση της υπηρεσίας που επιθυμούμε. Ο server αυτός, χρησιμοποιώντας το gria client api, με την βοήθεια του server2 στον οποίον έχουμε εγκαταστήσει το gria basic application, μας παρέχει πρόσβαση στα gria resources όπου υπάρχουν οι διαθέσιμες υπηρεσίες. Στον server 1, για τις ανάγκες της εργασίας, δημιουργήσαμε δύο βάσεις δεδομένων και ένα xml αρχείο τα οποία θα εξετάσουμε αναλυτικά στην επόμενη ενότητα. Τα services, σύμφωνα με το σχήμα, έχουν την δυνατότητα να έρθουν σε άμεση επικοινωνία με κάποιο από τα κινητά τηλέφωνα, αν βέβαια κριθεί ότι αυτό είναι απαραίτητο.

Στη συνέχεια θα δώσουμε ορισμένες πληροφορίες για καθένα από τα συστατικά μέρη του συστήματός μας.

Μέρος 1: Mobile application

Το πρώτο μέρος του συστήματός μας, θα αποτελείται από μία εφαρμογή η οποία θα είναι γραμμένη σε java και θα τρέχει στο κινητό τηλέφωνο πάνω από το android. Την εφαρμογή αυτή από εδώ και στο εξής θα την λέμε MobileApplication. Επειδή το κινητό έχει περιορισμένες δυνατότητες, θα προσπαθήσουμε η εφαρμογή αυτή να χρειάζεται όσο το δυνατόν λιγότερη υπολογιστική ισχύς.

Η εφαρμογή αυτή, θα μας δίνει τις παρακάτω δυνατότητες:

- ❖ Να ανεβάζουμε αρχεία στον RemoteHardDisk
- ❖ Να κατεβάζουμε αρχεία από τον RemoteHardDisk ή από κάποιο άλλο κινητό τηλέφωνο
- ❖ Να εκτελούμε διάφορες ενέργειες στα αρχεία που έχουμε είτε στο κινητό μας τηλέφωνο είτε στον RemoteHardDisk.

Μέρος 2 : My Services

Το δεύτερο μέρος του συστήματός μας , θα αποτελείται από ένα πρόγραμμα (για την ακρίβεια ένα σύνολο από services) το οποίο θα τρέχει σε έναν server και θα μας προσφέρει ορισμένες υπηρεσίες (τις οποίες θα δούμε παρακάτω) οι οποίες θα μας δίνουν την δυνατότητα να ανεβάζουμε δεδομένα από τον server 1 στα gria resources και αντίστροφα αλλά και να εκτελούμε υπηρεσίες στα gria resources. Το πρόγραμμα αυτό θα το λέμε myServices. Για την υλοποίησή του χρησιμοποιήσαμε την axis2 καθώς και το gria-client-api (version 5.2) προκειμένου να έχουμε πρόσβαση στις υπηρεσίες του πλέγματος μέσω του gria-basic-application.

Θα μπορούσε κανείς να σκεφτεί ότι το πρόγραμμα αυτό δεν είναι απαραίτητο και ότι θα μπορούσαμε να καλούμε το gria basic application κατευθείαν από το κινητό. Κάτι τέτοιο όμως δεν μπορεί να συμβεί (προς το παρών τουλάχιστον) καθώς οι συναρτήσεις του gria client library είναι πολύπλοκες και δεν θα μπορούσαν να εκτελεστούν στο κινητό.

Μέρος 3 : Gria

Το τρίτο μέρος του συστήματός μας, θα αποτελείται από μία εφαρμογή η οποία θα τρέχει σε έναν άλλον server (Server 2), και η οποία θα μας παρέχει πρόσβαση στις υπηρεσίες πλέγματος. Για τον σκοπό αυτό δεν θα δημιουργήσουμε μια νέα εφαρμογή αλλά θα χρησιμοποιήσουμε το gria-basic-application η οποία μας επιτρέπει με εύκολο τρόπο να ανεβάζουμε και να κατεβάζουμε δεδομένα από τα gria resources καθώς και να εκτελούμε υπάρχουσες υπηρεσίες σ' αυτά.

Επειδή όμως για την εφαρμογή μας θα χρειαστούμε να υλοποιήσουμε έναν συνδυασμό από τις υπάρχουσες υπηρεσίες, για τον λόγο αυτόν χρησιμοποιήσαμε το taverna που μας επιτρέπει να υλοποιήσουμε εύκολα workflow.

Μέρος 4 : Gria - resources

Το τέταρτο μέρος του συστήματός μας, θα αποτελούν τα gria resources, στα οποία θα ανεβάζουμε και θα κατεβάζουμε δεδομένα καθώς και να εκτελούμε διάφορες υπηρεσίες.

Ο τρόπος υλοποίησης των υπηρεσιών δεν θα μας απασχολήσει σε αυτήν την εργασία. Αυτό όμως στο οποίο θα δώσουμε ιδιαίτερη βαρύτητα είναι το γεγονός ότι οι υπηρεσίες αυτές θα έχουν την δυνατότητα κατά την διάρκεια της εκτέλεσής τους να έρθουν σε άμεση επικοινωνία με το κινητό και να ζητήσουν όποιες πληροφορίες χρειάζονται από αυτό κατά την διάρκεια εκτέλεσής τους (καθώς πολλές φορές δεν θα μπορούν να ξέρουν από πριν τι πληροφορίες θα χρειαστούν).

Το κόστος των υπηρεσιών, θεωρούμε ότι είναι μηδενικό, δηλαδή ότι οι υπηρεσίες προσφέρονται δωρεάν.

Όπως γίνεται αντιληπτό από τα παραπάνω, στόχος μας είναι να δημιουργήσουμε ένα σύστημα το οποίο δεν θα περιορίζεται απλώς στην εκτέλεση κάποιων ενεργειών στα δεδομένα μας, αλλά στην πραγματική επικοινωνία και συνεργασία με το κινητό μέσα από τις υπηρεσίες του πλέγματος καθ' όλη την διάρκεια εκτέλεσής τους.

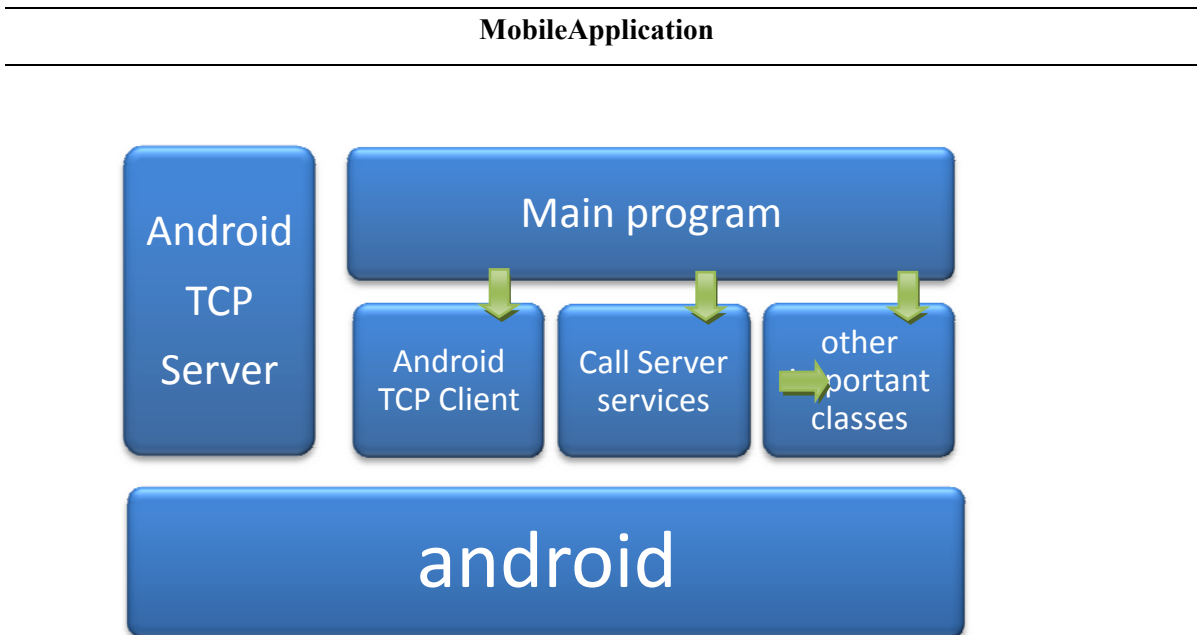
3.2 Αναλυτική περιγραφή του συστήματος

Στην ενότητα αυτή θα μπορούμε με περισσότερες λεπτομέρειες τον τρόπο υλοποίησης καθενός από τα παραπάνω μέρη.

Για την καλύτερη κατανόηση του συστήματός μας, την ανάλυσή μας την έχουμε χωρίσει σε δύο ενότητες.

3.2.1. Βασικά μέρη του *MobileApplication* και του *myServices*.

Στην ενότητα αυτή θα περιγράψουμε ποια είναι τα βασικά συστατικά μέρη του *MobileApplication* και του *myServices*, τι κάνει το καθένα από αυτά και πως συνεργάζονται μεταξύ τους.



Όπως φαίνεται από το παραπάνω σχήμα, η εφαρμογή μας στο κινητό τηλέφωνο αποτελείται από 5 μέρη.

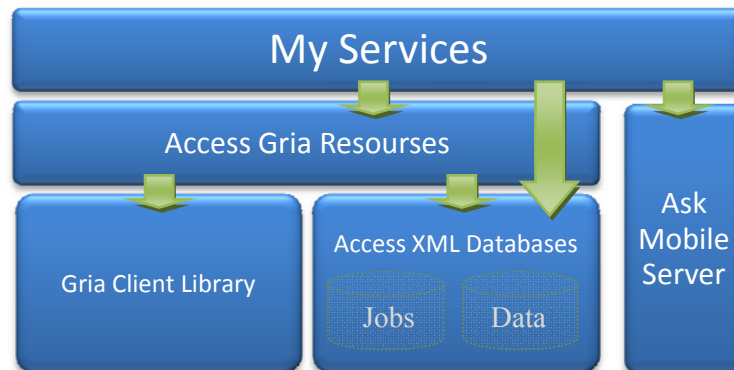
- Call Server services
Περιέχει όλες τις απαραίτητες κλάσεις - συναρτήσεις για να καλούμε τα services που είναι στον server1
- Android TCP Client
Περιέχει τις κλάσεις – συναρτήσεις για να μπορούμε να στέλνουμε ερωτήματα σε κάποιο άλλο κινητό τηλέφωνο.
- Other important classes
Περιέχει διάφορες κλάσεις που ήταν απαραίτητες για την εφαρμογή μας.
- Android TCP Server
Περιέχει τις απαραίτητες κλάσεις για την δημιουργία ενός απλού server. Ο server αυτός ξεκινά την πρώτη φορά που εκτελούμε την εφαρμογή μας και συνεχίζει να εκτελείται για πάντα. Δίνει την δυνατότητα στο κινητό μας τηλέφωνο να μπορεί να δέχεται ερωτήματα και να απαντάει σε αυτά.

- Main program

Περιέχει όλες τις απαραίτητες κλάσεις για την δημιουργία ενός γραφικού περιβάλλοντος που θα μας διευκολύνει να εκτελούμε τις εργασίες μας.

Οι κλάσεις αυτού του πακέτου, χρησιμοποιούν τις κλάσεις - συναρτήσεις των προηγούμενων πακέτων.

myServices



Σύμφωνα με το παραπάνω σχήμα, η εφαρμογή μας είναι χωρισμένη σε 5 τμήματα:

- Gria Client Library

Στο πακέτο αυτό βρίσκονται όλες οι απαραίτητες κλάσεις - συναρτήσεις για την επικοινωνία με το gria basic application.

- Access XML Databases

Το πακέτο αυτό περιέχει όλες τις απαραίτητες κλάσεις – συναρτήσεις για την δημιουργία και τον χειρισμό των βάσεων δεδομένων μας.

Οι βάσεις δεδομένων που χρειαστήκαμε είναι η Jobs Database που περιέχει πληροφορίες για τις συναρτήσεις που εκτελεί ο χρήστης του κινητού τηλεφώνου και η Data Database η οποία περιέχει πληροφορίες για τα αρχεία που ανεβάζουμε στον Remote Hard Disk. Οι δύο αυτές βάσεις δεδομένων στην άσκησή μας είναι δύο XML αρχεία.

- Access Gria resources

Μας προσφέρει όλες τις απαραίτητες συναρτήσεις για να ανεβάζουμε, κατεβάζουμε και να εκτελούμε υπηρεσίες στα gria-resources.

- Ask Mobile Server

Μας δίνει την δυνατότητα να στείλουμε ένα ερώτημα στον server που τρέχει στο κινητό τηλέφωνο.

- My Services

Εδώ έχουμε υλοποιήσει τις υπηρεσίες μας.

3.2.2. Αναλυτική περιγραφή MobileApplication του και του myServices .

Στην ενότητα αυτή, θα δώσουμε αρκετές πληροφορίες, σχετικά με το πως έχουν υλοποιηθεί καθένα από τα παραπάνω τμήματα.

Καταρχήν, στον server 1 κρατάμε ένα xml αρχείο που περιέχει πληροφορίες για τις συναρτήσεις που υλοποιούνται στο πλέγμα. Το αρχείο αυτό έχει την μορφή:

- supported_functions.xml

```
<supported_functions>
  <function>
    <name>      ...      </name>
    <url>       ...      </url>
    <description> ... </description>
    <inputs>    ...      </inputs>
    <outputs>   ...      </outputs>
    <type>      ...      </type>
  </function>
</supported_functions>
```

Στο αρχείο αυτό κρατάμε όλες τις απαραίτητες πληροφορίες για τις συναρτήσεις/υπηρεσίες που υπάρχουν στον Gria και θέλουμε να είναι ορατές από το κινητό μας τηλέφωνο.

Την συνέχεια της περιγραφή μας, για να είναι πιο κατανοητή, την έχουμε χωρίσει και εδώ σε δύο μέρη:

Μέρος Α

Το πρώτο μέρος περιλαμβάνει τον τρόπο υλοποίησης των τμημάτων Android TCP Client , Android TCP Server , Call Server Services, Other important classes στο κινητό τηλέφωνο, καθώς και τον τρόπο υλοποίησης των Access Gria resources, Access XML Database και Ask Mobile Server στο πρόγραμμα που τρέχει στον Server 1.

MobileApplication

- **Call Server Services**

Το πακέτο αυτό, μας προσφέρει όλες τις απαραίτητες συναρτήσεις για καλούμε τα services που βρίσκονται στον server 1.

Για να το πετύχουμε αυτό, έχουμε δημιουργήσει μία συνάρτηση η οποία στέλνει στον server 1 με http post request τα δεδομένα μας. Το url στο οποίο στέλνουμε τα δεδομένα μας έχει την μορφή:

http://server-ip:8080/axis2/services/my-class/function-name

Ακολουθεί ο κώδικας της συνάρτησης αυτής:

```

/** Call service: functionname with parameters: parameters ...
    And read results */
private static String executeServerFunction(String functionname,
String parameters ) throws IOException
{
    // URL Android-Server
    URL servlet = new URL(serviceUrl+functionname);

    // Send Data
    URLConnection conn = servlet.openConnection();
    conn.setDoOutput(true);
    conn.setDoInput(true);
    DataOutputStream dos = new DataOutputStream(conn.getOutputStream());
    dos.writeBytes(parameters);
    dos.close();

    // Receive Data
    DataInputStream dis = new DataInputStream(conn.getInputStream());
    String response="";
    String nextline;
    while ((nextline = dis.readLine()) != null)
    {
        response+=nextline;
    }
    dis.close();
}

```

Για κάθε service έχουμε δημιουργήσει και μια νέα συνάρτηση η οποία συγκεκριμενοποιώντας τα functionname και parameters καλεί το κατάλληλο service χρησιμοποιώντας την παραπάνω συνάρτηση.

Τα δεδομένα (parameters) που στέλνουμε με την http post request πρέπει να έχουν την μορφή:
parameter1=my-data-1¶meter2=my-data-2 ...

(Χρησιμοποιήσαμε Post Request για να μπορούμε να στέλνουμε όσα δεδομένα θέλουμε.)

Για την αποστολή του περιεχομένου των αρχείων μας από το κινητό τηλέφωνο στον server, χρησιμοποιήσαμε τον παραπάνω τρόπο στέλνοντας το ως μέρος της http post request αφού πρώτα τα μετατρέψαμε σε string. Για την μετατροπή των εικόνων σε string χρησιμοποιήσαμε το commons-codec-1.3.jar και πιο συγκεκριμένα την συνάρτηση encode (στον server αντίστοιχα χρησιμοποιήσαμε την decode).

Για την μεταφορά των αρχείων από τον server 1 στο κινητό μας τηλέφωνο δεν χρησιμοποιήσα τον παραπάνω τρόπο (για να είμαι ειλικρινής το δοκίμασα αλλά η μεταφορά των αρχείων με αυτόν τον τρόπο απαιτούσε αρκετό χρόνο). Έτσι για την μεταφορά των αρχείων δημιούργησα HTTP connection χρησιμοποιώντας ως URL το url του αρχείου.

- **Android TCP Server**

Το πακέτο αυτό, μας προσφέρει όλες τις απαραίτητες κλάσεις για την δημιουργία ενός server στο κινητό μας τηλέφωνο.

Για την δημιουργία του server χρησιμοποιήσαμε ένα νέο νήμα, έτσι ώστε να μην σταματάει η εκτέλεσή του προγράμματός μας. Μέσα στο νήμα αυτό περιμένουμε για tcp requests. Ο server μας ακούει στην θύρα 4444.

Για να συνεχίσει το νήμα να εκτελείται ακόμα και όταν δεν είμαστε μέσα στην εφαρμογή μας, το μετατρέπουμε σε service κάνοντας τις απαραίτητες αλλαγές στο αρχείο AndroidManifest.xml.

Μέσα λοιπόν στο νήμα που δημιουργήσαμε, περιμένουμε μέχρι να μας στείλει κάποιος κάποιο ερώτημα. Όταν αυτό συμβεί, απαντάμε στο ερώτημα (αν μπορούμε) επιστρέφοντάς του το αποτέλεσμα και συνεχίζουμε να περιμένουμε για το επόμενο ερώτημα. Αν δεχτούμε κάποιο ερώτημα, ενώ το προηγούμενο δεν έχει ακόμη ολοκληρωθεί, τότε θα περιμένουμε μέχρι να ολοκληρωθεί το προηγούμενο και ύστερα θα απαντήσουμε στο ερώτημα αυτό. Αυτό το κάναμε γιατί το κινητό έχει περιορισμένες δυνατότητες και αν απαντούσαμε παράλληλα σε περισσότερα από ένα ερωτήματα, αυτό θα μείωνε σημαντικά τις επιδόσεις του κινητού.

Τα ερωτήματα στα οποία απαντά ο server μας είναι της μορφής: question : parameters

Η μορφή των ερωτημάτων είναι σαν την μορφή με την οποία καλούμε μια συνάρτηση σε μία γλώσσα προγραμματισμού, οπότε θα μπορούσαμε να πούμε ότι ο server εκτελεί κάποιες συναρτήσεις οι παράμετροι των οποίων είναι string.

Οι συναρτήσεις που έχουμε υλοποιήσει μας δίνουν την δυνατότητα να ρωτάμε αν το κινητό υποστηρίζει μια συνάρτηση, να ζητάμε περισσότερες πληροφορίες για την συνάρτηση αυτή καθώς και να ζητάμε την εκτέλεση της συνάρτησης. Αν ζητήσουμε από τον server αυτόν κάτι που δεν γνωρίζει, επιστρέφει error.

Μία βασική συνάρτηση που υποστηρίζει ο server είναι αυτή με την οποία ζητάμε από το κινητό να μας στείλει ένα αρχείο κειμένου ή εικόνας. Την συνάρτηση αυτή την χρησιμοποιούμε όταν θέλουμε να πάρουμε ένα αρχείο κατευθείαν από κάποιο άλλο κινητό τηλέφωνο.

- **Android TCP Client**

Το πακέτο αυτό, μας προσφέρει μία κλάση – συνάρτηση με την οποία μπορούμε να στέλνουμε κάποιο ερώτημα σε κάποιο άλλο κινητό τηλέφωνο (στο οποίο βέβαια τρέχει ο server που περιγράψαμε παραπάνω). Ανάλογα με το ερώτημα που στέλναμε διαβάσαμε και τα κατάλληλα δεδομένα που μας έστειλε το άλλο κινητό τηλέφωνο, σαν απάντηση στο ερώτημα που στείλαμε.

- **Other important classes**

Το πακέτο αυτό, μας προσφέρει κάποιες απαραίτητες κλάσεις τις οποίες δεν μπορούσαμε να εντάξουμε σε κάποιο από τα παραπάνω τμήματα.

Μια απαραίτητη για την άσκηση μας κλάση είναι η `AndroidFile` η οποία μας παρέχει όλες τις συναρτήσεις που χρειαστήκαμε για τον χειρισμό των αρχείων που βρίσκονται στο κινητό τηλέφωνο. Πιο συγκεκριμένα μας προσφέρει τις συναρτήσεις για την δημιουργία και την λήψη του περιεχομένου ενός αρχείου κειμένου. Επίσης μας προσφέρει τις συναρτήσεις για την εμφάνιση μιας εικόνας. Τέλος μας παρέχει μια συνάρτηση για την λήψη των bytes ενός αρχείου.

Μια ακόμη απαραίτητη κλάση για την άσκησή μας ήταν η `Function` η οποία μας παρέχει διάφορες συναρτήσεις όπως η δημιουργία ενός πίνακα με strings από ένα μόνο string στο οποίο τα strings χωρίζονται με κενά. Επίσης μια συνάρτηση που επιστρέφει όλα τα ονόματα από τα αρχεία που βρίσκονται στο κινητό μας τηλέφωνο.

Μια ακόμα πολύ χρήσιμη συνάρτηση που υπάρχει είναι η `getText`. Η συνάρτηση αυτή μας επιτρέπει να απομονώσουμε το κείμενο που βρίσκεται σε ένα xml-string μεταξύ των `<tag>` και `</tag>`. Προϋποθέτει ότι υπάρχει ένα μόνο `<tag>` και ένα μόνο `</tag>` στο xml-string για να λειτουργεί σωστά.

```
public static String getText(String xmlstring, String childname)
{
    int indexstart = xmlstring.indexOf("<"+childname+">");
    int indexend = xmlstring.indexOf("</"+childname+">");
    if((indexstart>=0) && (indexend>=0))
    {
        int offset = childname.length()+2;
        String response =
            xmlstring.substring(indexstart+offset, indexend);
        return response;
    }
    else{ return null; }
}
```

Την συνάρτηση αυτήν θα την χρησιμοποιήσουμε όταν θα χρειαστούμε να χειριστούμε ένα XML αρχείο.

Μια ακόμη απαραίτητη κλάση είναι η `MyStringTokenizer`. Την κλάση αυτή την χρησιμοποιούμε για να χωρίσουμε ένα string σε τμήματα με βάση κάποιο άλλο string. Για να το πετύχουμε αυτό, κατά την δημιουργία ενός αντικειμένου της κλάσης αυτής με παραμέτρους τα string `s1` και `s2`, διαβάζουμε όλο το string `s1` και εντοπίζουμε τα σημεία στα οποία βρήκαμε το string `s2`. Έπειτα, χρησιμοποιώντας τις συναρτήσεις `hasMoreTokens()` και `nextToken()` που υλοποιήσαμε στην κλάση αυτή, μπορούμε να πάρουμε τα τμήματα του `s1` που χωρίζονται μεταξύ τους με το `s2`.

Χρησιμοποιώντας την παραπάνω κλάση και την συνάρτηση `getText` που περιγράψαμε νωρίτερα, έχουμε την δυνατότητα να χειριστούμε με αποδοτικό τρόπο xml αρχεία επιπέδου δύο.

Έστω για παράδειγμα το παρακάτω xml αρχείο:

```
<function>
    <name> ... </name>
    <url>    ... </url>
</function>
```

Για να βρούμε το `name` και το `url` καθεμιάς από τις `function` που υπάρχουν στο αρχείο χρησιμοποιούμε τον παρακάτω κώδικα:

```
MyStringTokenizer tokens =
new MyStringTokenizer(filecontent, "<function>");

while(tokens.hasMoreTokens())
{
    String tim = tokens.nextToken();
    String name = Function.getText(tim, "name");
    String url = Function.getText(tim, "url");
}
```

Τον παραπάνω τρόπο θα τον χρησιμοποιήσουμε για να χειριστούμε το xml αρχείο που περιέχει πληροφορίες για τις υπάρχουσες – ορατές συναρτήσεις.

Στο πακέτο αυτό βρίσκεται επίσης η κλάση `Thread_FunctionsInformations` την οποία χρησιμοποιούμε για να κατεβάσουμε πληροφορίες για τις υπηρεσίες που μας παρέχονται και η κλάση `Thread_PersonalSettings` την οποία χρησιμοποιούμε για να στείλουμε ή να μην στείλουμε στον server 1 την ip μας , ανάλογα με τις πληροφορίες που υπάρχουν στο τοπικό αρχείο `settings.txt`.

myServices

- **Gria client Library**

Μας προσφέρει low-level java objects για να μπορούμε να έχουμε πρόσβαση σε απομακρυσμένες υπηρεσίες.

- **Access XML Databases**

Μας προσφέρει τις συναρτήσεις για την δημιουργία των βάσεων δεδομένων (δηλαδή των xml αρχείων) καθώς και τις συναρτήσεις για την εισαγωγή, διαγραφή, τροποποίηση και εύρεση των στοιχείων στις βάσεις δεδομένων.

Οι Βάσεις δεδομένων που χρησιμοποιήσαμε έχουν την μορφή:

dataDatabase.xml

Εδώ κρατάμε τις πληροφορίες για κάθε αρχείο που ανεβάζουμε στον RemoteHardDisk.

```
<data>
  <file>
    <phoneid>      ... </phoneid>
    <name>         ... </name>
    <conversationid> ... </conversationid>
  </file>
</data>
```

jobDatabase.xml

Εδώ κρατάμε τις πληροφορίες για τις συναρτήσεις/υπηρεσίες που καλεί ο χρήστης από το κινητό του τηλέφωνο.

```
<function>
  <work>
    <phoneid>      ... </phoneid>
    <functionname> ... </functionname>
    <inputs>
      <input>      ... </input>
    </inputs>
    <outputs>
      <output>     ... </output>
    </outputs>
    <status>       ... </status>
    <timebegin>    ... </ timebegin >
    <timeend>     ... </ timeend >
    <conversationid> ... </conversationid>
  </work>
</function>
```

Και τα δύο αρχεία είναι well-formed, για να μπορούμε να τα επεξεργαστούμε με την βοήθεια του XML parser.

- **Access Gria resource**

Μας παρέχει τις συναρτήσεις για να κάνουμε upload, download και delete αρχεία καθώς επίσης και τις συναρτήσεις για να ξεκινάμε και να σταματάμε μια εργασία.

Οι συναρτήσεις αυτές έχουν υλοποιηθεί ως εξής:

Upload(mobileIp, filename)

Αρχικά ελέγχουμε αν για το συγκεκριμένο mobileIp, το αρχείο υπάρχει στην βάση δεδομένων dataDatabase.

Αν υπάρχει τότε επιστρέφουμε ότι το αρχείο υπάρχει, διαφορετικά ανεβάζουμε το αρχείο στα gria resources, χρησιμοποιώντας το gria client api και στην συνέχεια εισάγουμε στην dataDatabase το **ip** του κινητού στο οποίο ανήκει το αρχείο, το **όνομα** του αρχείου και το **conversation-id** (από το upload στα gria resources) και επιστρέφουμε ok αν όλα πάνε καλά ή error αν προκύψει κάποιο σφάλμα.

Download(mobileIp, filename)

Αρχικά εξετάζουμε αν για το συγκεκριμένο mobileIp, το αρχείο υπάρχει στη βάση δεδομένων dataDatabase.

Αν βρούμε ότι υπάρχει, χρησιμοποιώντας το **conversation-id** και το gria client api κάνουμε download το αρχείο και αν όλα πάνε καλά επιστρέφουμε ok διαφορετικά error.

Αν βρούμε όμως ότι το αρχείο δεν υπάρχει στη dataDatabase, τότε απλά επιστρέφουμε ότι το αρχείο που ζητάμε δεν υπάρχει.

Delete(mobileIp, filename)

Αρχικά εξετάζουμε αν το αρχείο μας που θέλουμε να διαγράψουμε υπάρχει στην dataDatabase και αν όντως υπάρχει, διαγράφουμε το αρχείο από τα gria resources, χρησιμοποιώντας το **conversation-id** και το gria client api.

StartFunctionExecution(mobileIp, functionurl,inputFileName)

Αρχικά εξετάζουμε αν η συνάρτηση αυτή με τα συγκεκριμένα ορίσματα υπάρχει στην βάση δεδομένων jobDatabase, δηλαδή αν εκτελείται εκείνη την στιγμή.

Αν δούμε ότι υπάρχει, τότε δεν ξεκινάμε την εκτέλεσή της και επιστρέφουμε ότι η συνάρτηση εκτελείται ήδη ή ότι έχει εκτελεστεί.

Αν δούμε ότι δεν υπάρχει τότε ξεκινάμε την εκτέλεση της συνάρτησης και κρατάμε το **conversation-id** καθώς επίσης και τον χρόνο (milliseconds) που ξεκίνησε η συνάρτηση και ότι η συνάρτηση μόλις άρχισε, και επιστρέφουμε ότι η συνάρτηση αυτή ξεκίνησε την εκτέλεσή της. Τα ορίσματα της συνάρτησης, αν αυτά προέρχονται από το κινητό μας τηλέφωνο τότε τα στέλνουμε στην συνάρτηση, ενώ αν αυτά προέρχονται από τον RemoteHardDisk, αφού τα εντοπίσουμε τον dataStager που βρίσκονται, με βάση το **conversation-id**, τα αντιγράφουμε από εκεί.

stopFunctionExecution(conversationId)

Χρησιμοποιώντας το conversationId εντοπίζουμε την συνάρτηση που εκτελείται και την σταματάμε.

- **Ask Mobile Server**

Μας παρέχει την συνάρτηση searchForFile(filename) η οποία ρωτάει όλα τα κινητά τηλέφωνα αν έχουν το συγκεκριμένο αρχείο.

Για να το πετύχουμε αυτό, κρατάμε στον server 1 μία λίστα με τα ip των κινητών τηλεφώνων. Για καθένα ip που βρίσκονται στην λίστα αυτή, στέλνουμε ένα ερώτημα ρωτώντας αν το συγκεκριμένο κινητό τηλέφωνο έχει το ζητούμενο αρχείο. Όταν βρούμε κάποιο κινητό τηλέφωνο με το ζητούμενο αρχείο σταματάμε την αναζήτηση.

Προφανώς δεν πρέπει να ψάξουμε στο δικό μας τηλέφωνο. Επειδή όμως έχουμε μόνο ένα κινητό τηλέφωνο , επιτρέπουμε η αναζήτηση να περιλαμβάνει και το δικό μας κινητό τηλέφωνο (διαφορετικά δεν θα έβρισκε τίποτα).

Μέρος Β

Στην ενότητα αυτή θα εξετάσουμε τον τρόπο υλοποίησης των τμημάτων main program και my services. Τα τμήματα αυτά θα τον εξετάσουμε παράλληλα, καθώς τα δύο αυτά τμήματα συνεργάζονται μεταξύ τους και για κάθε ενέργεια που ζητάμε με το MobileApplication υπάρχει ένα service στο myServices που ανταποκρίνεται σ' αυτήν.

Πριν συνεχίσουμε την ανάλυσή μας, θα θυμηθούμε ποιος είναι ο σκοπός του συστήματός μας. Όπως περιγράψαμε και στην ενότητα 2, σκοπός του συστήματός μας είναι να μπορούμε από το κινητό μας τηλέφωνο να κάνουμε τα παρακάτω:

- Να ανεβάσουμε κάποιο αρχείο στον RemoteHardDisk
 - Να κατεβάσουμε ένα αρχείο από τον RemoteHardDisk
 - Να βρίσκουμε και να κατεβάζουμε ένα αρχείο που βρίσκεται σε κάποιο άλλο κινητό τηλέφωνο.
 - Να καλούμε υπηρεσίες οι οποίες παίρνουν ως είσοδο αρχεία τα οποία βρίσκονται είτε στο κινητό μας τηλέφωνο είτε στον RemoteHardDisk.
-

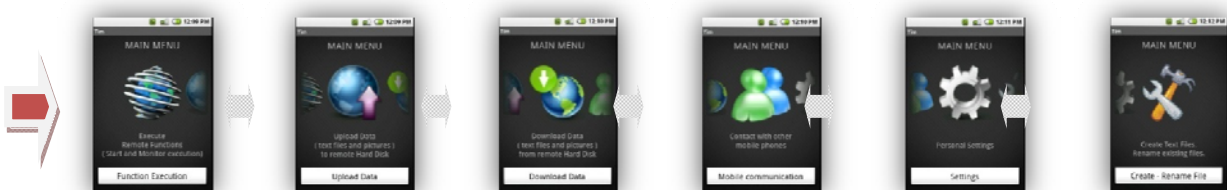
Τώρα θα δούμε πως έχουν υλοποιηθεί τα τμήματα main program και my services ώστε να πετύχουμε τους παραπάνω στόχους.

Αρχικά, όταν ξεκινάει η εφαρμογή μας στο κινητό τηλέφωνο, κάνουμε τις παρακάτω ενέργειες:

- Ξεκινάμε τον server του κινητού τηλεφώνου.
- Κατεβάζουμε πληροφορίες για τις υπάρχουσες υπηρεσίες.
- Στέλνουμε την ip μας (ή δεν στέλνουμε) στον server 1 , για να είμαστε (ή να μην είμαστε) ορατοί στα υπόλοιπα κινητά τηλέφωνα.

Για να μην σταματά η εκτέλεση της εφαρμογής μας, για την υλοποίηση των παραπάνω ενεργειών, έχουμε βάλει τον κώδικα για κάθε ενέργεια σε ένα νήμα, τα οποία και ξεκινάμε κατά την εκκίνηση της εφαρμογής μας.

Αφού λοιπόν ξεκινήσουμε την εκτέλεση των παραπάνω νημάτων μπαίνουμε στο βασικό μενού της εφαρμογής μας.



Όπως φαίνεται και από το παραπάνω σχήμα, το βασικό μενού της εφαρμογής μας, μας δίνει την δυνατότητα να διαλέξουμε επιλέξουμε τι θα θέλαμε να κάνουμε. Για την υλοποίηση του χρησιμοποιήσαμε τα στοιχεία Gallery , TextView και Button.

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10px">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="26px"
        android:textAlign="center"
        android:text="MAIN MENU\n"/>

    <Gallery
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/gallery"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:gravity="center_vertical"
        android:spacing="5" />

    <TextView
        android:id="@+id/info"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="20px"
        android:textAlign="center"
        android:text="\n\nSome information about\nthis selections\n"/>

    <Button
        android:id="@+id/selected"
        android:textSize="22px"
        android:textAlign="center"
        android:text="MyButton"
        android:layout_width="fill_parent"
        android:layout_height="50px" />

</LinearLayout>
```

Καθώς αλλάζει η εικόνα που βλέπουμε στην οθόνη του κινητού τηλεφώνου, προσαρμόζεται ανάλογα και κείμενο του TextView και Button.

Στην συνέχεια της ανάλυσής μας θα εξηγήσουμε πως έχουν υλοποιηθεί καθεμιά από τις ενέργειες που μας παρέχονται από το κεντρικό μενού.

Κάθε φορά που επιλέγουμε μια ενέργεια από το κεντρικό μενού, πηγαίνουμε σε μια νέα σελίδα. Το γραφικό περιβάλλον της σελίδας αυτής το έχουμε δημιουργήσει με xml αρχεία και τον χειρισμό των συμβάντων σε κάθε σελίδα με java classes.

➤ Upload Data

MobileApplication

Αρχικά επιλέγουμε το αρχείο που θέλουμε να μεταφέρουμε (από έναν Spinner).

Στην συνέχεια, έχουμε την δυνατότητα είτε να διαγράψουμε το αρχείο από το κινητό μας είτε να το ανεβάσουμε στον RemoteHardDisk.

Αν επιλέξουμε να διαγράψουμε το αρχείο, απλά το σβήνουμε από το κινητό μας και ανανεώνουμε το περιεχόμενο του Spinner.

Αν επιλέξουμε να ανεβάσουμε το αρχείο, τότε παίρνουμε τα byte του αρχείου, τα κωδικοποιούμε με βάση το 64 έτσι ώστε να μην τροποποιηθούν κατά την μεταφορά και δημιουργούμε ένα νέο string με τα νέα bytes. Στην συνέχεια καλούμε το service uploadFile του server 1 στέλνοντάς του το ip του κινητού μας, το όνομα του αρχείου και το string που δημιουργήσαμε.

myServices

Το service uploadFile, παίρνει τα bytes από το string που στείλαμε , τα αποκωδικοποιεί με βάση το 64 και αποθηκεύει το αποτέλεσμα σε ένα αρχείο στον server 1. Στη συνέχεια, ανεβάζει το αρχείο αυτό χρησιμοποιώντας την συνάρτηση upload του πακέτου Access gria resources και επιστρέφει ok αν όλα πάνε καλά ή error αν προκύψει κάποιο σφάλμα.

➤ Download Data

MobileApplication

Αρχικά κατεβάζουμε από τον server 1 τα ονόματα από τα αρχεία κειμένου ή εικόνας (ανάλογα με τι θέλουμε να κατεβάσουμε) που έχουμε στον RemoteHardDisk και τα βάζουμε σε έναν Spinner.

Έπειτα διαλέγουμε το αρχείο που μας ενδιαφέρει. Έχουμε την δυνατότητα είτε να κατεβάσουμε το αρχείο από τον RemoteHardDisk στο κινητό μας, είτε να το διαγράψουμε από τον RemoteHardDisk.

Αν επιλέξουμε να κατεβάσουμε το αρχείο στο κινητό μας τηλέφωνο, τότε καλούμε το service downloadFile του server 1 στέλνοντάς του το ip του κινητού μας και το όνομα του αρχείου που θέλουμε να κατεβάσουμε.

myServices

Το service downloadFile, κατεβάζει το ζητούμενο αρχείο χρησιμοποιώντας την συνάρτηση download του πακέτου Access gria resources, το αποθηκεύει τοπικά και ενημερώνει το κινητό τηλέφωνο ότι το αρχείο βρίσκεται στον server 1.

MobileApplication

Στη συνέχεια κατεβάζουμε το αρχείο από τον server 1 , χρησιμοποιώντας το url του αρχείου και στέλνουμε μήνυμα στον server 1 να διαγράψει το αρχείο.

Αν επιλέξουμε να διαγράψουμε το αρχείο από τον RemoteHardDisk, τότε καλούμε το service deleteGriaFile του server 1 στέλνοντάς του το ip του κινητού μας και το όνομα του αρχείου που θέλουμε να διαγράψουμε.

myServices

Το service deleteGriaFile, διαγράφει το ζητούμενο αρχείο χρησιμοποιώντας την συνάρτηση delete του πακέτου Access gria resources και επιστρέφει ok αν όλα πάνε καλά ή error αν προκύψει κάποιο σφάλμα.

➤ **Mobile communication**

MobileApplication

Αρχικά , αφού δώσουμε το όνομα του αρχείου που ψάχνουμε, καλούμε το service search στέλνοντάς του το όνομα του αρχείου που ψάχνουμε.

myServices

Το service search, ψάχνει σε όλα τα κινητά (τα ορατά...) για το συγκεκριμένο αρχείο και επιστρέφει το ip του κινητού που έχει το αρχείο , αν αυτό βρεθεί σε κάποιο κινητό τηλέφωνο, διαφορετικά επιστρέφει ότι δεν βρήκε τίποτα.

MobileApplication

Στη συνέχεια, χρησιμοποιώντας το ip του κινητού (αν βέβαια το αρχείο έχει βρεθεί) και την συνάρτηση askQuestion του πακέτου Mobile TCP Client κατεβάζουμε το αρχείο.

➤ **Function Execution**

Start function execution

MobileApplication

Αρχικά επιλέγουμε την συνάρτηση που θέλουμε να εκτελέσουμε από έναν Spinner. Ανάλογα με την συνάρτηση που επιλέγουμε, προσαρμόζεται και το γραφικό περιβάλλον σύμφωνα με χαρακτηριστικά της συνάρτησης (περιγραφή συνάρτησης, αριθμός εισόδων, αριθμός εξόδων).

Έχουμε την δυνατότητα να επιλέξουμε, για κάθε όρισμα της συνάρτησης ξεχωριστά, αν θα το ανεβάσουμε από το κινητό μας ή θα το πάρουμε από τον RemoteHardDisk.

Αφού επιλέξουμε την συνάρτηση που θέλουμε να εκτελέσουμε, τα ονόματα των αρχείων εισόδου της συνάρτησης και που βρίσκεται το καθένα από αυτά, ξεκινάμε την εκτέλεσή της. Για να το πετύχουμε αυτό, αρχικά ανεβάζουμε στον server 1 τα αρχεία που έχουμε επιλέξει να πάρουμε από το κινητό μας τηλέφωνο και έπειτα καλούμε το service startFunctionExecution του server 1 στέλνοντάς του ένα xml-string που περιέχει πληροφορίες για την συνάρτηση, όπως το url της συνάρτησης, τα ονόματα των ορισμάτων της συνάρτησης καθώς και από πού θα πάρουμε καθένα από τα ορίσματά της.

myServices

Το service startFunctionExecution, αναλύει το xml-string που στείλαμε χρησιμοποιώντας τις συναρτήσεις και κλάσεις που έχουμε φτιάξει για τον χειρισμό των xml αρχείων (μιας και δεν υπάρχει λόγος να χρησιμοποιήσουμε κάτι πιο πολύπλοκο). Στη συνέχεια ξεκινά την εκτέλεση της συνάρτησης με το δοθέν url, δίνοντάς της πληροφορίες για τα ονόματα των αρχείων εισόδου καθώς και πού βρίσκεται το καθένα (στον server1 ή στα gria resources).

Όταν ολοκληρωθεί η εκτέλεση της συνάρτησης, αποθηκεύουμε το αποτέλεσμα στα gria resources. Επίσης διαγράφουμε από τον server 1 τα αρχεία που είχαμε ανεβάσει από το κινητό τηλέφωνο.

Monitor function execution

Αφού ξεκινήσουμε την εκτέλεση μιας συνάρτησης, έχουμε την δυνατότητα να παρακολουθήσουμε την εκτέλεσή της, δηλαδή αν έχει τελειώσει ή αν έχει προκύψει κάποιο σφάλμα κατά την εκτέλεση καθώς επίσης και πόσος χρόνος έχει περάσει από την στιγμή έναρξης της εκτέλεσης.

Refresh function execution status

MobileApplication

Για να δούμε σε τι κατάσταση βρίσκεται η εκτέλεση των συναρτήσεων, καλούμε το service getRunningFunctionsInfo στέλνοντάς του το ip μας.

myServices

Το service getRunningFunctionsInfo, επιστρέφει για κάθε συνάρτηση που εκτελείται το conversation-id, το όνομά της, τα ονόματα των ορισμάτων της, και τον χρόνο που έχει περάσει. Τα παραπάνω τα επιστρέφει σαν ένα string και η πληροφορία για κάθε συνάρτηση χωρίζονται με τον χαρακτήρα '#'.
#

Stop function execution

MobileApplication

Για να σταματήσουμε την εκτέλεση μιας συνάρτησης, καλούμε το service stopFunctionExecution στέλνοντάς του το conversation-id της συνάρτησης που τρέχει.

myServices

Το service stopFunctionExecution, χρησιμοποιώντας το conversation-id της συνάρτησης που εκτελείται, αφού εντοπίσει την συνάρτηση που εκτελείται, την σταματά. Αν όλα πάνε καλά και καταφέρει να σταματήσει την εκτέλεση της συνάρτησης επιστρέφει ok διαφορετικά (αν η συνάρτηση προλάβει και ολοκληρώσει την εκτέλεσή της) error.

➤ **Settings**

MobileApplication

Εδώ έχουμε την δυνατότητα να ρυθμίσουμε τα εξής:

- ✓ Αν θέλουμε το κινητό μας να είναι ορατό από τα άλλα κινητά.

Αν επιλέξουμε το κινητό μας να είναι ορατό , τότε καλούμε το service sendIp, στέλνοντάς του το ip του κινητού μας και αυτό το κρατάει σε μια λίστα.

Αν επιλέξουμε το κινητό μας να μην είναι ορατό , τότε καλούμε το service removeIp, στέλνοντάς του το ip μας και αυτό το διαγράφει από την λίστα.

- ✓ Αν θέλουμε όταν ανεβάζουμε ένα αρχείο στον RemoteHardDisk να το διαγράφουμε από το κινητό μας.
- ✓ Αν θέλουμε όταν κατεβάζουμε ένα αρχείο από τον RemoteHardDisk να το διαγράφουμε από αυτόν.

Για τα δύο τελευταία, έχουμε δύο GLOBAL μεταβλητές τύπου Boolean και ανάλογα τις οποίες συμβουλευόμαστε κάθε φορά που θέλουμε να κάνουμε upload ή download και πράττουμε ανάλογα.

Τις παραπάνω πληροφορίες αυτές τις κρατάμε σε ένα αρχείο στο κινητό μας τηλέφωνο, έτσι ώστε να μην χρειάζεται κάθε φορά να τις ρυθμίζουμε.

➤ Create – Rename File

MobileApplication

Εδώ έχουμε δημιουργήσει ένα γραφικό περιβάλλον για να μπορούμε να δημιουργούμε εύκολα αρχεία κειμένου, αλλά και για να μπορούμε να αλλάζουμε το όνομα των αρχείων που έχουμε στο κινητό μας τηλέφωνο.

Την επιλογή αυτή την δημιουργήσαμε για δική μας διευκόλυνση.

Ο κώδικας της εφαρμογής που δημιουργήσαμε στο κινητό τηλέφωνο αλλά και ο κώδικας των services που δημιουργήσαμε στον server 1 , υπάρχουν στο παράρτημα Β.

4

Mobile Grid

Όπως είδαμε στην προηγούμενη ενότητα, στο κινητό μας τηλέφωνο τρέχει ένας server , ο οποίος μπορεί να απαντά στα ερωτήματά μας προς αυτόν. Στην ενότητα αυτή, θα εξετάσουμε πως θα μπορούσαν τα κινητά τηλέφωνα, στα οποία τρέχει ο παραπάνω server, να συμβάλουν στην άντληση πληροφοριών από τα κινητά τηλέφωνα.

4.1 Mobile Server

Όπως ανφέραμε και στην εισαγωγή της ενότητας αυτής, στο κινητό τηλέφωνο τρέχει ένας server ο οποίος μπορεί να απαντά σε ερωτήματα της μορφής :

question : parameters

Στην υποενότητα αυτή θα δώσουμε περισσότερες πληροφορίες για τα ερωτήματα που μπορεί να απαντάει ο server αυτός. Αυτά είναι:

- ✓ search : file
Ψάχνει στο κινητό τηλέφωνο να δει αν έχει το ζητούμενο αρχείο και αν το βρει επιστρέφει ok, διαφορετικά ότι δεν το βρήκε.
- ✓ getFile : file
Στέλνει σε αυτόν που ζητάει από το κινητό τηλέφωνο το ζητούμενο αρχείο.
- ✓ supportedfunctions :
Επιστρέφει τα ονόματα των συναρτήσεων που υποστηρίζει ο server αυτού του κινητού τηλεφώνου.

- ✓ description : function

Επιστρέφει μια περιγραφή (δηλαδή τι κάνει) για την ζητούμενη συνάρτηση.

- ✓ execute : function

Ζητάμε την εκτέλεση της επιθυμητής συνάρτησης. Επειδή το κινητό τηλέφωνο έχει περιορισμένες δυνατότητες, για τον λόγο αυτό οι συναρτήσεις που υποστηρίζει δεν κάνουν μια πολύπλοκη εργασία, αλλά απλά επιστρέφουν ορισμένες πληροφορίες.

4.2 *Mobile Grid*

Λέγοντας Mobile Grid εννοούμε την δυνατότητα που έχουμε από το κινητό μας τηλέφωνο, αλλά και από τον υπολογιστή μας να έχουμε πρόσβαση στα άλλα κινητά τηλέφωνα και να ζητάμε την εκτέλεση εργασιών από αυτά. Ο κύριος λόγος που το θέλουμε αυτό δεν είναι για την εκτέλεση πολύπλοκων εργασιών αλλά για την άντληση σημαντικών πληροφοριών από αυτά.

Στην υποενότητα αυτή, θα δούμε δύο εφαρμογές οι οποίες χρησιμοποιούν τον server που τρέχει στα κινητά τηλέφωνα για την άντληση πληροφοριών από αυτά.

Στην πρώτη εφαρμογή θα δούμε πως μπορούμε να πάρουμε κάποιο αρχείο από κάποιο άλλο κινητό τηλέφωνο και στην δεύτερη εφαρμογή θα δούμε πως θα μπορούσαμε να ζητήσουμε κάποια πληροφορία από ένα κινητό τηλέφωνο.

Εφαρμογή 1^η

Όπως είδαμε στην ενότητα 3, στην εφαρμογή που φτιάξαμε στο κινητό τηλέφωνο έχουμε την δυνατότητα να πάρουμε ένα αρχείο κατευθείαν από κάποιο άλλο κινητό τηλέφωνο.

Για να το πετύχουμε αυτό χρησιμοποιούμε τον server 1 ο οποίος κρατάει τα ip από όλα τα κινητά τηλέφωνα που θέλουν να είναι ορατά στους υπόλοιπους χρήστες κινητών τηλεφώνων. Ο server αυτός στέλνει σε όλα τα ορατά κινητά το ερώτημα

search : filename

ρωτώντας κάθε κινητό αν έχει το επιθυμητό αρχείο. Στη συνέχεια όταν βρεί κάποιο κινητό τηλέφωνο με το ζητούμενο αρχείο, μας επιστρέφει το ip του κινητού τηλεφώνου που έχει το αρχείο και στη συνέχεια εμείς ζητάμε από το κινητό αυτό να μας στείλει το αρχείο, στέλνοντάς του το ερώτημα:

getfile : filename

Το άλλο κινητό , απαντώντας στο ερώτημά μας, μας στέλνει το αρχείο που ψάχνουμε.

Εφαρμογή 2^η (AskMobilePhone)

Μια άλλη εφαρμογή που φτιάξαμε τρέχει σε κάποιον server (όχι απαραίτητα στον server1) και μας δίνει την δυνατότητα να αντλούμε πληροφορίες από το κινητό τηλέφωνο. Την εφαρμογή αυτή δεν την δημιουργήσαμε για να ζητάμε κάποιο αρχείο από το κινητό τηλέφωνο (όπως κάναμε στην προηγούμενη εφαρμογή), αλλά για να μπορούμε να ζητάμε πληροφορίες όπως σε ποια περιοχή βρίσκεται το κινητό τηλέφωνο.

Για να το πετύχουμε αυτό, αρχικά ζητάμε από το κινητό τηλέφωνο να μας στείλει τα ονόματα των συναρτήσεων που υποστηρίζει.

supportedfunctions :

Αν το όνομα της συνάρτησής μας (έστω ότι η ζητούμενη συναρτηση είναι η phoneplace) είναι μέσα σε αυτές που υποστηρίζει τότε ζητάμε την εκτέλεση της.

execute : phoneplace

Η απάντηση που μας στέλνει είναι η περιοχή που βρίσκεται το κινητό.

Την παραπάνω εργασία μπορούμε να την επαναλάβουμε για όλα τα ορατά κινητά τηλέφωνα.

Ο κώδικας της εφαρμογής αυτής βρίσκεται στο Παράρτημα Β.

5

Τα workflows που δημιουργήσαμε

Στην ενότητα αυτή θα περιγράψουμε τα workflows που δημιουργήσαμε. Αρχικά θα δούμε ποιες υπηρεσίες υπάρχουν ήδη και στη συνέχεια θα δείξουμε πώς χρησιμοποιήσαμε τις υπάρχουσες υπηρεσίες για την δημιουργία των workflow μας.

5.1 Υπάρχουσες υπηρεσίες

Στην υποενότητα αυτή θα παρουσιάσουμε τις υπάρχουσες/διαθέσιμες υπηρεσίες. Οι παρόντες υπηρεσίες μας προσφέρονται δωρεάν. Στις υπηρεσίες που ακολουθούν παρουσιάσαμε μόνο τις πληροφορίες εκείνες που θα μας χρειαστούν στη συνέχεια για την δημιουργία των workflow.

❖ **Paint**

Περιγραφή : Application to render an image into painting

Αριθμός εισόδων : 1

Αριθμός εξόδων : 1

❖ **Swirl**

Περιγραφή: Application to swirl an image

Αριθμός εισόδων : 1

Αριθμός εξόδων : 1

❖ Blend

Περιγραφή: Blends two images together

Αριθμός εισόδων : 2

Αριθμός εξόδων : 1

Τις παραπάνω πληροφορίες τις βρήκαμε από το wsdl αρχείο της κάθε συνάρτησης με την βοήθεια του GRIA.

5.2 Workflows

Το Taverna είναι ένα πρόγραμμα που μας επιτρέπει εύκολα να δημιουργούμε workflows (ροές εργασιών). Περισσότερες πληροφορίες για το taverna υπάρχουν στο παράρτημα Α.

Με την βοήθεια του Taverna, δημιουργήσαμε τα workflows που χρησιμοποιήσαμε στην εργασία αυτή.

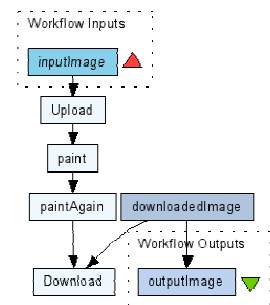
❖ Workflow 01

Περιγραφή : Paint image two times

Αριθμός εισόδων : 1

Αριθμός εξόδων : 1

Το workflow 01 είναι μια νέα υπηρεσία η οποία χρησιμοποιεί την υπηρεσία paint κατά την διάρκεια της εκτέλεσής της. Πιο συγκεκριμένα, ζωγραφίζει την δοθείσα εικόνα χρησιμοποιώντας την υπηρεσία paint και την εικόνα που προκύπτει την χανα ζωγραφίζει χρησιμοποιώντας πάλι την υπηρεσία paint.



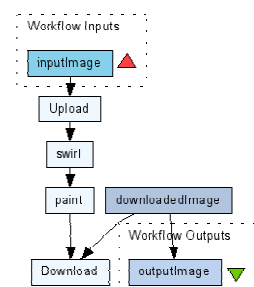
❖ Workflow 02

Περιγραφή : Swirl and after paint an image

Αριθμός εισόδων : 1

Αριθμός εξόδων : 1

Το workflow 02 είναι επίσης μια νέα υπηρεσία η οποία χρησιμοποιεί την υπηρεσία swirl και την paint κατά την διάρκεια της εκτέλεσής της. Πιο συγκεκριμένα, αρχικά χρησιμοποιεί την υπηρεσία swirl και την εικόνα που προκύπτει την ζωγραφίζει χρησιμοποιώντας την υπηρεσία paint.



6

Παραδείγματα εκτέλεσης – Αξιολόγηση συστήματος

Στην ενότητα αυτή, αρχικά θα καταγράψουμε τον χρόνο που απαιτείται για την μεταφορά ορισμένων αρχείων κειμένου και εικόνας από το κινητό μας τηλέφωνο στα gria resources και αντίστροφα. Επίσης θα καταγράψουμε τον χρόνο που απαιτείται για την εκτέλεση ορισμένων υπηρεσιών καθώς και των workflows (ροές εργασιών) που δημιουργήσαμε. Στη συνέχεια θα αξιολογήσουμε τα αποτελέσματα και θα εξετάσουμε, αν και κατά πόσο το σύστημά μας βοηθάει στην επέκταση των δυνατοτήτων ενός κινητού τηλεφώνου.

6.1 Παραδείγματα εκτέλεσης

Όπως ανέφερα λίγο πιο πρίν, σε αυτήν την υποενότητα θα εξετάσουμε τον χρόνο που χρειάζεται για την ολοκλήρωση των παρακάτω εργασιών:

❖ Upload Data

Μεταφορά δεδομένων (αρχείων κειμένου και εικόνας) από το κινητό μας τηλέφωνο στα gria resources. Υπενθυμίζουμε ότι τα αρχεία μας μεταφέρονται από το κινητό μας στον server 1 και εν συνεχεία στα gria resources. Όταν η μεταφορά ολοκληρωθεί, τα αρχεία διαγράφονται από τον server 1.

❖ Download Data

Μεταφορά δεδομένων (αρχείων κειμένου και εικόνας) από τα gria resources στο κινητό μας τηλέφωνο. Υπενθυμίζουμε ότι τα αρχεία μας μεταφέρονται από τα gria resources στον server 1

και εν συνεχεία στο κινητό μας τηλέφωνο και όταν ολοκληρωθεί η μεταφορά διαγράφουμε το αρχείο από τον server 1.

❖ Delete Data

Διαγραφή αρχείων τα οποία εμείς είχαμε ανεβάσει (κάποια στιγμή στο παρελθόν) στα gria resources.

❖ Function / Service execution

Εκτέλεση εργασιών μακριά από το κινητό μας τηλέφωνο. Εδώ, αρχικά θα παρουσιάσουμε τον χρόνο που απαιτείται για την ολοκλήρωση εργασιών που εκτελούνται σε έναν μόνο server και στην συνέχεια θα δούμε τον χρόνο που απαιτείται για την ολοκλήρωση εργασιών που δεν μπορούν να εκτελεστούν εξολοκλήρου σε έναν μόνο server. Πρόκειται για τα workflows που δημιουργήσαμε, τα οποία περιγράψαμε στην προηγούμενη ενότητα, τα οποία όπως είδαμε αποτελούνται από την σειριακή σύνδεση δύο υπηρεσιών, ανεξάρτητων μεταξύ τους, που θα μπορούσαν να βρίσκονται σε διαφορετικούς υπολογιστές / servers.

Τα αρχεία τα οποία έχουμε στο κινητό μας και τα οποία θα χρησιμοποιήσουμε στις παραπάνω εργασίες είναι αρχεία κειμένου (.txt) και αρχεία εικόνας (.png) .

Ο χρόνος που έχουμε καταγράψει και φαίνεται στους παρακάτω πίνακες, μας δείχνει πόσο περίπου χρονικό διάστημα απαιτήθηκε για την ολοκλήρωση της αντίστοιχης εργασίας. Θα πρέπει να λάβουμε υπόψη ότι ο χρόνος αυτός εξαρτάται από την ταχύτητα σύνδεσής μας στο ιντερνέτ καθώς και από τις εργασίες που εκτελούνται στο σύστημά μας εκείνη την στιγμή.

Στους πίνακες που ακολουθούν έχουμε καταγράψει (εκτός βέβαια από το όνομα του αρχείου το οποίο μας δείχνει και αν είναι αρχείο κειμένου ή εικόνας) το μέγεθος του αρχείου και το χρονικό διάστημα που απαιτήθηκε για την ολοκλήρωση της αντίστοιχης ενέργειας.

• Upload Data

File	Size (bytes)	Time (milliseconds)
Servlet_technology.txt	176	592
Java_history.txt	342	612
American_Sarkozy.txt	479	606
Silvio_Berlusconi.txt	571	598
Settings.png	5474	742
Football.png	11473	617
Zoom.png	12890	837
Basketball.png	13252	886
Play.png	81898	994

- **Download Data**

File	Size (bytes)	Time (milliseconds)
Servlet_technology.txt	176	1936
Java_history.txt	342	1969
American_Sarkozy.txt	479	1974
Silvio_Berlusconi.txt	571	1868
Settings.png	5474	1817
Football.png	11473	1769
Zoom.png	12890	1709
Basketball.png	13252	2023
Play.png	81898	2530

- **Delete Data**

File	Size (bytes)	Time (milliseconds)
Servlet_technology.txt	176	1060
Java_history.txt	342	1122
American_Sarkozy.txt	479	1030
Silvio_Berlusconi.txt	571	1007
Settings.png	5474	1029
Football.png	11473	1080
Zoom.png	12890	998
Basketball.png	13252	1151
Play.png	81898	1079

- **Function Execution**

Function name	Input File	Size (bytes)	Time (milliseconds)
---------------	------------	----------------	-----------------------

(Οι υπηρεσίες που υπάρχουν ήδη)

Paint	Football.png	11473	17156
	Basketball.png	13252	18467
	Play.png	81898	19603

Swirl	Football.png	11473	17156
	Basketball.png	13252	17467
	Play.png	81898	18603

Blend	Football.png	11473	19600
	Basketball.png	13252	
	Settings.png	5474	19391
	Zoom.png	12890	
	Zoom.png	12890	18834
	Football.png	11473	

(Τα workflow που δημιουργήσαμε)

Workflow 01 (paint ->paint)	Football.png	11473	47406
	Basketball.png	13252	39195
	Play.png	81898	46587

Workflow 02 (swirl -> paint)	Football.png	11473	39073
	Basketball.png	13252	38518
	Play.png	81898	44904

6.2 Αξιολόγηση του συστήματός μας

Στην υποενότητα αυτή θα αξιολογήσουμε τα αποτελέσματα που καταγράψαμε στην προηγούμενη υποενότητα.

Χρόνος εκτέλεσης εργασιών/υπηρεσιών.

Σύμφωνα με τα αποτελέσματα (δηλαδή την χρονική διάρκεια εκτέλεσης των εργασιών μας) που καταγράψαμε λίγο πιο πριν, ο χρόνος μεταφοράς των δεδομένων από το κινητό μας τηλέφωνο στα gria resources είναι πολύ μικρότερος από τον χρόνο που απαιτείται για την ολοκλήρωση των εργασιών paint, swirl και blend. Επίσης, αν κοιτάξουμε προσεκτικά τον χρόνο εκτέλεσης των εργασιών workflow 01 και workflow 02 αλλά και τον χρόνο εκτέλεσης των εργασιών paint και swirl τις οποίες χρησιμοποιήσαμε για την δημιουργία των workflows, θα παρατηρήσουμε ότι ο χρόνος μεταφοράς των ενδιάμεσων αποτελεσμάτων από την μία υπο-εργασία στην άλλη είναι πολύ μικρός και δεν αυξάνει σημαντικά τον συνολικό χρόνο ολοκλήρωσης των workflows.

Ακόμη, κάτι που ισχύει, αλλά δεν φαίνεται στον χρόνο που καταγράψαμε στα προηγούμενα παραδείγματα, είναι ότι από την στιγμή που ξεκινήσουμε την εκτέλεση μιας εργασίας, δεν χρειάζεται να περιμένουμε μέχρι την ολοκλήρωσή της αλλά έχουμε την δυνατότητα να ξεκινήσουμε την εκτέλεση και άλλων εργασιών ή ακόμη να βγούμε από την εφαρμογή μας και να κλείσουμε το κινητό! Οι εργασίες αυτές θα συνεχίσουν να εκτελούνται και όταν ολοκληρωθούν θα βάλουν τα αποτελέσματα στα gria resource (στα οποία έχουμε πρόσβαση μόνο ο κάτοχος του συγκεκριμένου τηλεφώνου) για να μπορούμε έπειτα να τα κατεβάσουμε.

Είναι λοιπόν ξεκάθαρο, ότι με το σύστημά μας έχουμε την δυνατότητα να εκτελούμε πολύπλοκες εργασίες, που θα απαιτούσαν αρκετό χρονικό διάστημα αν εκτελούνταν στο κινητό τηλέφωνο (ή και ενδεχομένως να μην μπορούσαν να εκτελεστούν σε αυτό) μακριά από αυτό. Οι υπηρεσίες αυτές, είναι δυνατόν να αποτελούνται από την σειριακή ή παράλληλη σύνδεση περισσότερων από μία υπηρεσιών.

Σύμφωνα λοιπόν με τα παραπάνω, έχουμε την δυνατότητα να καλούμε από το κινητό μας τηλέφωνο υπηρεσίες που αποτελούνται από τον συνδυασμό άλλων υπηρεσιών. Ο χρόνος που χρειάζεται να περιμένουμε είναι ο χρόνος για την μεταφορά των απαραίτητων αρχείων στα gria resources. Επειτα δεν χρειάζεται να περιμένουμε άλλο. Μπορούμε να κάνουμε οποιαδήποτε άλλη εργασία και απλά όταν ολοκληρωθούν τα αποτελέσματα μπορούμε να τα κατεβάσουμε.

Κόστος εκτέλεσης εργασιών/υπηρεσιών.

Ένας άλλος παράγοντας που πρέπει να εξετάσουμε είναι το κόστος για την εκτέλεση μιας υπηρεσίας/εργασίας/προγράμματος. Πολλές φορές χρειάζεται να χρησιμοποιήσουμε μια υπηρεσία μία μόνο φορά (ή λίγες φορές). Σε αυτήν την περίπτωση δεν θα είχε νόημα να αγοράσουμε ολόκληρη την υπηρεσία αλλά θα θέλαμε απλώς να χρησιμοποιήσουμε την υπηρεσία και να πληρώσουμε το αντίστοιχο κόστος φυσικά. Με το σύστημά μας έχουμε αυτήν την δυνατότητα, να χρησιμοποιήσουμε μια υπηρεσία χωρίς να χρειάζεται να την αποκτήσουμε. Βέβαια εμείς στην άσκησή μας περιοριστήκαμε στην χρησιμοποίηση υπηρεσιών που προσφέρονται δωρεάν, αλλά σχετικά εύκολα θα μπορούσαμε να το τροποποιήσουμε έτσι ώστε να δουλεύει και με υπηρεσίες που δεν είναι δωρεάν.

Τρόπος υλοποίησης των εργασιών/υπηρεσιών.

Ένα άλλο πολύ σημαντικό πλεονέκτημα που έχουμε με το σύστημά μας είναι ότι οι υπηρεσίες μπορούν να αλλάζουν τον τρόπο υλοποίησής τους, χωρίς αυτό να μας επηρεάζει. Το πλεονέκτημα αυτό είναι ακόμα πιο σημαντικό στην περίπτωση των workflows, καθώς μπορούμε να βελτιώσουμε τις επιμέρους υπηρεσίες, από τις οποίες αποτελείται το workflow μας, την μία ανεξάρτητα από την άλλη ή ακόμη μπορούμε να αντικαταστήσουμε κάποια από αυτές, αν δούμε ότι είναι προβληματική.

Αν την υπηρεσία την είχαμε στο κινητό μας τηλέφωνο (υποθέτοντας ότι θα μπορούσε να τρέξει σε αυτό), θα έπρεπε κάθε φορά που η εταιρία άλλαζε την υλοποίηση της υπηρεσία να κατεβάζουμε και να εγκαθιστούμε στο κινητό μας την νέα έκδοση.

Προσθήκη νέων υπηρεσιών

Ένα ακόμη πλεονέκτημα που έχουμε με το σύστημά μας, είναι ότι μπορούμε να ερημωνόμαστε αυτόματα για την προσθήκη και τον τρόπο χρήσης νέων υπηρεσιών. Αυτό είναι αρκετά σημαντικό καθώς οι ανακάλυψη υπηρεσιών/προγραμμάτων είναι δύσκολη ακόμα και για αυτούς που ασχολούνται με τον προγραμματισμό, πόσο μάλλον για τους απλούς χρήστες των κινητών τηλεφώνων.

Τα πλεονεκτήματα λοιπόν που έχουμε με το σύστημά μας είναι πολλά και σημαντικά καθώς αυξάνουν τις επιδόσεις του κινητού τηλεφώνου και του δίνους την δυνατότητα – αρα και σε μας που το χρησιμοποιούμε – να προσαρμόζεται καλύτερα στις εξελίξεις.

7

Επίλογος

7.1 Σύνοψη και συμπεράσματα

Όπως είδαμε στην εργασία αυτή, με το σύστημά που δημιουργήσαμε έχουμε την δυνατότητα να εκτελούμε εύκολα πολύπλοκες εργασίες μακριά από το κινητό τηλέφωνο. Η δυνατότητα αυτή αποδεσμεύει το κινητό τηλέφωνο από την εκτέλεση ορισμένων εργασιών που θα ήταν χρονοβόρες και θα επιβάρυναν την λειτουργία του καθώς από την στιγμή εκκίνησης των εργασιών το κινητό τηλέφωνο παύει να ασχολείται με αυτές παρα μόνο όταν χρειαστούμε να δούμε σε τι κατάσταση βρίσκονται ή αν επιλέξουμε να σταματίσουμε ορισμένες από αυτές επειδή καθυστερούν. Όμως το κινητό τηλέφωνο εκτός από την δυνατότητα αυτή, μπορεί να χρησιμεύσει για να παρέχει διάφορες χρήσιμες πληροφορίες όπως είδαμε. Η ένταξη των κινητών στο grid έχει πλέον αρχίσει και αναμένεται να ολοκληρωθεί στα επόμενα χρόνια.

8

Σχετικά Links

Android	http://code.google.com/android/index.html
Gria	http://www.gria.org/
Axis	http://ws.apache.org/axis/
Taverna	http://taverna.sourceforge.net/
Apache Tomcat	http://tomcat.apache.org/
Eclipse	http://www.eclipse.org/
Java	http://java.sun.com/
Wikipedia	http://en.wikipedia.org/wiki/Main_Page
other useful sites	http://www.anddev.org/
	http://www.detz.net/android
	http://blog.pocketjourney.com/
	http://java.dzone.com/articles/google-android-tutorial

9

Παράρτημα Α

9.1 Workflow - Taverna

Workflow

Όπως είδαμε και στην εισαγωγή, workflow είναι μια ακολουθία από εργασίες/λειτουργίες τις οποίες συνδιάζουμε για να δημιουργήσουμε μιας πιο πολύπλοκης εργασίας. Το workflow μας δίνει την δυνατότητα να χειριζόμαστε ένα σύνολο εργασιών, που συνεργάζονται για την επίτευξη μιας εργασίας, σαν μία ενότητα. Απ' την άλλη το γεγονός ότι αποτελείται από ένα σύνολο εργασιών με καθορισμένο ρόλο η κάθε μία, μας βοηθάει να αντιληφθούμε καλύτερα τον τρόπο υλοποίησης της εργασίας/workflow μας.

Taverna

Το Taverna είναι ένα πρόγραμμα που μας επιτρέπει να δημιουργούμε και να εκτελούμε εύκολα workflows. Τα συστατικά μέρη κάθε workflow μπορεί να βρίσκονται είτε στον ίδιον υπολογιστή/server είτε σε κάποιον άλλον υπολογιστή/server.

Με το γραφικό περιβάλλον που διαθέτει, μας βοηθάει στην κατανόηση της λειτουργίας κάθε workflow, ιδιαίτερα όταν αυτή είναι πολύπλοκη.

Κατά την διάρκεια εκτέλεσης ενός workflow μας δίνει την δυνατότητα να παρακολουθούμε την εκτέλεση και να ενημερωνόμαστε για τυχόν λάθη.

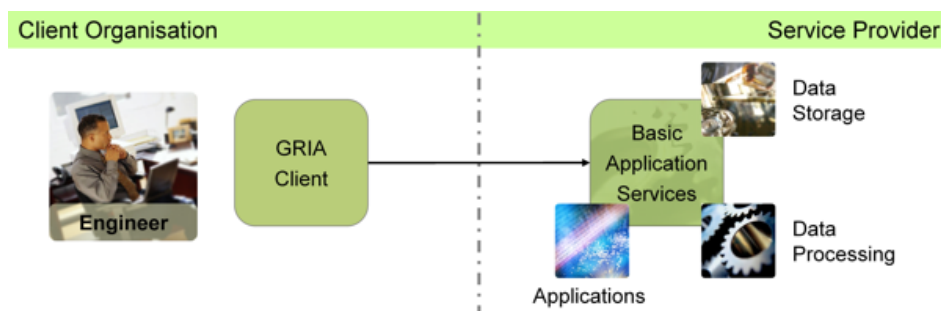
Το Taverna είναι γραμμένο σε JAVA 5 και τρέχει σε διάφορα λειτουργικά συστήματα όπως τα Windows, Linux, OSX και Unix.

9.2 Gria

Το Gria είναι ένα λογισμικό που συνδέει διάφορες υπηρεσίες και πόρους έτσι ώστε να μην χρειάζεται να γνωρίζουμε που βρίσκονται οι πόροι και οι υπηρεσίες.

Οι δυνατότητες που προσφέρει το Gria είναι πολλές. Εμείς στην εργασία αυτή περιοριστήκαμε στην χρησιμοποίηση του gria για την διασύνδεση μας με υπηρεσίες που εμπιστευόμαστε.

Όπως φαίνεται και στο παρακάτω σχήμα στην εργασία μας χρειαστήκαμε μόνο το gria client και το gria basic application services προκειμένου να έχουμε πρόσβαση στα υπάρχοντα services.



Το παραπάνω σχήμα, το πήρα από το site του Gria.

Gria Client

Το gria client api μας προσφέρει low lever java objects για τον χειρισμό των job και data services.

Basic Application Services

Το Basic Application Services μας δίνει την δυνατοτητα να εκτελούμε τα data και job services.

10

Παράρτημα Β

Στο Παράρτημα Β έχουμε βάλει τον κώδικα της εφαρμογής μας στο κινητό τηλέφωνο καθώς και των κώδικα των services που δημιουργήσαμε.

10.1 Κώδικας εφαρμογής κινητού τηλεφώνου

Main program

➤ **main.java**

```
package google.android.packagel;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;

public class main extends Activity
{
    protected Button select;
    protected TextView info;

    @Override
    public void onCreate(Bundle icle)
    {
        super.onCreate(icle);

        GLOBAL.activity = this;

        // Start android server
        Intent i = new Intent();
        i.setClassName("google.android.packagel",
            "google.android.packagel.Server");

        Bundle b = new Bundle();
        startService( i,b );

        // Thread - Download functions informations
        Thread_FunctionsInformations mythread1 = new
            Thread_FunctionsInformations(this);
        mythread1.setName("Thread-FunctionsInformations");
        mythread1.setPriority(Thread.NORM_PRIORITY);
        mythread1.start();

        // Thread - Load Personal Settings
        Thread_PersonalSettings mythread2 = new
            Thread_PersonalSettings(this);
        mythread2.setName("Thread-PersonalSettings");
        mythread2.setPriority(Thread.NORM_PRIORITY);
        mythread2.start();

        // Main Menu
        new MainMenu(this);
    }
}
```

Main Menu



➤ main.xml

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10px">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="26px"
        android:textAlign="center"
        android:text="MAIN MENU\n"/>

    <Gallery xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/gallery"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:gravity="center_vertical"
        android:spacing="5" />

    <TextView
        android:id="@+id/info"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="20px"
        android:textAlign="center"
        android:text="\n\nSome information about\nthis selections\n"/>

    <Button
        android:id="@+id/selected"
        android:textSize="22px"
        android:textAlign="center"
        android:text="MyButton"
        android:layout_width="fill_parent"
        android:layout_height="50px" />

</LinearLayout>
```

➤ MainMenu.java

```
package google.android.package1;

import android.app.Activity;
```

```

import android.content.Context;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.Gallery;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.AdapterView.OnItemClickListener;

public class MainMenu
{
    private Activity activity;
    private Gallery gallery;
    private Button select;
    private TextView info;
    public MainMenu(Activity activ)
    {
        activity = activ;
        activity setContentView(R.layout.main);

        gallery = (Gallery) activity.findViewById(R.id.gallery);
        info = (TextView) activity.findViewById(R.id.info);
        select = (Button) activity.findViewById(R.id.selected);

        select.setOnClickListener(new selectButtonListener());

        gallery.setAdapter(new ImageAdapter(activity));
        gallery.setOnItemClickListener(new galleryListener());
    }
    private class ImageAdapter extends BaseAdapter
    {
        private Context mContext;
        private String[] mImageNames = { "Function Execution",
            "Upload Data",
            "Download Data",
            "Mobile communication",
            "Settings",
            "Create - Rename File"};
    }
}

```

```

        private String[] mImageDescrip =
    {
        "\nExecute\nRemote Functions\n( Start and Monitor execution)\n",
        "\nUpload Data\n( text files and pictures )\nto remote Hard Disk\n",
        "\nDownload Data\n( text files and pictures )\nfrom remote Hard Disk\n",
        "\nContact with other\nmobile phones\n\n",
        "\n\nPersonal Settings\n\n",
        "\n\nCreate Text Files.\nRename existing files.\n"
    };

        private Integer[] mImageIds = { R.drawable.world_connected,
                                        R.drawable.upload_data,
                                        R.drawable.download_data,
                                        R.drawable.phone_to_phone,
                                        R.drawable.settings,
                                        R.drawable.create_rename};

    public ImageAdapter(Context c)      { mContext = c; }

    public int getCount()              { return mImageIds.length; }

    public Object getItem(int position) { return position; }

    public long getItemId(int position)
    {
        info.setText(mImageDescrip[position]);
        select.setText(mImageNames[position]);
        return position;
    }

    public View getView(int position, View convertView, ViewGroup parent)
    {
        ImageView i = new ImageView(mContext);
        i.setImageResource(mImageIds[position]);
        i.setScaleType(ImageView.ScaleType.FIT_CENTER);
        i.setLayoutParams(new Gallery.LayoutParams(160, 160));
        return i;
    }

    public float getAlpha(boolean focused, int offset)
    {
        return Math.max(0, 1.0f - (0.2f * Math.abs(offset)));
    }
}

```

```

public float getScale(boolean focused, int offset)
{
    return Math.max(0, 1.0f - (0.2f * Math.abs(offset)));
}
}

private class galleryListener implements OnItemClickListener
{
    public void onItemClick(AdapterView parent,View v,int position,long id)
    {
        Toast.makeText(activity, "" + position,Toast.LENGTH_SHORT).show();
    }
}

private class selectButtonListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        if((select.getText()).equals("Function Execution"))
        {
            new FunctionExecution(activity);
        }
        else if((select.getText()).equals("Upload Data"))
        {
            new UploadData(activity);
        }
        else if((select.getText()).equals("Download Data"))
        {
            new DownloadData(activity);
        }
        else if((select.getText()).equals("Mobile communication"))
        {
            new MobileCommunication(activity);
        }
        else if((select.getText()).equals("Settings"))
        {
            new Settings(activity);
        }
        else if((select.getText()).equals("Create - Rename File"))
        {
            new CreateRenameFile(activity);
        }
    }
}
}
}
}

```


Upload – Download Data

➤ `upload_download_data.xml`

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="20px">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="26px"
        android:textAlign="center"
        android:text="Choose Category\n"/>

    <ImageButton android:id="@+id/datatext"
        android:src="@drawable/text"
        android:layout_gravity="center"
        android:layout_width="250px"
        android:layout_height="120px"/>

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="14px"
        android:textAlign="center"
        android:text="\n"/>

    <ImageButton android:id="@+id/dataimage"
        android:src="@drawable/image"
        android:layout_gravity="center"
        android:layout_width="250px"
        android:layout_height="120px"/>

</LinearLayout>
```

➤ `UploadData.java`

```
package google.android.package1;

import android.app.Activity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;

public class UploadData
{
    private Activity activity;
    private ImageButton text;
```

```

private ImageButton image;

public UploadData(Activity activ)
{
    activity = activ;
    activity.setContentViews(R.layout.upload_download_data);

    text = (ImageButton) activity.findViewById(R.id.datatext);
    image = (ImageButton) activity.findViewById(R.id.dataimage);

    ButtonsListener bl = new ButtonsListener();

    text.setOnClickListener(bl);
    image.setOnClickListener(bl);
}

private class ButtonsListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        if (v.equals(text))
        {
            new UploadDataText(activity);
        }
        else if (v.equals(image))
        {
            new UploadDataImage(activity);
        }
    }
}
}

```

➤ **DownloadData.java**

```

package google.android.packagel;

import android.app.Activity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;

```

```

public class DownloadData
{
    private Activity    activity;
    private ImageButton text;
    private ImageButton image;

    public DownloadData(Activity activ)
    {
        activity = activ;
        activity.setContentview(R.layout.upload_download_data);

        text = (ImageButton) activity.findViewById(R.id.datatext);
        image = (ImageButton) activity.findViewById(R.id.dataimage);

        ButtonsListener bl = new ButtonsListener();

        text.setOnClickListener(bl);
        image.setOnClickListener(bl);
    }

    private class ButtonsListener implements Button.OnClickListener
    {
        public void onClick(View v)
        {
            if (v.equals(text))
            {
                new DownloadDataText(activity);
            }
            else if (v.equals(image))
            {
                new DownloadDataImage(activity);
            }
        }
    }
}

```

Upload Data - Text

➤ upload_textfiles.xml

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="5px">

    <Spinner android:id="@+id/spinnertext"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_x="70px"
        android:layout_y="42px">

    </Spinner>

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:padding="5px">

        <Button android:id="@+id/showtext"
            android:text="Show"
            android:layout_width="80px"
            android:layout_height="45px" />

        <Button android:id="@+id/deletetext"
            android:text="Delete"
            android:layout_width="80px"
            android:layout_height="45px"/>

        <Button android:id="@+id/uploadtext"
            android:textStyle="bold"
            android:text="Upload"
            android:layout_width="80px"
            android:layout_height="45px"/>

        <Button android:id="@+id/backtext"
            android:text="B"
            android:layout_width="60px"
            android:layout_height="45px"/>

    </LinearLayout>

    <EditText android:id="@+id/areatext"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:lines="15"
        android:editable="false"
        android:scrollHorizontally="false"
        android:textSize="16px"/>

</LinearLayout>
```

➤ UploadDataText.java

```
package google.android.package1;

import google.android.package2.RemoteServer;
import java.io.File;
import android.app.Activity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;

public class UploadDataText
{
    private Activity activity;
    private Spinner spinner;
    private Button show;
    private Button delete;
    private Button back;
    private Button upload;
    private EditText areatext;

    public UploadDataText(Activity activ)
    {
        activity = activ;
        activity setContentView(R.layout.upload_textfiles);

        spinner = (Spinner) activity.findViewById (R.id.spinnertext);
        show = (Button) activity.findViewById (R.id.showtext);
        delete = (Button) activity.findViewById (R.id.deletetext);
        back = (Button) activity.findViewById (R.id.backtext);
        upload = (Button) activity.findViewById (R.id.uploadtext);
        areatext= (EditText)activity.findViewById (R.id.areatext);

        Function.updateSpinner(activity, spinner, "textfiles");
        ButtonsListener bl = new ButtonsListener();
        show.setOnClickListener(bl);
        delete.setOnClickListener(bl);
        back.setOnClickListener(bl);
        upload.setOnClickListener(new UploadListener());
    }
}
```

```

private class ButtonsListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        String    filenam = (String)spinner.getSelectedItem();
        boolean fileExists = false;
        if (filenam != null)
        {
            fileExists = true;
        }
        else
        {
            areatext.setText("No file selected.");
        }
        if ( v.equals(show))
        {
            if (fileExists)
            {
                areatext.setText( filenam+"\n"+AndroidFile.getText(activity, filenam));
            }
        }
        else if ( v.equals(delete))
        {
            if (fileExists)
            {
                long start    = System.currentTimeMillis();

                (new
File("data/data/google.android.package1/files/textfiles/"+filenam)).delete();

                long stop    = System.currentTimeMillis();
                long duration = stop - start;

                Function.updateSpinner(activity, spinner, "textfiles");

                areatext.setText("Delete time: "+duration+" milliseconds");
            }
        }
        else if ( v.equals(back))
        {
            new MainMenu(activity);
        }
    }
}
}
}

```

```

private class UploadListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        String filename = (String)spinner.getSelectedItem();

        if(filename != null)
        {
            try
            {
                long start = System.currentTimeMillis();

                String response = RemoteServer.uploadFile(activity, filename,true);

                long stop = System.currentTimeMillis();
                long duration = stop - start;

                if (response.equals("ok"))
                {
                    areatext.setText("File "+filename+
                    "\n\nuploaded successfully.\n\nUpload time: "+duration+" milliseconds.");
                }
                else if (response.equals("exist"))
                {
                    areatext.setText("File "+filename+" is already uploaded.");
                }
                else
                {
                    areatext.setText("Error: "+response);
                }

                if (GLOBAL.flaglocal)
                {
                    (new
                    File("data/data/google.android.package1/files/textfiles/"+filename)).delete();

                    stop = System.currentTimeMillis();
                    duration = stop - start;

                    Function.updateSpinner(activity, spinner, "textfiles");
                }
            }
        }
    }
}

```



```

        <Button android:id="@+id/deleteimage"
            android:text="Delete"
            android:layout_width="80px"
            android:layout_height="45px"/>

        <Button android:id="@+id/uploadimage"
            android:textStyle="bold"
            android:text="Upload"
            android:layout_width="80px"
            android:layout_height="45px"/>

        <Button android:id="@+id/backimage"
            android:text="B"
            android:layout_width="60px"
            android:layout_height="45px"/>

    </LinearLayout>

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="10px"
        android:textAlign="center"
        android:text="\n"/>

    <ImageView
        android:id="@+id/image"
        android:src="@drawable/icon"
        android:layout_gravity="center"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

    <TextView
        android:id="@+id/imagetransferinfo"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="12px"
        android:textAlign="center"
        android:text=""/>

</LinearLayout>

```

➤ UploadDataImage.java

```

package google.android.package1;
import google.android.package2.RemoteServer;
import java.io.File;
import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;

```

```

public class UploadDataImage
{
    private Activity activity;
    private Spinner spinner;
    private Button show;
    private Button delete;
    private Button back;
    private Button upload;
    private TextView info;
    private ImageView image;

    public UploadDataImage(Activity activ)
    {
        activity = activ;
        activity setContentView(R.layout.upload_imagfiles);

        spinner = (Spinner) activity.findViewById (R.id.spinnerimage);
        show = (Button) activity.findViewById (R.id.showimage);
        delete = (Button) activity.findViewById (R.id.deleteimage);
        back = (Button) activity.findViewById (R.id.backimage);
        upload = (Button) activity.findViewById (R.id.uploadimage);
        info = (TextView) activity.findViewById (R.id.imagetransferinfo);
        image = (ImageView) activity.findViewById (R.id.image);

        image.setVisibility(View.INVISIBLE);

        Function.updateSpinner(activity, spinner, "imagefiles");

        ButtonsListener bl = new ButtonsListener();
        show.setOnClickListener(bl);
        delete.setOnClickListener(bl);
        back.setOnClickListener(bl);
        upload.setOnClickListener(new UploadListener());
    }

    private class ButtonsListener implements Button.OnClickListener
    {
        public void onClick(View v)
        {
            info.setText("");
        }
    }
}

```

```

String filename = (String)spinner.getSelectedItem();
boolean fileExists = false;

if (filename != null)
{
    fileExists = true;
}
if ( v.equals(show) )
{
    if(fileExists)
    {
        try
        {
            AndroidFile.drawImage(activity, image, filename, 250);
        }
        catch(Exception e)
        {
        }
    }
    else
    {
        info.setText("\nNo file selected.");
    }
}
else if ( v.equals(delete) )
{
    if(fileExists)
    {
        long start = System.currentTimeMillis();

(new
File("data/data/google.android.package1/files/imagefiles/"+filename)).delete();

        long stop = System.currentTimeMillis();
        long duration = stop - start;

Function.updateSpinner(activity, spinner, "imagefiles");

AndroidFile.drawImage(activity, image, R.drawable.ok);

```

```

info.setText("\nDelete Time: "+duration+" milliseconds");

        }
        else
        {
            info.setText("\nNo file selected.");
        }
    }
    else if ( v.equals(back))
    {
        new MainMenu(activity);
    }
}

}

private class UploadListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        String filename = (String)spinner.getSelectedItem();
        if (filename != null)
        {
            try
            {
                long start = System.currentTimeMillis();

String response = RemoteServer.uploadFile(activity,filename,true);

                long stop = System.currentTimeMillis();
                long duration = stop - start;

                if (response.equals("ok"))
                {
                    AndroidFile.drawImage(activity,image,R.drawable.ok);

                    info.setText("\nUpload time: "+duration+" milliseconds");

                    if (GLOBAL.flaglocal)
                    {

```


Download Data - Text

➤ download_textfiles.xml

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="5px">

    <Spinner android:id="@+id/downspinnertext"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_x="70px"
        android:layout_y="42px">

    </Spinner>

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:padding="5px">

        <Button android:id="@+id/downloadtext"
            android:textStyle="bold"
            android:text="Download"
            android:layout_width="120px"
            android:layout_height="45px" />

        <Button android:id="@+id/downdeletetext"
            android:textStyle="bold"
            android:text="Delete"
            android:layout_width="120px"
            android:layout_height="45px"/>

        <Button android:id="@+id/downbacktext"
            android:text="B"
            android:layout_width="60px"
            android:layout_height="45px"/>

    </LinearLayout>

    <EditText android:id="@+id/downareatext"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:lines="15"
        android:editable="false"
        android:scrollHorizontally="false"
        android:textSize="16px"/>

</LinearLayout>
```

➤ **DownloadDataText.java**

```
package google.android.package1;
import google.android.package2.RemoteServer;
import java.io.File;
import android.app.Activity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;

public class DownloadDataText
{
    private Activity activity;
    private Spinner spinner;
    private Button download;
    private Button delete;
    private Button back;
    private EditText areatext;
    private String filesString;
    public DownloadDataText(Activity activ)
    {
        activity = activ;
        activity setContentView(R.layout.download_textfiles);
        spinner = (Spinner) activity.findViewById (R.id.downspinnertext);
        download = (Button) activity.findViewById (R.id.downloadtext);
        delete = (Button) activity.findViewById (R.id.downdeletetext);
        back = (Button) activity.findViewById (R.id.downbacktext);
        areatext = (EditText)activity.findViewById (R.id.downareatext);
        filesString = RemoteServer.getListTextFiles();
        Function.updateSpinner(activity, spinner, filesString);
        areatext.setText("Choose the text file that you want to download.");
        download.setOnClickListener(new DownloadFileListener());
        delete.setOnClickListener(new DeleteFileListener());
        back.setOnClickListener(new Button.OnClickListener()
        {
            public void onClick(View v)
            {
                new MainMenu(activity);
            }
        });
    }
}
```

```

private class DownloadFileListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        String filename = (String)spinner.getSelectedItem();

        if (filename == null)
        {
            areatext.setText("No file selected");
        }
        else if( new File("data/data/google.android.package1
                        /files/textfiles/"+filename).exists() )
        {
areatext.setText("\nFile "+filename+" already exists in Mobile phone");
        }
        else
        {
            long start = System.currentTimeMillis();
            String answer = RemoteServer.downloadFile(filename);
            long stop,duration;
            if (answer.equals("ok"))
            {
String s = (String)RemoteServer.downloadServerFile(activity,filename);

                RemoteServer.deleteServerFile(filename);

                stop = System.currentTimeMillis();
                duration = stop - start;

areatext.setText("Downloadtime: "+duration+" milliseconds\n"+filename+"\n"+s);

                if (GLOBAL.flagremote)
                {
String answer2 = RemoteServer.deleteFile(filename);

                    stop = System.currentTimeMillis();
                    duration = stop - start;

                    if(answer2.equals("ok"))
                    {
filesString = RemoteServer.getListTextFiles();

```



```

Function.updateSpinner(activity, spinner, filesString);
areatext.setText("Download+Delete time: "+duration+"
                milliseconds\n"+filename+"\n"+s);
                }
                Else
                {
areatext.setText("File "+filename+" downloaded succesfully, but an error ocured
while trying to delete it from the remote place.\n\nTotal time: "+duration+"
milliseconds");
                }
                }
                }
        }
    }
}

```

```

private class DeleteFileListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        String filename = (String)spinner.getSelectedItem();

        if (filename == null )
        {
            areatext.setText("No file selected");
        }
        else
        {
            long start = System.currentTimeMillis();

            String answer = RemoteServer.deleteFile(filename);

            long stop = System.currentTimeMillis();
            long duration = stop - start;

            if(answer.equals("ok"))
            {
                filesString = RemoteServer.getListTextFiles();
                Function.updateSpinner(activity, spinner, filesString);
                areatext.setText("File: "+filename+" was deleted.\n\n
                                Delete time: "+duration+" milliseconds");
            }
        }
    }
}

```

```

        }
        else
        {
areatext.setText("An error ocured while deleting file: "+filename+ " .");
        }
    }
}
}
}
}

```

Download Data image

➤ **download_imagefiles.xml**

```

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="5px">

    <Spinner android:id="@+id/downspinnerimage"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_x="70px"
        android:layout_y="42px">

    </Spinner>

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:padding="5px">

        <Button android:id="@+id/downloadimage"
            android:textStyle="bold"
            android:text="Download"
            android:layout_width="120px"
            android:layout_height="45px" />

        <Button android:id="@+id/downdeleteimage"
            android:textStyle="bold"
            android:text="Delete"
            android:layout_width="120px"
            android:layout_height="45px"/>

        <Button android:id="@+id/downbackimage"
            android:text="B"
            android:layout_width="60px"
            android:layout_height="45px"/>

    </LinearLayout>

```

```

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize ="10px"
    android:textAlign="center"
    android:text="\n"/>

<ImageView
    android:id="@+id/downimage"
    android:src="@drawable/icon"
    android:layout_gravity="center"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>

<TextView
    android:id="@+id/imagetransferinfo2"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize ="12px"
    android:textAlign="center"
    android:text=""/>

</LinearLayout>

```

➤ **DownloadDataImage.java**

```

package google.android.package1;

import java.io.File;
import google.android.package2.RemoteServer;
import android.app.Activity;
import android.graphics.Bitmap;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;

public class DownloadDataImage
{
    private Activity activity;
    private Spinner spinner;
    private Button download;
    private Button delete;
    private Button back;
    private TextView info;
    private ImageView image;
    private String filesString;

```

```

public DownloadDataImage(Activity activ)
{
    activity = activ;
    activity setContentView(R.layout.download_imagefiles);

    spinner = (Spinner) activity.findViewById (R.id.downspinnerimage);
    download= (Button) activity.findViewById (R.id.downloadimage);
    delete = (Button) activity.findViewById (R.id.downdeleteimage);
    back = (Button) activity.findViewById (R.id.downbackimage);
    info = (TextView) activity.findViewById (R.id.imagetransferinfo2);
    image = (ImageView) activity.findViewById (R.id.downimage);
    image.setVisibility(View.INVISIBLE);
    filesString = RemoteServer.getListImageFiles();
    Function.updateSpinner(activity, spinner, filesString);

    download.setOnClickListener(new DownloadFileListener());
    delete.setOnClickListener(new DeleteFileListener());

    back.setOnClickListener( new Button.OnClickListener()
        {
            public void onClick(View v)
            {
                new MainMenu(activity);
            }
        });
}

private class DownloadFileListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        String filename = (String)spinner.getSelectedItem();

        if(filename == null)
        {
            info.setText("No file selected");
        }
        else if ((newFile("data/data/google.android.package1
            /files/imagefiles/"+filename)).exists())
        {
            info.setText("\nFile already exists in Mobile phone");
        }
    }
}

```

```

else
{
    long start = System.currentTimeMillis();

    String answer = RemoteServer.downloadFile(filename);

    long stop,duration;

    if (answer.equals("ok"))
    {
Bitmap bm = (Bitmap)RemoteServer.downloadServerFile(activity,filename);

        RemoteServer.deleteServerFile(filename);

        stop      = System.currentTimeMillis();
        duration = stop - start;

        AndroidFile.drawImage(activity, image, bm);

info.setText("\nDownload time:  "+duration+"  milliseconds");

        if (GLOBAL.flagremote)
        {

String answer2  = RemoteServer.deleteFile(filename);

            if(answer2.equals("ok"))
            {

                stop = System.currentTimeMillis();

                duration = stop - start;

                filesString = RemoteServer.getListImageFiles();
Function.updateSpinner(activity, spinner, filesString);

                AndroidFile.drawImage(activity, image, R.drawable.ok);

info.setText("\nDownload+Delete time:  "+duration+"  milliseconds");

            }
        }
    }
else

```

```

        {
            AndroidFile.drawImage(activity, image, R.drawable.error);

            info.setText("\nError: "+answer2);
        }
    }
}
else
{
    AndroidFile.drawImage(activity, image, R.drawable.error);
    info.setText("\nError: "+answer);
}
}
}

private class DeleteFileListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        String filename = (String)spinner.getSelectedItem();

        if(filename!=null)
        {
            long start = System.currentTimeMillis();

            String answer = RemoteServer.deleteFile(filename);

            long stop = System.currentTimeMillis();

            long duration = stop - start;

            if(answer.equals("ok"))
            {
                filesString = RemoteServer.getListImageFiles();
                Function.updateSpinner(activity, spinner, filesString);
            }
        }
    }
}

```

```
        AndroidFile.drawImage(activity, image, R.drawable.ok);

        info.setText("\nDelete time: "+duration+" milliseconds");
    }
    else
    {
        AndroidFile.drawImage(activity, image, R.drawable.error);
        info.setText("\nError: "+answer);
    }

}

else
{
    info.setText("No file selected");
}

}

}
```

Start function execution

➤ **functionexecution.xml**

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10px">

    <Spinner android:id="@+id/spinnerfunction"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_x="70px"
        android:layout_y="42px">

</Spinner>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:padding="12px"
    android:textSize="18px"
    android:textAlign="center"
    android:text="Input-Output Description"/>

<EditText
    android:id="@+id/areainfo"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:lines="2"
    android:editable="false"
    android:scrollHorizontally="false"
    android:textSize="16px"/>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize="12px"
    android:textAlign="center"
    android:text=" "/>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:padding="5px">

<TextView
    android:layout_width="90px"
    android:layout_height="30px"
    android:textSize="18px"
    android:textAlign="center"
    android:text="INPUTS:"/>
```



```

<TextView    android:id="@+id/inputnumber"
            android:layout_width="50px"
            android:layout_height="30px"
            android:textSize="18px"
            android:textAlign="center"
            android:text="0"/>

</LinearLayout>

<LinearLayout android:id="@+id/llayout"
            xmlns:android="http://schemas.android.com/apk/res/android"
            android:orientation="vertical"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:padding="0px">

</LinearLayout>

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="6px"
        android:textAlign="center"
        android:text=" "/>

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:padding="5px">

        <TextView
            android:layout_width="100px"
            android:layout_height="30px"
            android:textSize="18px"
            android:textAlign="center"
            android:text=" OUTPUTS: "/>

        <TextView android:id="@+id/outputnumber"
            android:layout_width="50px"
            android:layout_height="30px"
            android:textSize="18px"
            android:textAlign="center"
            android:text="0"/>

    </LinearLayout>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:padding="5px">

    <Button    android:id="@+id/execute"
            android:textSize="18px"
            android:textStyle="bold"
            android:text="Execute"
            android:layout_width="90px"
            android:layout_height="40px"/>

```

```

        <Button        android:id="@+id/functionmonitor"
                    android:textStyle="bold"
                    android:textSize = "18px"
                    android:text="Monitor >"
                    android:layout_width="140px"
                    android:layout_height="40px"/>

        <Button        android:id="@+id/functionback"
                    android:text="Back"
                    android:layout_width="80px"
                    android:layout_height="40px"/>

    </LinearLayout>
</LinearLayout>

```

➤ **FunctionExecution.java**

```

package google.android.packagel;

import google.android.package2.RemoteServer;
import google.android.package2.FunctionInformation;
import android.app.Activity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.AdapterView.OnItemClickListener;

public class FunctionExecution
{
    private Activity  activity;
    private Spinner  spinner;
    private EditText  areainfo;
    private Button    execute;
    private Button    back;
    private Button    monitor;
    private TextView  inputs, outputs;
    private FunctionInformation[]  funobj;
    private String[]  arrayfunction;

```

```

private String    remotetexts,remoteimages;
private LinearLayout llayout;
private int index;
private selectFileLocationListener selectFileLocationlistener;
private String    currentfiletype;
public Spinner[]  filespinner  = new Spinner[2];
public ImageView[] fileimage    = new ImageView[2];
public int        ii;

public FunctionExecution(Activity activ)
{
    activity = activ;

    activity setContentView(R.layout.functionexecution);

    index = 1000;
    selectFileLocationlistener = new selectFileLocationListener();

    spinner = (Spinner) activity.findViewById (R.id.spinnerfunction);
    areainfo= (EditText)activity.findViewById (R.id.areainfo);
    execute = (Button)  activity.findViewById (R.id.execute);
    back    = (Button)  activity.findViewById (R.id.functionback);
    monitor = (Button)  activity.findViewById (R.id.functionmonitor);
    inputs  = (TextView)activity.findViewById (R.id.inputnumber);
    outputs = (TextView)activity.findViewById (R.id.outputnumber);

    llayout = (LinearLayout)activity.findViewById(R.id.llayout);

    while( (GLOBAL.funobj == null) || (GLOBAL.functionsarray==null) )
    {
        try
        {
            Thread.sleep(100);
        }
        catch(InterruptedException ie)
        {
        }
    }

    funobj = GLOBAL.funobj;
}

```

```

arrayfunction = GLOBAL.functionsarray;
ArrayAdapter<String> adapter = new ArrayAdapter<String>(activity,
    android.R.layout.simple_spinner_item, arrayfunction);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinner.setAdapter(adapter);

if (GLOBAL.textsFlag)
{
    remotetexts = RemoteServer.getListTextFiles();
}

if (GLOBAL.imagesFlag)
{
    remoteimages = RemoteServer.getListImageFiles();
}
spinner.setOnItemClickListener(new OnItemSelectedListener()
{
    public void onItemClick(AdapterView parent, View v,
        int position, long id)
    {
        String funName = (String)spinner.getSelectedItem();

        for (int i=0;i<arrayfunction.length;i++)
        {
            if (arrayfunction[i].equals(funName))
            {
                areainfo.setText(funobj[i].getDescription());

                int inputnumber = funobj[i].getInputsNumber();
                inputs.setText (""+inputnumber);

                currentfiletype = funobj[i].getType();

                index=1000;
                llayout.removeAllViews();

                for(ii=0; ii<inputnumber; ii++)
                {
                    addInputRow();
                }
            }
        }
    }
}

```

```

filespinner[0] = (Spinner) activity.findViewById(1100+0);
fileimage[0]   = (ImageView)activity.findViewById(1050+0);

filespinner[0].setOnItemSelectedListener(
    new OnItemSelectedListener()
    {
        @Override
        public void onItemSelected(AdapterView arg0,
            View arg1, int arg2,long arg3)
        {
            String filename = (String)filespinner[0].getSelectedItem();
            try
            {
                If (new
                File("data/data/google.android.package1/files/imagefiles/"+filename).exists()){
                    fileimage[0].setVisibility(View.VISIBLE);
                    AndroidFile.drawImage(activity, fileimage[0], filename, 50);
                }else{
                    fileimage[0].setVisibility(View.INVISIBLE); }
            }
            catch(Exception e)
            {
            }
        }
    });

@Override
public void onNothingSelected(AdapterView arg0)
{
}

if(ii==1)
{
    filespinner[1] = (Spinner) activity.findViewById(1100+1);
    fileimage[1]   = (ImageView)activity.findViewById(1050+1);
    filespinner[1].setOnItemSelectedListener(
        new OnItemSelectedListener()
        {
            @Override
            public void onItemSelected(AdapterView arg0,
                View arg1, int arg2,long arg3)

```

```

        {
            String filename = (String)filespicker[1].getSelectedItem();
            try
            {
                If (new
                File("data/data/google.android.package1/files/imagefiles/"+filename).exists()){
                    fileimage[1].setVisibility(View.VISIBLE);
                    AndroidFile.drawImage(activity, fileimage[1], filename, 50);
                }else{
                    fileimage[1].setVisibility(View.INVISIBLE); }
                    }
                catch(Exception e)
                {
                    }
                }
            }

            @Override
            public void onNothingSelected(AdapterView arg0)
            {
                }
            }

        });
    }

    outputs.setText(""+funobj[i].getOutpoutNumber());
}
}

public void onNothingSelected(AdapterView arg0)
{
}
});

```

```

execute.setOnClickListener(new executeButtonsListener());
back.setOnClickListener(

        new Button.OnClickListener()
        {
            public void onClick(View v)
            {
                new MainMenu(activity);
            }
        }
);

monitor.setOnClickListener(

        new Button.OnClickListener()
        {
            public void onClick(View v)
            {
                new FunctionMonitor(activity);
            }
        }
);

}

private void addInputRow()
{
    LinearLayout linearlayout = new LinearLayout(activity);
    linearlayout.setOrientation(0);
    linearlayout.setId(index);

    ImageView im = new ImageView(activity);
    im.setId(index+50);

    Spinner sp = new Spinner(activity);
    sp.setId(index+100);

    CheckBox b = new CheckBox(activity);
    b.setId(index+200);
    b.setSelected(false);
}

```

```

linearlayout.addView(im, new LinearLayout.LayoutParams(60,50));
linearlayout.addView(sp, new LinearLayout.LayoutParams(200,50));
linearlayout.addView(b, new LinearLayout.LayoutParams(40,40));

llyout.addView(linearLayout,new LinearLayout.LayoutParams(350,50));
if(currentfiletype.equals("txt"))
{
    Function.updateSpinner(activity, sp, "textfiles");
}
else
{
    Function.updateSpinner(activity, sp, "imagefiles");
}

b.setOnClickListener(selectFileLocationlistener);

index++;
}

```

```

private class selectFileLocationListener implements Button.OnClickListener
{
    public void onClick(View v)
    {

CheckBox tempcheckbox= (CheckBox) activity.findViewById (v.getId());
Spinner tempspinner = (Spinner) activity.findViewById (v.getId()-100);
ImageView tempimage = (ImageView)activity.findViewById (v.getId()-150);

        if ( !tempcheckbox.isChecked() )
        {
            if(currentfiletype.equals("txt"))
            {
                Function.updateSpinner(activity, tempspinner, "textfiles");
            }
            else
            {
                Function.updateSpinner(activity, tempspinner, "imagefiles");
            }

            tempimage.setVisibility(View.VISIBLE);
        }
    }
}

```



```

else
{
    if(currentfiletype.equals("txt"))
    {
Function.updateSpinner(activity, tempspinner, remotetexts);
    }
    else
    {
Function.updateSpinner(activity, tempspinner, remoteimages);
    }
    tempimage.setVisibility(View.INVISIBLE);
}
}
}

```

```

private class executeButtonsListener implements Button.OnClickListener

```

```

{
    public void onClick(View v)
    {
        StringBuffer sb = new StringBuffer();
        String functionName = (String) spinner.getSelectedItem();
        sb.append("<function-execution>");
        int totalinputs = 0;
        for (int i=0;i<arrayfunction.length;i++)
        {
            if (arrayfunction[i].equals(functionName))
            {
sb.append("<function><url>"+funobj[i].getUrl()+"</url></function>");
                totalinputs = funobj[i].getInputsNumber();
            }
        }

        try
        {
            for(int i=0;i<totalinputs;i++)
            {
Spinner sp = (Spinner) activity.findViewById(1100+i);
CheckBox ch = (CheckBox) activity.findViewById(1200+i);
                sb.append("<input>");
            }
        }
    }
}

```

```

        String filename = (String)sp.getSelectedItem();
        sb.append("<name>"+filename+"</name>");
        if (ch.isChecked())
        {
            sb.append("<from>gria</from>");
        }
        else
        {
            sb.append("<from>mobile</from>");
        }

        RemoteServer.uploadFile(activity, filename, false);
    }
    sb.append("</input>");
}

sb.append("</function-execution>");

String answer = RemoteServer.executeFunction(sb.toString());

        areainfo.setText(answer);
    }
    catch(Exception ex)
    {
        areainfo.setText("Error type 2 in function: "+functionName+" start!");
    }
}
}
}

```

Monitor Function Execution

➤ **functionmonitor.xml**

```

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10px">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"

```

```

        android:textSize ="18px"
        android:textAlign="center"
        android:padding="5px"
        android:text="Conversation ID"/>

<Spinner      android:id="@+id/spinnerfunctmonitor"
              android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:layout_x="70px"
              android:layout_y="42px">

</Spinner>

<TextView     android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:textSize ="18px"
              android:textAlign="center"
              android:padding="5px"
              android:text="Functions Status"/>

<EditText     android:id="@+id/functionstatus"
              android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:lines="5"
              android:editable="false"
              android:scrollHorizontally="false"
              android:textSize="16px"/>

<TextView     android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:textSize ="10px"
              android:textAlign="center"
              android:text=" "/>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:orientation="horizontal"
              android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:padding="0px">

    <Button     android:id="@+id/refresh"
              android:textSize ="18px"
              android:text="Refresh"
              android:layout_width="100px"
              android:layout_height="40px"/>

    <Button     android:id="@+id/stopexecution"
              android:textSize ="18px"
              android:text="Stop"
              android:textStyle="bold"
              android:layout_width="70px"
              android:layout_height="40px"/>

    <Button     android:id="@+id/delfailedorstopped"
              android:textSize ="18px"
              android:text="Clear History"
              android:layout_width="140px"
              android:layout_height="40px"/>

</LinearLayout>

```

```

        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textSize ="6px"
            android:textAlign="center"
            android:text=" "/>

        <EditText
            android:id="@+id/monitorarea"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:lines="2"
            android:editable="false"
            android:scrollHorizontally="false"
            android:textSize="16px"/>

        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textSize ="8px"
            android:textAlign="center"
            android:text=" "/>

        <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
            android:orientation="horizontal"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:padding="0px">

            <Button
                android:id="@+id/monitorexecution"
                android:textSize ="18px"
                android:text="back"
                android:layout_width="100px"
                android:layout_height="40px"/>

            <Button
                android:id="@+id/monitortomenu"
                android:textSize ="18px"
                android:text="Main menu"
                android:layout_width="200px"
                android:layout_height="40px"/>

        </LinearLayout>
    </LinearLayout>

```

➤ **FunctionMonitor.java**

```

package google.android.package1;

import google.android.package2.RemoteServer;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.StringTokenizer;
import android.app.Activity;
import android.view.View;
import android.widget.AdapterView;

```

```

import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.AdapterView.OnItemClickListener;

public class FunctionMonitor
{
    private Activity activity;
    private Spinner spinner;
    private EditText areafundesc;
    private EditText areainfo;
    private Button refresh;
    private Button stop;
    private Button delete;
    private Button rettoexecution;
    private Button back;

    private String[] convidarray;
    private String[] convidinfo;

    public FunctionMonitor(Activity activ)
    {
        activity = activ;
        activity setContentView(R.layout.functionmonitor);

        spinner = (Spinner) activity.findViewById (R.id.spinnerfunctmonitor);
        areafundesc = (EditText) activity.findViewById (R.id.functionstatus);
        areainfo = (EditText) activity.findViewById (R.id.monitorarea);
        refresh = (Button) activity.findViewById (R.id.refresh);
        stop = (Button) activity.findViewById (R.id.stopexecution);

        delete = (Button) activity.findViewById (R.id.delfailedorstopped);
        rettoexecution = (Button) activity.findViewById (R.id.monitorexecution);
        back = (Button) activity.findViewById (R.id.monitortomenu);

        functionsstatus();
        spinner.setOnItemClickListener(new OnItemClickListener()
        {
            public void onItemClick(AdapterView parent, View v,int
                position, long id)

```

```

        {
            String convid = (String) spinner.getSelectedItem();
            for (int i=0;i<convidarray.length;i++)
            {
                if (convidarray[i].equals(convid))
                {
                    areafundesc.setText(convidinfo[i]);
                }
            }
        }

        public void onNothingSelected(AdapterView arg0)
        {
        }
    }
);
refresh.setOnClickListener(new refreshButtonListener());
stop.setOnClickListener(new stopButtonListener());
delete.setOnClickListener(new clearHistoryButtonListener());
back.setOnClickListener(

        new Button.OnClickListener()
        {
            public void onClick(View v)
            {
                new MainMenu(activity);
            }
        }
);
rettoexecution.setOnClickListener(

        new Button.OnClickListener()
        {
            public void onClick(View v)
            {
                new FunctionExecution(activity);
            }
        }
);
}

```

```

private void functionsstatus()
{
    String result = RemoteServer.getRunningFunctionsInfo();
    if(result !=null)
    {
        ArrayList<String> arrayconvid = new ArrayList<String>();
        ArrayList<String> arrayconvinfo= new ArrayList<String>();
        StringTokenizer tokens = new StringTokenizer(result);
        while(tokens.hasMoreTokens())
        {
            arrayconvid.add(tokens.nextToken("#"));
            arrayconvinfo.add(tokens.nextToken("#"));
        }
        convidarray = (String[]) arrayconvid.toArray(new String[arrayconvid.size()]);
        convidinfo = (String[]) arrayconvinfo.toArray(new String[arrayconvinfo.size()]);
    }
    Arrays.sort(convidarray);
    ArrayAdapter<String> adapter = new ArrayAdapter<String>(activity,
        android.R.layout.simple_spinner_item, convidarray);

    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spinner.setAdapter(adapter);
}
private class refreshButtonListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        functionsstatus();
        areainfo.setText("Functions status - refreshed.");
    }
}
private class stopButtonListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        String answer =
        RemoteServer.stopFunctionExecution((String) spinner.getSelectedItem());
        areainfo.setText(answer);
    }
}
}

```

```

private class clearHistoryButtonListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        String answer = RemoteServer.clearFunctionExecutionHistory();
        areainfo.setText(answer);
        areafundesc.setText("");
        functionsstatus();
    }
}
}

```

Mobile Communication

➤ mobilecommunication.xml

```

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10px">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="26px"
        android:textAlign="center"
        android:text="Mobile Communication"/>

    <TextView
        android:id="@+id/ourvisibility"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="16px"
        android:textAlign="center"
        android:text="' You are invisible '\n"/>

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="24px"
        android:text=" File:"/>

    <EditText
        android:id="@+id/searchfile"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:lines="1"
        android:scrollHorizontally="false"
        android:textSize="20px"/>

    <TextView
        android:layout_width="fill_parent"

```



```

        android:layout_height="wrap_content"
        android:textSize = "6px"
        android:text=" " />

<Button        android:id="@+id/search"
        android:textSize = "18px"
        android:textStyle="bold"
        android:text="Search"
        android:layout_width="150px"
        android:layout_height="50px"/>

<TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize = "10px"
        android:text=" " />

<TextView        android:id="@+id/searchinfo"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize = "16px"
        android:text=" Search results"/>

<TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize = "8px"
        android:text=" " />

<Spinner        android:id="@+id/availablefiles"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">

</Spinner>

<TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize = "6px"
        android:text=" " />

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:padding="0px">

<Button        android:id="@+id/directdownload"
        android:textSize = "18px"
        android:textStyle="bold"
        android:text="Download"
        android:layout_width="150px"
        android:layout_height="50px"/>

<TextView
        android:layout_width="60px"
        android:layout_height="wrap_content"

```

```

        android:textSize ="16px"
        android:text=" "/>

    <Button        android:id="@+id/mobileback"
        android:textSize ="18px"
        android:text="Back"
        android:layout_width="80px"
        android:layout_height="50px"/>

</LinearLayout>

</LinearLayout>

```

➤ **MobileCommunication.java**

```

package google.android.package1;

import java.io.File;
import google.android.package2.RemoteServer;
import google.android.package2.AndroidTCPClient;
import android.app.Activity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;

public class MobileCommunication
{
    private Activity  activity;
    private Button    search;
    private Button    download;
    private Button    back;
    private TextView  info,visib;
    private EditText  area;
    private Spinner   spinner;

    public MobileCommunication(Activity activ)
    {
        activity = activ;
        activity setContentView(R.layout.mobilecommunication);

        search    = (Button)  activity.findViewById(R.id.search);
        download  = (Button)  activity.findViewById(R.id.directdownload);

```

```

back      = (Button)    activity.findViewById(R.id.mobileback);
visib     = (TextView) activity.findViewById(R.id.ourvisibility);
info      = (TextView) activity.findViewById(R.id.searchinfo);
area      = (EditText) activity.findViewById(R.id.searchfile);
spinner   = (Spinner)  activity.findViewById(R.id.availablefiles);

visib.setText(GLOBAL.mobilevisibility);
search.setOnClickListener(new searchListener());
download.setOnClickListener(new downloadListener());
back.setOnClickListener(

        new Button.OnClickListener()
        {
            public void onClick(View v)
            {
                new MainMenu(activity);
            }
        }
);
}

private class searchListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        String filename = ((area.getText()).toString()).trim();

        if( (filename.trim()).equals(""))
        {
            info.setText(" Please give the filename.");
        }
        else if
        ( (!filename.endsWith(".txt")) && (!filename.endsWith(".png")) )
        {
            info.setText(" Filename is not correct.");
        }
        else
        {
            String answer = RemoteServer.search(filename);

            if(answer.matches("[0-9]+[.][0-9]+[.][0-9]+[.][0-9]+"))

```

```

        {
            info.setText(" File found in mobile phone:");

            String [] array = { answer };
            ArrayAdapter<String> adapter = new ArrayAdapter<String>(activity,
                android.R.layout.simple_spinner_item, array);
            adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
            spinner.setAdapter(adapter);
        }
        else
        {
            info.setText(" "+answer);
        }
    }
}

```

```

private class downloadListener implements Button.OnClickListener

```

```

{
    public void onClick(View v)
    {
        String ip = (String) spinner.getSelectedItem();

        if( ip == null )
        {
            info.setText(" Warning - no mobile phone selected.");
        }
        else
        {
            String filename = ((area.getText()).toString()).trim();

            AndroidTCPClient.askQuestion(activity,ip, "getfile", filename);

            if (
                (
                    new
                    File("data/data/google.android.package1/files/"+AndroidFile.getFileType(filename)+f
                    ilename) ).exists() )
            {
                info.setText(" File (new) "+filename+" downloaded.");
            }
            else

```

```

        {
            info.setText(" Error: file (new)"+filename+" is not exists.");
        }
    }
}
}
}
}

```

Create – Rename File

➤ **createrenamefile.xml**

```

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10px">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="26px"
        android:textAlign="center"
        android:text="Create Text File"/>

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:padding="0px">

        <Button
            android:id="@+id/createfile"
            android:textSize="18px"
            android:textStyle="bold"
            android:text="Create"
            android:layout_width="100px"
            android:layout_height="50px"/>

        <EditText
            android:id="@+id/filename"
            android:layout_width="200px"
            android:layout_height="50px"
            android:lines="1"
            android:scrollHorizontally="true"
            android:textSize="20px"/>

    </LinearLayout>

    <EditText
        android:id="@+id/filetext"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:lines="6"
        android:scrollHorizontally="false"
        android:textSize="16px"/>

```

```

<TextView
    android:layout_width="fill_parent"
    android:layout_height="5px"
    android:textSize="5px"
    android:text="" />

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize="26px"
    android:textAlign="center"
    android:text="Rename file"/>

<Spinner
    android:id="@+id/existingfiles"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
</Spinner>

<EditText
    android:id="@+id/newfilename"
    android:layout_width="295px"
    android:layout_height="50px"
    android:lines="1"
    android:scrollHorizontally="true"
    android:textSize="20px"/>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="2px"
    android:textSize="2px"
    android:text="" />

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:padding="0px">

    <Button
        android:id="@+id/renamefile"
        android:textSize="18px"
        android:textStyle="bold"
        android:text="Rename"
        android:layout_width="100px"
        android:layout_height="50px"/>

    <TextView
        android:layout_width="105px"
        android:layout_height="wrap_content"
        android:textSize="16px"
        android:text="" />

    <Button
        android:id="@+id/createrenamesback"
        android:textSize="18px"
        android:text="Back"
        android:layout_width="90px"
        android:layout_height="50px"/>

</LinearLayout>

</LinearLayout>

```

➤ CreateRenameFile.java

```
package google.android.packagel;

import java.io.File;
import java.util.ArrayList;
import java.util.Arrays;
import android.app.Activity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;

public class CreateRenameFile
{
    private Activity activity;
    private Button create;
    private Button rename;
    private EditText filename;
    private EditText filecontent;
    private Spinner spinner;
    private EditText newname;
    private Button back;
    private String[] array;

    public CreateRenameFile(Activity activ)
    {
        activity = activ;
        activity setContentView(R.layout.createrenamefile);

        create = (Button) activity.findViewById (R.id.createfile);
        rename = (Button) activity.findViewById (R.id.renamefile);
        back = (Button) activity.findViewById (R.id.createrenamesback);
        filename = (EditText) activity.findViewById (R.id.filename);
        filecontent = (EditText) activity.findViewById (R.id.filetext);
        newname = (EditText) activity.findViewById (R.id.newfilename);
        spinner = (Spinner) activity.findViewById (R.id.existingfiles);

        Function.updateSpinner(activity, spinner, "textsandimages");
        create.setOnClickListener(new createListener());
    }
}
```

```

rename.setOnClickListener(new renameListener());
back.setOnClickListener( new Button.OnClickListener()
    {

        public void onClick(View v)
        {
            new MainMenu(activity);
        }
    });
}

private class createListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        String fname = (filename.getText()).toString();
        int index = fname.indexOf('.');

        if(index!=-1){ fname = fname.substring(0,index)+".txt"; }
        else { fname = fname+".txt"; }
        String fcontent = (filecontent.getText()).toString();
        try
        {
            AndroidFile.create(activity, fname, fcontent);
            new MainMenu(activity);
        }
        catch(Exception e)
        {
            filecontent.setText("Error ... "+e.toString());
        }
    }
}

private class renameListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        String fname = (String)spinner.getSelectedItem();
        String newfname = (newname.getText()).toString();
        int index = newfname.indexOf('.');
        if(index!=-1)

```



```

        {
            newfname = newfname.substring(0,index)+"";
        }
        if (fname.endsWith(".png"))
        {
            (new
File("data/data/google.android.package1/files/imagefiles/"+fname)).renameT(
new File("data/data/google.android.package1/files/imagefiles/"+newfname+".png")
);

            new MainMenu(activity);
        }
        else if (fname.endsWith(".txt"))
        {
            (new
File("data/data/google.android.package1/files/textfiles/"+fname)).renameTo(
new File("data/data/google.android.package1/files/textfiles/"+newfname+".txt")
);

            new MainMenu(activity);
        }
    }
}
}
}

```

Settings

➤ settings.xml

```

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10px">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="26px"
        android:textAlign="center"
        android:text="Settings\n"/>

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="22px"
        android:textAlign="center"
        android:text="M o b i l e   p h o n e\n"/>

```

```

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize ="26px"
    android:textAlign="center"
    android:text="127.0.0.1\n"/>

<CheckBox    android:id="@+id/settingsvisible"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=" Visible from other mobile phones" >
</CheckBox>

<CheckBox    android:id="@+id/settingsdeletelocalfiles"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=" Delete file from mobile phone,\n after
                uploading." >
</CheckBox>

<CheckBox    android:id="@+id/settingsdeleteremotefiles"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=" Delete file from remote hard disk,\n after
                downloading." >
</CheckBox>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize ="26px"
    android:text=" "/>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:padding="5px">

    <Button    android:id="@+id/settingsstore"
        android:textSize ="18px"
        android:text="Store change"
        android:layout_width="200px"
        android:layout_height="50px"/>

    <Button    android:id="@+id/settingsback"
        android:textSize ="18px"
        android:text="Back"
        android:layout_width="90px"
        android:layout_height="50px"/>

</LinearLayout>

</LinearLayout>

```

➤ Settings.java

```
package google.android.package1;

import google.android.package2.RemoteServer;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.util.StringTokenizer;
import android.app.Activity;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;

public class Settings
{
    private Activity activity;
    private Button store;
    private Button back;
    private CheckBox checkvisible;
    private CheckBox checklocaldelete;
    private CheckBox checkremotedelete;
    public Settings(Activity activ)
    {
        activity = activ;
        activity setContentView(R.layout.settings);
        checkvisible = (CheckBox) activity.findViewById(R.id.settingsvisible);
        checklocaldelete = (CheckBox) activity.findViewById(R.id.settingsdeletelocalfiles);
        checkremotedelete = (CheckBox) activity.findViewById(R.id.settingsdeleteremotefiles);
        store = (Button) activity.findViewById(R.id.settingsstore);
        back = (Button) activity.findViewById(R.id.settingsback);
        loadPersonalSetting();
        store.setOnClickListener(new storeListener());
        back.setOnClickListener(new Button.OnClickListener()
        {
            public void onClick(View v)
            {
                new MainMenu(activity);
            }
        });
    }
}
```

```

private void loadPersonalSetting()
{
    try
    {
        FileInputStream fIn = activity.openFileInput("mysettings.txt");
        InputStreamReader isr = new InputStreamReader(fIn);
        char[] inputBuffer = new char[200];
        isr.read(inputBuffer);
        String ssettings = new String(inputBuffer);

        StringTokenizer tokens = new StringTokenizer(ssettings);
        if ((tokens.nextToken()).equals("true"))
        {
            checkvisible.setChecked(true);
        }
        if ((tokens.nextToken()).equals("true"))
        {
            checklocaldelete.setChecked(true);
        }
        if ((tokens.nextToken()).equals("true"))
        {
            checkremotedelete.setChecked(true);
        }
    }
    catch(Exception e)
    {
    }
}

private class storeListener implements Button.OnClickListener
{
    public void onClick(View v)
    {
        try
        {

        FileOutputStream fOut =
        activity.openFileOutput( "mysettings.txt",activity.MODE_WORLD_WRITEABLE);
        OutputStreamWriter osw = new OutputStreamWriter(fOut);

            String text = "";

```

```
if (checkvisible.isChecked())
{
    text += "true ";
    RemoteServer.sendIP("127.0.0.1");
    GLOBAL.mobilevisibility = "' You are Visible '";
}
else
{
    text += "false ";
    RemoteServer.removeIP("127.0.0.1");
    GLOBAL.mobilevisibility = "' You are Invisible '";
}

if (checklocaldelete.isChecked())
{
    text += "true ";
    GLOBAL.flaglocal = true;
}
else
{
    text += "false ";
    GLOBAL.flaglocal = false;
}

if (checkremotedelete.isChecked())
{
    text += "true ";
    GLOBAL.flagremote = true;
}
else
{
    text += "false ";
    GLOBAL.flagremote = false;
}

osw.write(text);
osw.flush();
osw.close();
```

```
        new MainMenu(activity);
    }
    catch (Exception e)
    {
    }
}
}
```

Total images



Other important classes

➤ **AndroidFile.java**

```
package google.android.packagel;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStreamWriter;
import java.nio.ByteBuffer;
import java.nio.channels.FileChannel;
import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Matrix;
import android.graphics.Bitmap.CompressFormat;
import android.view.View;
import android.widget.ImageView;

public class AndroidFile
{
    private static String text    = "textfiles\\";
    private static String image = "imagefiles\\";

    public static void create(Activity activity, String filename,
String text) throws IOException
    {
        String path = getFileType(filename);

        FileOutputStream fOut =
            activity.openFileOutput(path+filename,activity.MODE_WORLD_WRITEABLE);
        OutputStreamWriter osw = new OutputStreamWriter(fOut);
        osw.write(text);
        osw.flush();
        osw.close();
    }
}
```



```

public static String getText(Activity activity, String filename)
{
    String path = getFileType(filename);
    try
    {
        FileInputStream fIn = activity.openFileInput(path+filename);
        InputStreamReader isr = new InputStreamReader(fIn);
        char[] inputBuffer = new char[1000];
        isr.read(inputBuffer);
        String readString = new String(inputBuffer);
        return readString.trim();
    }
    catch(IOException e)
    {
        return "Error: "+e.toString();
    }
}

public static void createFile(Activity activity, String filename,
byte[] bytearray) throws IOException
{
    String path = getFileType(filename);
    FileOutputStream fos =
    activity.openFileOutput(path+filename,activity.MODE_WORLD_WRITEABLE);
    fos.write(bytearray);
    fos.flush();
    fos.close();
}

public static byte[] getBytes(Activity activity, String filename)
throws Exception
{
    String path = getFileType(filename);
    // Get file bytes
    FileInputStream fis = activity.openFileInput(path+filename);
    FileChannel fc = fis.getChannel();
    byte[]data = new byte[(int) (fc.size())];
    ByteBuffer bb = ByteBuffer.wrap(data);
    fc.read(bb);
    return data;
}

```

```

public static String getFileType(String filename)
{
    if (filename.endsWith(".png"))
    {
        return image;
    }
    else if (filename.endsWith(".txt"))
    {
        return text;
    }
    else
    {
        return "";
    }
}

public static void storeImage(Activity activity, String imagename,
Bitmap bm)
{
    try
    {
        FileOutputStream fos = activity.openFileOutput(image+imagename,
activity.MODE_WORLD_READABLE);
        bm.compress(CompressFormat.PNG, 100, fos);
        fos.flush();
        fos.close();
    }
    catch (Exception e)
    {
    }
}

public static Bitmap resize(Bitmap bm, int newWidth, int newHeight)
{
    int        width,height;
    float      scaleWidth,scaleHeight;
    Matrix     matrix;
    width      = bm.width();
    height     = bm.height();
    scaleWidth = ((float) newWidth) / width ;
    scaleHeight = ((float) newHeight)/ height;
}

```

```

        matrix      = new Matrix();
        matrix.postScale(scaleWidth, scaleHeight);
        return Bitmap.createBitmap(bm, 0, 0,width, height, matrix, true);
    }

    public static void drawImage(Activity activity, ImageView image,
    String filename, int size) throws FileNotFoundException
    {
        FileInputStream fIn =
        activity.openFileInput("imagefiles\\"+filename);
        Bitmap bm = BitmapFactory.decodeStream(fIn);
        bm = resize(bm, size, size);
        image.setVisibility(View.VISIBLE);
        image.setImageBitmap(bm);
    }

    public static void drawImage(Activity activity, ImageView image,
    int imageid)
    {
        Bitmap bm =
        BitmapFactory.decodeResource(activity.getResources(), imageid);
        bm = resize(bm, 250, 250);
        image.setVisibility(View.VISIBLE);
        image.setImageBitmap(bm);
    }

    public static void drawImage(Activity activity, ImageView image,
    Bitmap bm)
    {
        bm = AndroidFile.resize(bm, 250, 250);
        image.setVisibility(View.VISIBLE);
        image.setImageBitmap(bm);
    }
}

```

➤ **Function.java**

```

package google.android.package1;

import java.io.File;
import java.util.ArrayList;

```

```

import java.util.Arrays;
import java.util.StringTokenizer;
import android.app.Activity;
import android.widget.ArrayAdapter;
import android.widget.Spinner;

public class Function
{
    public static String getText(String xmlstring, String childname)
    {
        int indexstart = xmlstring.indexOf("<"+childname+">");
        int indexend = xmlstring.indexOf("</"+childname+">");
        if((indexstart>=0)&&(indexend>=0))
        {
            int offset = childname.length()+2;
            String response =
            xmlstring.substring(indexstart+offset, indexend);
            return response;
        }
        else
        {
            return null;
        }
    }

    public static String[] makeArray( String s )
    {
        ArrayList<String> array = new ArrayList<String>();
        StringTokenizer tokens = new StringTokenizer(s);
        while (tokens.hasMoreTokens())
        {
            array.add((String) tokens.nextToken());
        }
        return (String[]) array.toArray(new String[array.size()]);
    }

    public static void updateSpinner(Activity activity, Spinner spinner,
    String files)
    {
        String[] array;
        if(files.equals("textfiles") || files.equals("imagefiles"))

```

```

{
    array =
        (new
File("data/data/google.android.package1/files/"+files)).list();
}
else if (files.equals("textsandimages"))
{
    array = textandimageFiles();
}
else
{
    array = Function.makeArray(files);
}
Arrays.sort(array);
ArrayAdapter<String> adapter = new ArrayAdapter<String>(activity,
android.R.layout.simple_spinner_item, array);
adapter.setDropDownViewResource(android.R.layout.
simple_spinner_dropdown_item);
spinner.setAdapter(adapter);
}

```

```

public static String[] textandimageFiles()

```

```

{
    String [] texts =
        (new File("data/data/google.android.package1/files/textfiles")).list();
    String [] images =
        (new
File("data/data/google.android.package1/files/imagefiles")).list();
    ArrayList<String> alist = new ArrayList<String>();
    int i;
    if (texts!=null)
    {
        for (i=0;i<texts.length;i++)
        {
            alist.add(texts[i]);
        }
    }
    if (images!=null)
    {
        for (i=0;i<images.length;i++)
        {

```

```

        alist.add(images[i]);
    }
}
return (String[]) alist.toArray(new String[alist.size()]);
}
}

```

➤ GLOBAL.java

```

package google.android.package1;
import google.android.package2.FunctionInformation;
import android.app.Activity;

public class GLOBAL
{
    public static Activity        activity;
    public static boolean        flaglocal;
    public static boolean        flagremote;
    public static String         mobilevisibility = "' You are Invisible '";
    public static FunctionInformation[] funobj;
    public static String[]       functionsarray;
    public static boolean        textsFlag;
    public static boolean        imagesFlag;
}

```

➤ Thread_FunctionsInformations.java

```

package google.android.package1;
import java.util.ArrayList;
import google.android.package2.FunctionInformation;
import google.android.package2.RemoteServer;
import android.app.Activity;

public class Thread_FunctionsInformations extends Thread
{
    private Activity activity;
    public Thread_FunctionsInformations(Activity activity)
    {
        this.activity = activity;
    }
}

```

```

public void run()
{
    GLOBAL.funobj = RemoteServer.downloadFunctionInfo(activity);

    GLOBAL.functionsarray = makeArrayNames(GLOBAL.funobj);
}

private String[] makeArrayNames(FunctionInformation[] funobj)
{
    ArrayList<String> array = new ArrayList<String>();
    for(int i=0;i<funobj.length;i++)
    {
        array.add(funobj[i].getName());

        if ( (funobj[i].getType()).equals("txt") )
        {
            GLOBAL.textsFlag = true;
        }
        else if ( (funobj[i].getType()).equals("png") )
        {
            GLOBAL.imagesFlag = true;
        }
    }
    return (String[]) array.toArray(new String[array.size()]);
}
}

```

➤ **Thread_PersonalSettings.java**

```

package google.android.package1;

import google.android.package2.RemoteServer;
import java.io.FileInputStream;
import java.io.InputStreamReader;
import java.util.StringTokenizer;
import android.app.Activity;

public class Thread_PersonalSettings extends Thread
{
    private Activity activity;

```

```

public Thread_PersonalSettings(Activity activity)
{
    this.activity = activity;
}

public void run()
{
    // Load Personal Settings
    try
    {
        FileInputStream fIn =
            activity.openFileInput("mysettings.txt");
        InputStreamReader isr = new InputStreamReader(fIn);
        char[] inputBuffer = new char[200];
        isr.read(inputBuffer);
        String ssettings = new String(inputBuffer);

        StringTokenizer tokens = new StringTokenizer(ssettings);

        if ((tokens.nextToken()).equals("true"))
        {
            RemoteServer.sendIP("127.0.0.1");
            GLOBAL.mobilevisibility = "' You are Visible '";
        }
        else
        {
            RemoteServer.removeIP("127.0.0.1");
            GLOBAL.mobilevisibility = "' You are Invisible '";
        }

        if ((tokens.nextToken()).equals("true"))
        {
            GLOBAL.flaglocal = true;
        }
        else
        {
            GLOBAL.flaglocal = false;
        }
    }
}

```



```

        if ((tokens.nextToken()).equals("true"))
        {
            GLOBAL.flagremote = true;
        }
        else
        {
            GLOBAL.flagremote = false;
        }
    }
    catch(Exception e)
    {
    }
}
}

```

Android TCP Server

➤ **StartServer.java**

```

package google.android.packagel;

import android.content.Context;
import android.content.Intent;
import android.content.IntentReceiver;

public class StartServer extends IntentReceiver
{
    static final String ACTION = "android.intent.action.BOOT_COMPLETED";

    @Override
    public void onReceiveIntent(Context context, Intent intent)
    {
        if (intent.getAction().equals(ACTION))
        {
            context.startService(new Intent(context, Server.class), null);
        }
    }
}

```

➤ **Server.java**

```
package google.android.package1;

import google.android.package2.AndroidTCPServer;
import android.app.Service;
import android.content.Intent;
import android.os.Bundle;
import android.os.IBinder;

public class Server extends Service
{
    @Override
    public IBinder onBind(Intent intent)
    {
        return null;
    }

    @Override
    protected void onStart(int startId, Bundle arguments)
    {
        super.onStart( startId, arguments );

        // Start Android Server
        Thread sThread = new Thread(new AndroidTCPServer(GLOBAL.activity));
        sThread.start();
    }

    @Override
    protected void onDestroy()
    {
        super.onDestroy();
    }
}
```

➤ **AndroidTCPServer.java**

```
package google.android.package2;

import google.android.package1.AndroidFile;
import java.net.ServerSocket;
import java.net.Socket;
```

```

import java.nio.ByteBuffer;
import java.nio.channels.FileChannel;
import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import android.app.Activity;

public class AndroidTCPServer implements Runnable
{
    public static final String IP = "127.0.0.1";
    public static final int PORT = 4444;
    private Activity activity;
    private Socket client;

    private String[] function =
    {"Random_Number", "Get_a_String", "Get_an_Image", "function4"};
    private String[] descript = {"Return a random number.",
    "Return a string.", "Return an image.", "Description of function 4."};

    public AndroidTCPServer(Activity activ)
    {
        activity = activ;
    }

    public void run()
    {
        try
        {
            ServerSocket serverSocket = new ServerSocket(PORT);
            while (true)
            {
                client = serverSocket.accept();
            }
            try
            {
                // Get request
                InputStream is = client.getInputStream();
            }
        }
    }
}

```

```

BufferedInputStream bis = new BufferedInputStream(is);
DataInputStream dis = new DataInputStream(bis);

// Annalize the request
String request = new String(dis.readUTF());
int index = request.indexOf(':');
String query = request.substring(0, index);
String file = request.substring(index+1,request.length());

// Send response
if (query.equals("upload"))
{
    upload(file);
}
else if (query.equals("search"))
{
    searchForFile(file);
}
else if (query.equals("getfile"))
{
    sendFile(file);
}
else if (query.equals("supportedfunctions"))
{
    sendFunctionsNames();
}
else if (query.equals("description"))
{
    try
    {
        int desindex = Integer.parseInt(file);
        sendDescription(desindex);
    }
    catch(Exception e)
    {

    }
}
else if (query.equals("execute"))
{
    try

```

```

        {
            int funindex = Integer.parseInt(file);
            executeFunction(funindex);
        }
        catch(Exception e)
        {
        }
    }
}
catch(Exception e)
{
}
finally
{
    client.close();
}
}

}
catch (Exception e) { }

}

private void upload(String filename) throws IOException
{
    FileInputStream fis =
    activity.openFileInput("imagefiles\\"+filename);
    FileChannel fc = fis.getChannel();
    byte[]data = new byte[(int) (fc.size())];
    ByteBuffer bb = ByteBuffer.wrap(data);
    fc.read(bb);

    // Send bytes
    OutputStream os = client.getOutputStream ();
    BufferedOutputStream bos = new BufferedOutputStream (os);
    DataOutputStream dos = new DataOutputStream (bos);
    dos.write(data);
    dos.flush();
    dos.close();
}

```

```

private void searchForFile(String filename) throws IOException
{
    String answer;
    File f1 =
    new File("data/data/google.android.package1/files/textfiles/" +filename);
    File f2 =
    new File("data/data/google.android.package1/files/imagefiles/"+filename);

    if ((f1.exists()) || (f2.exists()))
    {
        answer = "ok";
    }
    else
    {
        answer = "not exists";
    }

    OutputStream          os    = client.getOutputStream ();
    BufferedOutputStream  bos    = new BufferedOutputStream (os);
    DataOutputStream      dos    = new DataOutputStream (bos);

    dos.writeUTF(answer);
    dos.flush();
    dos.close();
}

private void sendFile(String filename) throws IOException
{
    if (filename.endsWith(".png"))
    {
        // Get file bytes
        FileInputStream fis  =
        activity.openFileInput("imagefiles\\"+filename);
        FileChannel fc = fis.getChannel();
        byte[] data    = new byte[(int) (fc.size())];
        ByteBuffer bb = ByteBuffer.wrap(data);
        fc.read(bb);

        // Send bytes
        OutputStream os = client.getOutputStream ();
        BufferedOutputStream bos  = new BufferedOutputStream (os);
        DataOutputStream      dos  = new DataOutputStream (bos);
    }
}

```

```

        dos.write(data);
        dos.flush();
        dos.close();
    }
    else if (filename.endsWith(".txt"))
    {
        String text = AndroidFile.getText(activity, filename);
        OutputStream os = client.getOutputStream ();
        BufferedOutputStream bos = new BufferedOutputStream (os);
        DataOutputStream dos = new DataOutputStream (bos);

        dos.writeUTF(text);
        dos.flush();
        dos.close();
    }
}

```

private void sendFunctionsNames() throws IOException

```

{
    StringBuffer sb = new StringBuffer();
    for (int i=0;i<function.length;i++)
    {
        sb.append(function[i]+" ");
    }
    OutputStream os = client.getOutputStream ();
    BufferedOutputStream bos = new BufferedOutputStream (os);
    DataOutputStream dos = new DataOutputStream (bos);
    dos.writeUTF(sb.toString());
    dos.flush();
    dos.close();
}

```

private void sendDescription(int index) throws IOException

```

{
    OutputStream os = client.getOutputStream ();
    BufferedOutputStream bos = new BufferedOutputStream (os);
    DataOutputStream dos = new DataOutputStream (bos);
    dos.writeUTF(descript[index]);
    dos.flush();
    dos.close();
}

```

```

private void executeFunction(int index) throws IOException
{
    OutputStream          os    = client.getOutputStream ();
    BufferedOutputStream  bos   = new BufferedOutputStream (os);
    DataOutputStream      dos   = new DataOutputStream (bos);

    String text = "";
    if (index == 0)      // function : Random_Number
    {
        text = ""+15;
    }

    else if (index == 1) // function : Get_a_String
    {
        text = "tim chros";
    }
    else if (index == 2) // function : Get_an_Image
    {
        text = "image ... xaxa";
    }
    else if (index == 3) // function : function4
    {

    }

    dos.writeUTF(text);
    dos.flush();
    dos.close();
}
}

```

Android TCP Client

➤ **AndroidTCPClient.java**

```

package google.android.package2;

import google.android.package1.AndroidFile;
import java.io.BufferedReader;
import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;

```



```

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.InetAddress;
import java.net.Socket;
import android.app.Activity;

public class AndroidTCPClient
{
    public static void askQuestion(Activity activity,String ip,
                                   String action, String filename)
    {
        Socket socket = null ;
        try
        {
            InetAddress serverAddr = InetAddress.getByName(ip);
            socket = new Socket (serverAddr, 4444);

            OutputStream os          = socket.getOutputStream ();
            BufferedOutputStream bos= new BufferedOutputStream (os);
            DataOutputStream dos     = new DataOutputStream (bos);

            dos.writeUTF(action+": "+filename);
            dos.flush();

            InputStream is = socket.getInputStream();
            BufferedInputStream bis = new BufferedInputStream(is);
            DataInputStream dis = new DataInputStream(bis);

            if (filename.endsWith(".png"))
            {
                byte[] data3 = new byte[100000];
                byte[] data2 = new byte[100000];
                int readData=0;
                int readData2;
            }
        }
    }
}

```

```

while ((readData2 = dis.read(data2))>0)
{
    for(int i=readData;i<readData+readData2;i++)
    {
        data3[i]=data2[i-readData];
    }
    readData+=readData2;
}

dos.close();
dis.close();

FileOutputStream fos = activity.openFileOutput(
    "imagefiles\\"+"new_"+filename,activity.MODE_WORLD_WRITEABLE)

fos.write(data3);
fos.flush();
fos.close();
}
else if (filename.endsWith(".txt"))
{
    String text = dis.readUTF();
    dos.close();
    dis.close();
    AndroidFile.create(activity, "new_"+filename, text);
}
}
catch(Exception e)
{
}
finally {
    try
    {
        if (socket!=null) socket.close();
    }
    catch(Exception e)
    {
}
}
}
}

```

Call Services

➤ RemoteServer.java

```
package google.android.package2;

import google.android.package1.AndroidFile;
import google.android.package1.Function;
import java.io.BufferedInputStream;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLConnection;
import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import org.apache.commons.codec.binary.*;

public class RemoteServer
{
    private static final String server          = "192.168.1.3";
    private static final String downloadUrl    =
        "http://" + server + ":8080/mydata/gria_to_android/"; //194.219.51.97
    private static final String serviceUrl     =
        "http://" + server + ":8080/axis2/services/TimService/function/";
    private static final String functionInfoUrl =
        "http://" + server + ":8080/mydata/supported_functions.xml";
    private static final String mobileIp      = "127.0.0.1";

    /** -----
     *                               SEND - REMOVE IP
     * ----- */

    public static void sendIP(String ip) throws IOException
    {
        try
        {
            String functionname = "sendIp";
```

```

        String parameters    = "ip="+ip;
        executeServerFunction(functionname,parameters);
    }
    catch(Exception e)
    {
    }
}

public static void removeIP(String ip) throws IOException
{
    try
    {
        String functionname = "removeIp";
        String parameters    = "ip="+ip;
        executeServerFunction(functionname,parameters);
    }
    catch(Exception e)
    {
    }
}

/** -----
 *                SEARCH FILE IN OTHER MOBILE PHONE
 * ----- */

public static String search(String filename)
{
    try
    {
        String functionname = "search";
        String parameters    = "filename="+filename;
        String response =
        executeServerFunction(functionname,parameters);
        return response;
    }
    catch(Exception e)
    {
        return "Local error.";
    }
}

```

```

/** -----
 *
 *          UPLOAD FILE
 * ----- **/

public static String uploadFile(Activity activity, String
filename,boolean flag)
{
    try
    {
        byte[] bytearray    =
        Base64.encodeBase64(AndroidFile.getBytes(activity,filename));
        String filecontent  = new String(bytearray);
        String flagvalue    = (flag) ? "true" : "false";
        String functionname = "uploadFile";
        String parameters   =
        "ip="+mobileIp+"&filename="+filename+"&filecontent="+
        filecontent+"&flag="+flagvalue;

        String response =
        executeServerFunction(functionname,parameters);
        return response;
    }
    catch(Exception e)
    {
        return "local error";
    }
}

```

```

/** -----
 *
 *          DOWNLOAD FILE
 * ----- **/

```

```

public static String  getListTextFiles()
{
    try
    {
        String functionname = "getFileNames";
        String parameters   = "ip="+mobileIp+"&filetype=text";
    }
}

```

```

        String response =
            executeServerFunction(functionname,parameters);
        return response;
    }
    catch(Exception e)
    {
        return " ";
    }
}

public static String    getListImageFiles()
{
    try
    {
        String functionname = "getFileNames";
        String parameters    = "ip="+mobileIp+"&filetype=image";
        String response =
            executeServerFunction(functionname,parameters);
        return response;
    }
    catch(Exception e)
    {
        return " ";
    }
}

public static String    downloadFile(String filename)
{
    try
    {
        String functionname = "downloadFile";
        String parameters    = "ip="+mobileIp+"&filename="+filename;
        String response =
            executeServerFunction(functionname,parameters);
        return response;
    }
    catch(Exception e)
    {
        return "local error";
    }
}
}

```

```

public static Object downloadServerFile(Activity activity,String filename)
{
    try
    {
        URL myUrl = new URL(downloadUrl+filename);
        HttpURLConnection conn=(HttpURLConnection)myUrl.openConnection();
        conn.setDoInput(true);
        conn.connect();
        InputStream is = conn.getInputStream();
        BufferedInputStream bis = new BufferedInputStream(is);

        if (filename.endsWith(".png"))
        {
            Bitmap bm = BitmapFactory.decodeStream(bis);

            bis.close();
            is.close();

            AndroidFile.storeImage(activity,filename, bm);

            return bm;
        }
        else if (filename.endsWith(".txt"))
        {
            StringBuffer sb = new StringBuffer();

            DataInputStream dis = new DataInputStream(bis);

            String line;
            while( (line = dis.readLine())!=null )
            {
                sb.append(line);
            }

            dis.close();

            AndroidFile.create(activity, filename, sb.toString());

            return sb.toString();
        }
    }
}

```

```

        else
        {
            return null;
        }
    }
    catch(Exception e)
    {
        return null;
    }
}

public static String deleteServerFile(String filename)
{
    try
    {
        String functionname = "deleteFileFromServer";
        String parameters = "filename="+filename;
        String response =
            executeServerFunction(functionname,parameters);
        return response;
    }
    catch(Exception e)
    {
        return "local error";
    }
}

/** -----
 *
 * DELETED FILE
 * ----- */

public static String deleteFile(String filename)
{
    try
    {
        String functionname = "deleteGriaFile";
        String parameters = "ip="+mobileIp+"&filename="+filename;
        String response =
            executeServerFunction(functionname,parameters);
        return response;
    }
}

```



```

        catch(Exception e)
        {
            return "local error";
        }
    }

/** -----
 *           FUNCTION EXECUTION : START AND MONITOR
 * ----- **/

// START

public static FunctionInformation[] downloadFunctionInfo(Activity activity)
{
    try{
        URL myUrl = new URL(functionInfoUrl);
        HttpURLConnection conn=(HttpURLConnection)myUrl.openConnection();
        conn.setDoInput(true);
        conn.connect();

        StringBuffer sb = new StringBuffer();

        InputStream is = conn.getInputStream();
        DataInputStream dis = new DataInputStream(is);

        String line;
        while( (line = dis.readLine())!=null )
        {
            sb.append(line);
        }

        dis.close();

        String functioninfo = sb.toString();

        MyStringTokenizer tokens = new MyStringTokenizer(functioninfo,"<function>");

        FunctionInformation[] funobj = new FunctionInformation[tokens.getNumber()];

        int i=0;

```

```

while(tokens.hasMoreTokens())
{
    String tim = tokens.nextToken();

    String name = Function.getText(tim,"name");
    String url = Function.getText(tim,"url");
    String description = Function.getText(tim,"description");
    String type = Function.getText(tim,"type");
    int inputs = Integer.parseInt(Function.getText(tim,"inputs"));
    int outputs = Integer.parseInt(Function.getText(tim,"outputs"));

    funobj[i++] = new
    FunctionInformation(name,url,description,type,inputs,outputs);
}

return funobj;
}

catch(Exception e)
{
    FunctionInformation[] funobj = new FunctionInformation[1];
    funobj[0] = new FunctionInformation("name","url",e.toString(),"type",1,1);
    return funobj;
}
}

public static String executeFunction(String xmlstring)
{
    try
    {
        String functionname = "startFunctionExecution";
        String parameters =
        "ip="+mobileIp+"&functioninfo="+xmlstring;
        String response =
        executeServerFunction(functionname,parameters);
        return response;
    }
    catch(Exception e){
        return "local error";
    }
}
}

```

```
// MONITOR
```

```
public static String getRunningFunctionsInfo()
```

```
{  
    try  
    {  
        String functionname = "getRunningFunctionsInfo";  
        String parameters   = "ip="+mobileIp;  
        String response =  
            executeServerFunction(functionname,parameters);  
        return response;  
    }  
    catch(Exception e){  
        return "";  
    }  
}
```

```
public static String stopFunctionExecution(String convid)
```

```
{  
    try  
    {  
        String functionname = "stopFunctionExecution";  
        String parameters   = "convId="+convid;  
        String response =  
            executeServerFunction(functionname,parameters);  
        return response;  
    }  
  
    catch(Exception e)  
    {  
        return "local error";  
    }  
}
```

```
public static String clearFunctionExecutionHistory()
```

```
{  
    try  
    {  
        String functionname = "clearFunctionExecutionHistory";  
        String parameters   = "ip="+mobileIp;
```

```

        String response =
            executeServerFunction(functionname,parameters);
        return response;
    }
    catch(Exception e)
    {
        return "local error";
    }
}

/*****/
private static String executeServerFunction(String functionname,
String parameters ) throws IOException
{
    // URL Android-Server
    URL servlet = new URL(serviceUrl+functionname);
    // Send Data
    URLConnection conn = servlet.openConnection();
    conn.setDoOutput(true);
    conn.setDoInput(true);
    DataOutputStream dos = new DataOutputStream(conn.getOutputStream());
    dos.writeBytes(parameters);
    dos.close();
    // Receive Data
    DataInputStream dis = new DataInputStream(conn.getInputStream());
    String response="";
    String nextline;
    while ((nextline = dis.readLine()) != null)
    {
        response+=nextline;
    }
    dis.close();
    response = response.substring(response.indexOf("<ns:return>")+11,
response.indexOf("</ns:return>"));

    return response;
}

/*****/
}

```

➤ **FunctionInformation.java**

```
package google.android.package2;
```

```
public class FunctionInformation  
{
```

```
    private String name;  
    private String url;  
    private String description;  
    private String type;  
    private int    inputs;  
    private int    outputs;
```

```
public FunctionInformation(String name, String url, String description,  
String type, int inputs, int outputs)
```

```
{  
    this.name          = name;  
    this.url           = url;  
    this.description   = description;  
    this.type          = type;  
    this.inputs        = inputs;  
    this.outputs       = outputs;  
}
```

```
public String getName()
```

```
{  
    return name;  
}
```

```
public String getUrl()
```

```
{  
    return url;  
}
```

```
public String getDescription()
```

```
{  
    return description;  
}
```

```
public String getType()
```

```
{  
    return type;  
}
```

```
public int getInputsNumber()
```

```

    {
        return inputs;
    }
    public int getOutpoutNumber()
    {
        return outputs;
    }
}

```

➤ **MyStringTokenizer.java**

```

package google.android.package2;

import java.util.ArrayList;

public class MyStringTokenizer
{
    private String      s;
    private char[]      array;
    private char[]      mystring;
    private int         index;
    private ArrayList<String>  arrayindex;
    private String[]    myindex;
    private int         total;

    public MyStringTokenizer(String s1, String s2)
    {
        s = s1;
        array = s1.toCharArray();
        mystring = s2.toCharArray();
        arrayindex = new ArrayList<String>();

        total = 0;
        for(int i=0;i<array.length;i++)
        {
            boolean flag;
            if(i+mystring.length<=array.length)
            {
                flag = true;

```

```

    }
    else
    {
        flag = false;
    }
    for (int j=i; (j<i+mystring.length)&&(flag);j++)
    {
        if (array[j] != mystring[j-i])
        {
            //System.out.println(j);
            flag = false;
            break;
        }
    }
    if (flag)
    {
        total++;
        arrayindex.add((String) ""+i);
    }
}

if (total!=0)
{
    myindex = (String[]) arrayindex.toArray(
        new String[arrayindex.size()]);
    index = 0;
}
}

public int getNumber()
{
    return total;
}

public boolean hasMoreTokens()
{
    if (index < total)
    {
        return true;
    }
}

```

```

        else
        {
            return false;
        }
    }

    public String nextToken()
    {
        int begin = Integer.parseInt(myindex[index]);
        int end;
        if(index+1<total)
        {
            end = Integer.parseInt(myindex[index+1]);
        }
        else
        {
            end = s.length();
        }
        index++;
        return s.substring(begin, end);
    }
}

```


10.2 Κώδικας Web Services

Services

➤ **function.java**

```
package packagel;

import java.io.BufferedOutputStream;
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.util.ArrayList;
import java.util.StringTokenizer;
import org.apache.commons.codec.binary.Base64;;

public class function
{
    private Gria                gia;
    private ArrayList<String>    arrayIp;
    private String[]            functionName;
    private String[]            functionUrl;

    public function()
    {
        try
        {
            gia = new Gria(
                "C:/apache                tomcat/apache-tomcat-
                6.0.13/webapps/mydata/android_to_gria/",
                "C:/apache                tomcat/apache-tomcat-
                6.0.13/webapps/mydata/gria_to_android/");

            if (GLOBAL.arrayip == null)
            {
                arrayIp = new ArrayList<String>();
                GLOBAL.arrayip    = arrayIp;
            }
        }
        else
    }
}
```

```

{
    arrayIp = GLOBAL.arrayip;
}

if ( !GLOBAL.flag )
{
    try
    {
        FileReader fr = new FileReader("C:/apache tomcat/
apache-tomcat-6.0.13/webapps/mydata/supported_functions.xml");

        BufferedReader br = new BufferedReader(fr);
        StringBuffer sb = new StringBuffer();

        String line;
        while( (line = br.readLine()) != null )
        {
            sb.append(line);
        }

        br.close();
        fr.close();

        int i = 0;
        MyStringTokenizer tokens =
        new MyStringTokenizer(sb.toString(), "<function>");
        functionName = new String[tokens.getNumber()];
        functionUrl = new String[tokens.getNumber()];
        while(tokens.hasMoreTokens())
        {
            String tim = tokens.nextToken();

            String name= XML.getText(tim, "name");
            String url = XML.getText(tim, "url");
            functionName[i] = name;
            functionUrl[i] = url;
            i++;
        }
        GLOBAL.functionName = functionName;
        GLOBAL.functionUrl = functionUrl;
    }
}

```

```

        catch(Exception e)
        {
            System.out.println("Error in reading functions info.");
        }

        GLOBAL.flag = true;
    }
    else
    {
        functionName      = GLOBAL.functionName;
        functionUrl        = GLOBAL.functionUrl;
    }

}
catch(Exception e)
{

}
}

/** -----
 *                               SEND - REMOVE IP
 * -----**/

public String sendIp(String ip)
{
    if ( !arrayIp.contains(ip) )
    {
        arrayIp.add(ip);
        return "ok";
    }
    else
    {
        return "ipexists";
    }
}

public String removeIp(String ip)
{
    arrayIp.remove(ip);
    return "ok";
}

```

```

/** -----
 *          SEARCH FOR FILE IN OTHER MOBILE PHONE
 * -----**/

public String search(String filename)
{
    try
    {
        String response = Mobile.searchForFile(filename);
        return response;
    }
    catch(Exception e)
    {
        return "Error in server.";
    }
}

/** -----
 *          UPLOAD DATA
 * -----**/

public String uploadFile(String ip,String filename, String filecontent,
boolean flag )
{
    try
    {
        FileOutputStream fos = new FileOutputStream("C:/apache tomcat/
apache-tomcat-6.0.13/webapps/mydata/android_to_gria/"+filename);

        BufferedOutputStream bos = new BufferedOutputStream(fos);
        DataOutputStream dos = new DataOutputStream(bos);

        byte[] arraybyte = Base64.decodeBase64(filecontent.getBytes());

        dos.write(arraybyte);

        bos.flush();
        bos.close();
        dos.close();
        fos.close();
    }
}

```

```

String response = "ok";

if ( flag )
{
    // upload file with gria
    response = gria.uploadFile(ip,filename);

    // delete file from server
    (new File("C:/apache tomcat/apache-tomcat 6.0.13/webapps/
        mydata/android_to_gria/"+filename)).delete();
}

return response;
}
catch(Exception e)
{
    System.out.println("Error: "+e.toString());
    return "error";
}
}

/** -----
 *                DOWNLOAD DATA
 * -----**/

public String getFileNames(String ip,String filetype)
{
    try
    {
        String response = "";

        if (filetype.equals("image"))
        {
            response = gria.getImageList(ip);
        }
        else if (filetype.equals("text"))
        {
            response = gria.getTextList(ip);
        }
        return response;
    }
}

```

```

        catch(Exception e)
        {
            return "error";
        }
    }

    public String downloadFile(String ip,String filename)
    {
        try
        {
            String response = gria.downloadFile(ip,filename);

            return response;
        }
        catch(Exception e)
        {
            return "error";
        }
    }

    public String deleteFileFromServer(String filename)
    {
        (new File("C:/apache tomcat/apache-tomcat-6.0.13/webapps/
                mydata/gria_to_android/"+filename)).delete();
        return "ok";
    }
}

/** -----
 *                DELETE DATA
 * -----**/

    public String deleteGriaFile(String ip,String filename)
    {
        try
        {
            String response = gria.deleteFile(ip,filename);
            return response;
        }
        catch(Exception e)
        {
            return "error";
        }
    }
}

```

```

/** -----
 *
 * -----**/

// START

public String startFunctionExecution(String ip,String functioninfo)
{
    String url = XML.getText(functioninfo,"url");

    String funname =
    (String) url.subSequence(url.lastIndexOf("/")+1, url.length());

    try
    {
        // upload data that we want
        fileInput[] inputs ;
        ArrayList<fileInput> arraylist = new ArrayList<fileInput>();

        fileInput[] outputs = new fileInput[1];
        outputs[0] = new fileInput("output_"+funname+"_"+
                                GLOBAL.index+".png", "");
        GLOBAL.index++;

        MyStringTokenizer tokens = new
        MyStringTokenizer(functioninfo,"<input>");
        while(tokens.hasMoreTokens())
        {
            String tim = tokens.nextToken();
            String filename = (XML.getText(tim,"name"));
            String from      = (XML.getText(tim,"from"));

            fileInput fi = new fileInput(filename,from);
            arraylist.add(fi);

            if (from.equals("gria"))
            {
                if (gria.fileExist(ip, filename))
                {
                    // ok ...
                }
            }
        }
    }
}

```

```

        else
        {
            throw new Exception();
        }
    }
    else
    {
        // files already uploaded
    }
}

inputs = (fileInput[]) arraylist.toArray(
    new fileInput[arraylist.size()]);

String response = gria.startFunctionExecution(ip,url,inputs,
    outputs);

return response;
}
catch(Exception e)
{
    System.out.println("error");

    return "Error in server - We can't start function execution";
}
}

// MONITOR
public String getRunningFunctionsInfo(String ip)
{
    try
    {
        String response = gria.getRunningFunctionInfo(ip);
        return response;
    }
    catch(Exception e)
    {
        return "error";
    }
}
}

```



```

public String stopFunctionExecution(String convId)
{
    try
    {
        String response = gria.stopFunctionExecution(convId);
        return response;
    }
    catch(Exception e)
    {
        return "error";
    }
}

public String clearFunctionExecutionHistory(String ip)
{
    return gria.clearFunctionExecutionHistory(ip);
}

```

XML Files

➤ XmlFile.java

```

package package1;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import com.sun.org.apache.xml.internal.serialize.OutputFormat;
import com.sun.org.apache.xml.internal.serialize.XMLSerializer;

public class XmlFile
{
    protected String          FileName;

```

```

protected File                file;
protected DocumentBuilder    db;

public XmlFile(String filename) throws
ParserConfigurationException,FileNotFoundException, IOException
{
    FileName    = new String(filename+".xml");
    file        = new File(FileName);

    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
    db = dbf.newDocumentBuilder();

    // Create xml file if not exists
    if (!file.exists())
    {
        create();
    }
}

private void create() throws FileNotFoundException, IOException
{
    Document doc = db.newDocument();
    Element rootElement = doc.createElement("data");
    doc.appendChild(rootElement);

    // Γραφουμε το xml αρχείο
    FileOutputStream fos = new FileOutputStream(FileName);
    OutputFormat of = new OutputFormat("XML","ISO-8859-1",true);
    of.setIndent(1);
    of.setIndenting(true);
    XMLSerializer serializer = new XMLSerializer(fos,of);
    serializer.asDOMSerializer();
    serializer.serialize( doc.getDocumentElement() );
    fos.close();
}

public String toString()
{
    StringBuffer sb=new StringBuffer();
    try
    {

```

```

        FileReader fr          = new FileReader(file);
        BufferedReader br = new BufferedReader(fr);

        String line;
        int recCount = 0;
        while( (line = br.readLine()) != null )
        {
            sb.append(recCount + ": " + line+"\n");
        }

        br.close();
        fr.close();
    }
    catch(Exception e)
    {
        sb = new StringBuffer("Error");
    }

    return sb.toString();
}
}

```

➤ **XmlFile1.java**

```

package packagel;

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

import org.xml.sax.SAXException;
import com.sun.org.apache.xml.internal.serialize.OutputFormat;
import com.sun.org.apache.xml.internal.serialize.XMLSerializer;

```

```

public class XmlFile1 extends XmlFile
{
    public XmlFile1(String filename)
throws ParserConfigurationException,FileNotFoundException, IOException
    {
        super(filename);
    }

    public String getFileNames(String phoneid)
throws IOException, SAXException, ParserConfigurationException
    {
        StringBuffer sb = new StringBuffer();

        String phoneId=null,fileName=null;

        Document doc = db.parse(file);
        doc.getDocumentElement().normalize();

        NodeList nodeLst = doc.getElementsByTagName("file");

        for (int s = 0; s < nodeLst.getLength(); s++)
        {
            Node fstNode = nodeLst.item(s);
            if (fstNode.getNodeType() == Node.ELEMENT_NODE)
            {
                Element fstElmnt= (Element) fstNode;

                NodeList n111=fstElmnt.getElementsByTagName("phoneid");
                Element e1    = (Element) n111.item(0);
                NodeList n112= e1.getChildNodes();
                phoneId      =((Node)n112.item(0)).getNodeValue();

                if(phoneId.equals(phoneid)){
                    NodeList n121 = fstElmnt.getElementsByTagName("name");
                    Element e2    = (Element) n121.item(0);
                    NodeList n122 = e2.getChildNodes();
                    fileName      =((Node) n122.item(0)).getNodeValue();
                    sb.append(fileName+" ");
                }
            }
        }
    }
}

```

```

    }
    return sb.toString();
}

public void add(String phoneid, String filename, String conversationid)
throws IOException, SAXException, ParserConfigurationException
{
    Document doc = db.parse(file);
    doc.getDocumentElement().normalize();

    Element data = doc.getDocumentElement();

    if (!file.delete()) System.err.println("Error...");

    Element file = doc.createElement("file");

    Element phoneId = doc.createElement("phoneid");
    phoneId.appendChild(doc.createTextNode(phoneid));
    file.appendChild(phoneId);

    Element fileName = doc.createElement("name");
    fileName.appendChild(doc.createTextNode(filename));
    file.appendChild(fileName);

    Element convId = doc.createElement("conversationid");
    convId.appendChild(doc.createTextNode(conversationid));
    file.appendChild(convId);

    data.appendChild(file);

    FileOutputStream fos = new FileOutputStream(fileName);
    OutputFormat of = new OutputFormat("XML", "ISO-8859-1", true);
    of.setIndent(1);
    of.setIndenting(true);
    XMLSerializer serializer = new XMLSerializer(fos, of);
    serializer.asDOMSerializer();
    serializer.serialize( doc.getDocumentElement() );
    fos.close();
}

```

```

public String getConvId(String phoneid, String filename)
throws IOException, SAXException, ParserConfigurationException
{
    String phoneId=null,fileName=null,convId=null;

    Document doc = db.parse(file);
    doc.getDocumentElement().normalize();

    NodeList nodeList = doc.getElementsByTagName("file");

    for (int s = 0; s < nodeList.getLength(); s++)
    {
        Node fstNode = nodeList.item(s);
        if (fstNode.getNodeType() == Node.ELEMENT_NODE)
        {
            Element fstElmnt = (Element) fstNode;

            NodeList n111=fstElmnt.getElementsByTagName("phoneid");
            Element e1 = (Element) n111.item(0);
            NodeList n112 = e1.getChildNodes();
            phoneId=((Node) n112.item(0)).getNodeValue();

            NodeList n121 = fstElmnt.getElementsByTagName("name");
            Element e2 = (Element) n121.item(0);
            NodeList n122 = e2.getChildNodes();
            fileName=((Node) n122.item(0)).getNodeValue();

            NodeList n131 =
            fstElmnt.getElementsByTagName("conversationid");
            Element e3 = (Element) n131.item(0);
            NodeList n132 = e3.getChildNodes();
            convId=((Node) n132.item(0)).getNodeValue();

            if((phoneId.equals(phoneid)) && (fileName.equals(filename)))
            {
                return convId;
            }
        }
    }
    return null;
}

```

```

public void delete(String phoneid, String filename)
throws SAXException, IOException
{
    String phoneId=null,fileName=null;

    Document doc = db.parse(file);
    doc.getDocumentElement().normalize();

    if (!file.delete()) System.err.println("Error...");

    NodeList parentlist = doc.getElementsByTagName("data");
    Node parent = parentlist.item(0);

    NodeList nodeLst = doc.getElementsByTagName("file");

    for (int s = 0; s < nodeLst.getLength(); s++)
    {
        Node fstNode = nodeLst.item(s);
        if (fstNode.getNodeType() == Node.ELEMENT_NODE)
        {
            Element fstElmnt = (Element) fstNode;

            NodeList n111 =
            fstElmnt.getElementsByTagName("phoneid");
            Element e1 = (Element) n111.item(0);
            NodeList n112 = e1.getChildNodes();
            phoneId=((Node) n112.item(0)).getNodeValue();

            NodeList n121 = fstElmnt.getElementsByTagName("name");
            Element e2 = (Element) n121.item(0);
            NodeList n122 = e2.getChildNodes();
            fileName=((Node) n122.item(0)).getNodeValue();
            if ((phoneId.equals(phoneid)) && (fileName.equals(filename)))
            {
                parent.removeChild(fstNode);
                break;
            }
        }
    }

    FileOutputStream fos = new FileOutputStream(fileName);
    OutputFormat of = new OutputFormat("XML", "ISO-8859-1", true);

```

```

        of.setIndent(1);
        of.setIndenting(true);
        XMLSerializer serializer = new XMLSerializer(fos,of);
        serializer.asDOMSerializer();
        serializer.serialize( doc.getDocumentElement() );
        fos.close();
    }
}

```

➤ **XmlFile2.java**

```

package package1;

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;
import com.sun.org.apache.xml.internal.serialize.OutputFormat;
import com.sun.org.apache.xml.internal.serialize.XMLSerializer;

public class XmlFile2 extends XmlFile
{
    public XmlFile2(String filename)
    throws ParserConfigurationException, FileNotFoundException, IOException
    {
        super(filename);
    }

    public void deleteStopedFailedFinished(String ip)
    throws IOException, SAXException
    {
        String Status=null, phoneId=null;

        Document doc = db.parse(file);
        doc.getDocumentElement().normalize();
    }
}

```



```

if (!file.delete()) System.err.println("Error...");

NodeList parentlist = doc.getElementsByTagName("data");
Node parent = parentlist.item(0);

NodeList nodeList = doc.getElementsByTagName("work");

for (int s = 0; s < nodeList.getLength(); s++)
{
    Node fstNode = nodeList.item(s);
    if (fstNode.getNodeType() == Node.ELEMENT_NODE )
    {
        Element fstElmnt= (Element) fstNode;

        NodeList n111 =
        fstElmnt.getElementsByTagName("phoneid");
        Element e1     = (Element) n111.item(0);
        NodeList n112 = e1.getChildNodes();
        phoneId       = ((Node) n112.item(0)).getNodeValue();

        NodeList n121 =
        fstElmnt.getElementsByTagName("status");
        Element e2     = (Element) n121.item(0);
        NodeList n122 = e2.getChildNodes();
        Status        =((Node) n122.item(0)).getNodeValue();

        if((phoneId.equals(ip))&& ( !(Status.equals("functionStarted")) ) )
        {
            parent.removeChild(fstNode);
        }
    }
}

FileOutputStream fos = new FileOutputStream(fileName);
OutputFormat of = new OutputFormat("XML", "ISO-8859-1", true);
of.setIndent(1);
of.setIndenting(true);
XMLSerializer serializer = new XMLSerializer(fos, of);
serializer.asDOMSerializer();
serializer.serialize( doc.getDocumentElement() );
fos.close();
}

```

```

public String getFunctionConversationId(String phoneid,
String functionname, String[] inputs) throws SAXException, IOException
{
    String phoneId=null,functionName=null,functioninput=null,convId=null;

    Document doc = db.parse(file);
    doc.getDocumentElement().normalize();

    NodeList nodeList = doc.getElementsByTagName("work");
    for (int s = 0; s < nodeList.getLength(); s++)
    {
        Node fstNode = nodeList.item(s);
        if (fstNode.getNodeType() == Node.ELEMENT_NODE)
        {
            Element fstElmnt= (Element) fstNode;

            NodeList n111 =
            fstElmnt.getElementsByTagName("phoneid");
            Element e1      = (Element) n111.item(0);
            NodeList n112 = e1.getChildNodes();
            phoneId        = ((Node) n112.item(0)).getNodeValue();

            NodeList n121 =
            fstElmnt.getElementsByTagName("functionname");
            Element e2      = (Element) n121.item(0);
            NodeList n122 = e2.getChildNodes();
            functionName   = ((Node) n122.item(0)).getNodeValue();

            if((phoneId.equals(phoneid))&&(functionName.equals(functionname)))
            {
                NodeList n131=
                fstElmnt.getElementsByTagName("inputs");
                Element e3      = (Element) n131.item(0);
                NodeList n1      = e3.getElementsByTagName("input");
                boolean flag = true;
                if ( inputs.length != n1.getLength())
                {
                    flag = false;
                }
                for (int i=0; (i<n1.getLength())&&( flag ); i++)
                {

```

```

        Element e    = (Element) nl.item(i);
        NodeList nli= e.getChildNodes();
        functioninput=((Node)nli.item(0)).getNodeValue();

        if( !functioninput.equals(inputs[i]) )
        {
            flag = false;
            break;
        }
    }

    if (flag)
    {
        NodeList nl41 =
        fstElmnt.getElementsByTagName("conversationid");
        Element e4 = (Element) nl41.item(0);
        NodeList nl42 = e4.getChildNodes();
        convId=((Node) nl42.item(0)).getNodeValue();

        return convId;
    }
}

}

return null;
}

```

```

public void add(String phoneid, String functionname, String[] inputs,
String[] outputs, String status, String timebegin, String timeend,
String conversationid)
throws IOException, SAXException, ParserConfigurationException
{
    Document doc = db.parse(file);
    doc.getDocumentElement().normalize();

    Element data = doc.getDocumentElement();

    if (!file.delete()) System.err.println("Error...");

    Element work = doc.createElement("work");

```

```

Element phoneId = doc.createElement("phoneid");
phoneId.appendChild(doc.createTextNode(phoneid));
work.appendChild(phoneId);

Element functionName = doc.createElement("functionname");
functionName.appendChild(doc.createTextNode(functionname));
work.appendChild(functionName);

Element functionInputs = doc.createElement("inputs");
Element input;
for (int i=0;i<inputs.length;i++)
{
    input = doc.createElement("input");
    input.appendChild(doc.createTextNode(inputs[i]));
    functionInputs.appendChild(input);
}
work.appendChild(functionInputs);

Element functionOutputs = doc.createElement("outputs");
Element output;
for (int i=0;i<outputs.length;i++)
{
    output = doc.createElement("output");
    output.appendChild(doc.createTextNode(outputs[i]));
    functionOutputs.appendChild(output);
}
work.appendChild(functionOutputs);

Element functionStatus = doc.createElement("status");
functionStatus.appendChild(doc.createTextNode(status));
work.appendChild(functionStatus);

Element functiontimebegin = doc.createElement("timebegin");
functiontimebegin.appendChild(doc.createTextNode(timebegin));
work.appendChild(functiontimebegin);

Element functiontimeend = doc.createElement("timeend");
functiontimeend.appendChild(doc.createTextNode(timeend));
work.appendChild(functiontimeend);

```

```

Element convId = doc.createElement("conversationid");
convId.appendChild(doc.createTextNode(conversationid));
work.appendChild(convId);

data.appendChild(work);

FileOutputStream fos = new FileOutputStream(fileName);
OutputFormat of = new OutputFormat("XML", "ISO-8859-1", true);
of.setIndent(1);
of.setIndenting(true);
XMLSerializer serializer = new XMLSerializer(fos, of);
serializer.asDOMSerializer();
serializer.serialize( doc.getDocumentElement() );
fos.close();

}

public void setStatus(String conversationid, String status, long timeend)
throws SAXException, IOException
{
    String conversationId=null;

    Document doc = db.parse(file);
    doc.getDocumentElement().normalize();

    NodeList nodeList = doc.getElementsByTagName("work");

    if (!file.delete()) System.err.println("Error...");

    for (int s = 0; s < nodeList.getLength(); s++)
    {

        Node fstNode = nodeList.item(s);
        if (fstNode.getNodeType() == Node.ELEMENT_NODE)
        {
            Element fstElmnt= (Element) fstNode;

            NodeList n111 =
            fstElmnt.getElementsByTagName("conversationid");
            Element e1      = (Element) n111.item(0);

```

```

NodeList nl12 = e1.getChildNodes();
conversationId= ((Node) nl12.item(0)).getNodeValue();

if (conversationId.equals(conversationid))
{
    NodeList nl21 =
        fstElmnt.getElementsByTagName("status");
    Element e2    = (Element) nl21.item(0);
    NodeList nl22 = e2.getChildNodes();
    ((Node) nl22.item(0)).setNodeValue(status);

    NodeList nl31 =
        fstElmnt.getElementsByTagName("timeend");
    Element e3    = (Element) nl31.item(0);
    NodeList nl32 = e3.getChildNodes();
    ((Node) nl32.item(0)).setNodeValue(""+timeend);
}
}
}

FileOutputStream fos = new FileOutputStream(fileName);
OutputFormat of = new OutputFormat("XML","ISO-8859-1",true);
of.setIndent(1);
of.setIndenting(true);
XMLSerializer serializer = new XMLSerializer(fos,of);
serializer.asDOMSerializer();
serializer.serialize( doc.getDocumentElement() );
fos.close();

}

```

```

public String getExistingWorks(String phoneid)
throws SAXException, IOException
{
    String phoneId=null,functionName=null,functioninput=null,
        convId=null,status=null;

    long duration;
    StringBuffer sb = new StringBuffer();

```

```

Document doc = db.parse(file);
doc.getDocumentElement().normalize();

NodeList nodeList = doc.getElementsByTagName("work");
for (int s = 0; s < nodeList.getLength(); s++)
{
    Node fstNode = nodeList.item(s);

    if (fstNode.getNodeType() == Node.ELEMENT_NODE)
    {
        if (s!=0)
        {
            sb.append("#");
        }

        Element fstElmnt= (Element) fstNode;

        NodeList n111 =
        fstElmnt.getElementsByTagName("phoneid");
        Element e1     = (Element) n111.item(0);
        NodeList n112 = e1.getChildNodes();
        phoneId       = ((Node) n112.item(0)).getNodeValue();
        if(phoneId.equals(phoneid))
        {
            NodeList n141 =
            fstElmnt.getElementsByTagName("conversationid");
            Element e4 = (Element) n141.item(0);
            NodeList n142 = e4.getChildNodes();
            convId= ((Node) n142.item(0)).getNodeValue();

            sb.append(convId+"#");

            NodeList n121 =
            fstElmnt.getElementsByTagName("functionname");
            Element e2     = (Element) n121.item(0);
            NodeList n122 = e2.getChildNodes();
            functionName =
            ((Node) n122.item(0)).getNodeValue();
            functionName =
functionName.substring(functionName.lastIndexOf('/')+1,functionName.length())
;

```

```

sb.append(functionName+":");

NodeList nl31=
fstElmnt.getElementsByTagName("inputs");
Element e3  = (Element) nl31.item(0);
NodeList nl  = e3.getElementsByTagName("input");
for (int i=0; (i<nl.getLength()) ; i++)
{
    Element e  = (Element) nl.item(i);
    NodeList nli = e.getChildNodes();
    functioninput= ((Node)

    nli.item(0)).getNodeValue();
    sb.append(functioninput+" ");
}

NodeList nl61 =
fstElmnt.getElementsByTagName("status");
Element e6  = (Element) nl61.item(0);
NodeList nl62 = e6.getChildNodes();
Status = ((Node) nl62.item(0)).getNodeValue();

sb.append(" - "+status);

NodeList nl71 =
fstElmnt.getElementsByTagName("timebegin");
Element e7  = (Element) nl71.item(0);
NodeList nl72 = e7.getChildNodes();

String tbegin =
((Node) nl72.item(0)).getNodeValue();

if (status.equals("functionStarted"))
{
    duration = System.currentTimeMillis() -
                Long.parseLong(tbegin);
}
else
{
    NodeList nl81 =

```



```

        fstElmnt.getElementsByTagName("timeend");
        Element e8      = (Element) nl81.item(0);
        NodeList nl82 = e8.getChildNodes();

        String tend     =
            ((Node) nl82.item(0)).getNodeValue();

        duration        = Long.parseLong(tend) -
                          Long.parseLong(tbegin);
    }

    sb.append(" - Total time: "+duration+" milliseconds");
}
}
return sb.toString();
}
}
}

```

➤ XML.java

```

package package1;
public class XML
{
    public static String getText(String s, String childname)
    {
        int indexstart = s.indexOf("<"+childname+">");
        int indexend   = s.indexOf("</"+childname+">");
        if((indexstart>=0)&&(indexend>=0))
        {
            int offset = childname.length()+2;
            String response = s.substring(indexstart+offset, indexend);
            return response;
        }
        else
        {
            return "Nothing found";
        }
    }
}

```

Gria

➤ Gria.java

```
package packagel;

import java.io.File;
import java.io.IOException;
import java.rmi.RemoteException;
import javax.activation.DataHandler;
import javax.activation.FileDataSource;
import javax.xml.parsers.ParserConfigurationException;
import org.apache.axis.message.addressing.EndpointReferenceType;
import org.xml.sax.SAXException;
import uk.ac.soton.ecs.iam.grid.client.staterepos.MemoryStateRepository;
import uk.ac.soton.ecs.iam.grid.comms.client.DataConversation;
import uk.ac.soton.ecs.iam.grid.comms.client.JobConversation;
import uk.ac.soton.ecs.iam.grid.comms.client.RemoteDataService;
import uk.ac.soton.ecs.iam.grid.comms.client.RemoteJobService;
import uk.ac.soton.ecs.iam.grid.comms.client.StateRepository;
import uk.ac.soton.itinnovation.grid.comms.wsrf.ResourceNotDestroyedFault;
import uk.ac.soton.itinnovation.grid.comms.wsrf.ResourceUnknownFault;
import uk.ac.soton.itinnovation.grid.service.types.JobStatus;
import uk.ac.soton.itinnovation.grid.types.ConversationID;

public class Gria
{
    private static String DATA_SERVICE_ENDPOINT =
        "https://localhost:8443/gria-basic-app-services/services/DataService";
    private static String JOB_SERVICE_ENDPOINT =
        "https://localhost:8443/gria-basic-app-services/services/JobService";

    public static String OK = "ok";
    public static String ERROR = "error";
    public static String EXISTS_IN_DATABASE = "exist";
    public static String NOT_EXISTS_IN_DATABASE = "notexist";

    private XmlFile1 imagefile, textfile;
    private XmlFile2 functionfile;

    private String downloadFolder, uploadFolder;
```

```

public Gria(String uploadfolder, String downloadfolder)
throws IOException, ParserConfigurationException
{
    imagefile = new XmlFile1 ("C:/apache tomcat/apache-tomcat-
                               6.0.13/webapps/mydata/image");
    textfile  = new XmlFile1 ("C:/apache tomcat/apache-tomcat-
                               6.0.13/webapps/mydata/text");

    functionfile = new XmlFile2("C:/apache tomcat/apache-tomcat-
                               6.0.13/webapps/mydata/functions");

    uploadFolder  = uploadfolder;
    downloadFolder = downloadfolder;
}

/* ----- */
/*           FUNCTION-PART 1
/* ----- */

/** FUNCTION FOR UPLOAD - DOWNLOAD - DELETE DATA **/
/** Upload to Gria the given file **/
private String upload(String filename) throws RemoteException
{
    StateRepository repository = new MemoryStateRepository();
    RemoteDataService dataService = (RemoteDataService)repository.
        getOrCreateObject(RemoteDataService.class,
            ConversationID.getEPR(DATA_SERVICE_ENDPOINT));
    DataConversation data = dataService.createStagingArea(filename);
    String DATA_STAGER_ID =
        ConversationID.getConversationFromEPR(data.getEndpointRef());
    DataHandler inputHandler= new DataHandler(
                               new FileDataSource(filename));
    data.save(inputHandler);
    return DATA_STAGER_ID;
}

/** Download from Gria a file ( image or icon ) giving the conversation
    id and the name for downloaded file **/
private File download(String conversationId,String filename)
throws RemoteException, IOException
{
    File f=null;
    StateRepository repository = new MemoryStateRepository();

```

```

RemoteDataService dataService =
    (RemoteDataService) repository.getOrCreateObject (RemoteDataService.
        class, ConversationID.getEPR (DATA_SERVICE_ENDPOINT));
DataConversation data = null;
EndpointReferenceType[] eprs = dataService.getResources ();
for (EndpointReferenceType epr : eprs)
{

if (ConversationID.getConversationFromEPR (epr).equals (conversationId))
    {
        data =
            repository.getOrCreateObject (DataConversation.class, epr);
    }
}
data.read (f=new File (filename));
return f;
}

/** Delete data from Gria with the given conversation id */
private void delete (String conversationId)
throws ResourceUnknownFault, ResourceNotDestroyedFault, RemoteException
{
    StateRepository repository = new MemoryStateRepository ();
    RemoteDataService dataService =
        (RemoteDataService) repository.getOrCreateObject (RemoteDataService.
            class, ConversationID.getEPR (DATA_SERVICE_ENDPOINT));
    DataConversation data = null;
    EndpointReferenceType[] eprs = dataService.getResources ();

    for (EndpointReferenceType epr : eprs)
    {
        if (ConversationID.getConversationFromEPR (epr).equals (conversationId))
        {
            data =
                repository.getOrCreateObject (DataConversation.class, epr);
        }
    }
    data.destroy ();
}

```

```

/** ATTENTION
 * Delete ALL data from Gria ! Only used from Gria manager and me! */
private static void deleteAllData()throws RemoteException
{
    StateRepository repository = new MemoryStateRepository();
    RemoteDataService dataService =
        (RemoteDataService)repository.getOrCreateObject(RemoteDataService.
        class,ConversationID.getEPR(DATA_SERVICE_ENDPOINT));
    DataConversation data = null;
    EndpointReferenceType[] eprs = dataService.getResources();

    for(EndpointReferenceType epr : eprs)
    {
        data =
            repository.getOrCreateObject(DataConversation.class,epr);
        data.destroy();
    }
}

/** FUNCTION EXECUTION */

/** Start the execution of the function and return the conversation id
 (... before the function ends...) */
private String startFunction(String mobileip,String applicationUrl,
fileInput[] inputfiles, fileInput[] outputfiles)
{
    try
    {
        StateRepository repository = new MemoryStateRepository();
        RemoteJobService jobService =
            (RemoteJobService)repository.getOrCreateObject(RemoteJobService.
            class,ConversationID.getEPR(JOB_SERVICE_ENDPOINT));
        JobConversation jobConv =
            jobService.createJob(applicationUrl, null, "");

        String conversationId =
            ConversationID.getConversationFromEPR(jobConv.getEndpointRef());

        FunExecution thread =
            new FunExecution(mobileip,jobConv,inputfiles,outputfiles);
        thread.start();
    }
}

```

```

        return conversationId;
    }
    catch(Exception e)
    {
        return null;
    }
}

/** Stop the execution of the function with the given conversation id
**/
public String stopFunctionExecution(String conversationId)
{
    try
    {
        StateRepository repository      = new MemoryStateRepository();

        EndpointReferenceType epr      =
        ConversationID.getEPR(JOB_SERVICE_ENDPOINT+"#" + conversationId);
        JobConversation      jobConv    =
        repository.getOrCreateObject(JobConversation.class, epr);

        //jobConv.finishJob();
        jobConv.destroy();

        return "Function stopped succesfully.";
    }
    catch(Exception e)
    {
        return "Error while trying to stop function execution.";
    }
}

private class FunExecution extends Thread
{
    private JobConversation jobConv;
    private fileInput[]      inputfile,outputfile;
    private String          mobileip;

    public FunExecution(String mobileip,JobConversation jobConv,
fileInput[] inputfiles,fileInput[] outputfiles)
    {

```

```

        this.jobConv          = jobConv;
        this.inputfile        = inputfiles;
        this.outputfile       = outputfiles;
        this.mobileip         = mobileip;
    }

    public void run()
    {
        try
        {
            DataConversation[] inputs = jobConv.getInputs();
            DataConversation[] outputs = jobConv.getOutputs();

            // Give inputs //
            for (int i=0; i<inputfile.length;i++)
            {
                if (inputfile[i].newDataStager())
                {
                    inputs[i].save(new DataHandler(new FileDataSource(
                        uploadFolder+""+inputfile[i].getFileName())));

                    String DATA_STAGER_ID =
ConversationID.getConversationFromEPR(inputs[i].getEndpointRef());

                    System.out.println("DATA_STAGER_ID:"+DATA_STAGER_ID+":");
                }
                else
                {
                    String filename = inputfile[i].getFileName();

                    String convId = getConvId(mobileip, filename);

                    StateRepository repository = new MemoryStateRepository();

                    RemoteDataService          dataService          =
                    (RemoteDataService) repository.getOrCreateObject(RemoteDataService
                    .class, ConversationID.getEPR(DATA_SERVICE_ENDPOINT));

                    DataConversation data = null;
                    EndpointReferenceType[] eprs =
                    dataService.getResources();

```

```

        for(EndpointReferenceType epr : eprs)
        {

if(ConversationID.getConversationFromEPR(epr).equals(convid))
        {

                                data =
                                repository.getOrCreateObject(DataConversation.class,epr);
                                }
                                }
                                if (data == null) System.out.println("ErrorErrorError...");
                                inputs[i].copyFrom(data);
                                }
                                }

jobConv.submitJob(null, new String[] {});

// poll job status
while(jobConv.checkJob().getInProgress()) {
    System.out.println(".");
    Thread.sleep(2000);
}

JobStatus jobStatus = jobConv.checkJob();

if(jobStatus.getExitStatus() != 0) {
    // handle the failure
    throw new RuntimeException("job failed. Here is the log:\n" +
jobStatus.getLogText());
}

// Take outputs //
for (int i=0;i<outputfile.length;i++)
{
    StateRepository repository = new MemoryStateRepository();
    RemoteDataService dataService =
(RemoteDataService) repository.getOrCreateObject(RemoteDataService.
class,ConversationID.getEPR(DATA_SERVICE_ENDPOINT));

```



```

        DataConversation data2 =
        dataService.createStagingArea(outputfile[i].getFileName());
        String DATA_STAGER_ID2 =
        ConversationID.getConversationFromEPR(data2.getEndpointRef());
        data2.copyFrom(outputs[0]);
        updateDataDatabase(mobileip, outputfile[i].getFileName(),
                           DATA_STAGER_ID2);
    }

    // clean up resources at the server
    jobConv.destroy();
functionfile.setStatus(ConversationID.getConversationFromEPR(
jobConv.getEndpointRef()), "functionFinisted", System.currentTimeMillis());

    }
    catch(Exception e)
    {
        System.out.println("Error: "+e.toString());
        try
        {

functionfile.setStatus(ConversationID.getConversationFromEPR(
jobConv.getEndpointRef()), "functionFailedOrStopped", System.currentTimeMillis());
        }
        catch(Exception ee)
        {
            System.out.println("Error: "+ee.toString());
        }
    }
}

/* ----- */
/*          FUNCTION-PART 2 USE FUNCTION-PART 1
/* ----- */

public boolean fileExist(String mobileip, String filename)
throws IOException, SAXException, ParserConfigurationException
{
    XmlFile1 xmlfile = type(filename);

```

```

String  convId  = xmlfile.getConvId(mobileip, filename);

if (convId!=null)
{
    return true;
}
else
{
    return false;
}
}

/** Upload file from upload directory to Gria  and update the xml
    database [ store the conversation id ]. */
public String uploadFile(String mobileip,String filename)
throws IOException, SAXException, ParserConfigurationException
{
    XmlFile1 xmlfile = type(filename);

    String  convId  = xmlfile.getConvId(mobileip, filename);
    if (convId!=null)
    {
        return EXISTS_IN_DATABASE;
    }
    else
    {
        convId = upload(uploadFolder+filename);
        xmlfile.add(mobileip, filename, convId);
        return OK;
    }
}

/** Download file from Gria [ find the conversation id from xml
    database ] to download directory  and update the xml database . */
public String downloadFile(String mobileip,String filename)
throws IOException, SAXException, ParserConfigurationException
{
    XmlFile1 xmlfile = type(filename);
    String  convId  = xmlfile.getConvId(mobileip, filename);
    if (convId==null)

```

```

        {
            return NOT_EXISTS_IN_DATABASE;
        }
        else
        {
            download(convid,downloadFolder+filename);
            return OK;
        }
    }

public String clearFunctionExecutionHistory(String ip)
{
    try
    {
        functionfile.deleteStopedFailedFinished(ip);
        return "We clear function execution history.";
    }
    catch(Exception e)
    {
        return "Error...";
    }
}

/** Delete file from Gria [ find the conversation id from xml database
    and update the xml database . */
public String deleteFile(String mobileip,String filename)
throws IOException, SAXException, ParserConfigurationException
{
    XmlFile1 xmlfile = type(filename);
    String convid = xmlfile.getConvId(mobileip,filename);
    if (convid==null)
    {
        return NOT_EXISTS_IN_DATABASE;
    }
    else
    {
        delete(convid);
        xmlfile.delete(mobileip, filename);
        return OK;
    }
}

```

```

public String getImageList(String mobileip)
throws IOException, SAXException, ParserConfigurationException
{
    return imagefile.getFileNames(mobileip);
}

public String getTextList(String mobileip)
throws IOException, SAXException, ParserConfigurationException
{
    return textfile.getFileNames(mobileip);
}

public String findFiles(String mobileip)
throws IOException, SAXException, ParserConfigurationException
{
    StringBuffer sb = new StringBuffer();
    sb.append("TextFiles: " +textfile.getFileNames(mobileip) +"\n");
    sb.append("ImageFiles: "+imagefile.getFileNames(mobileip)+"\n");
    return sb.toString();
}

private XmlFile1 type(String filename)
{
    if (filename.endsWith(".png"))
    {
        return imagefile;
    }
    else if (filename.endsWith(".txt"))
    {
        return textfile;
    }
    else
    {
        return null;
    }
}

/** FUNCTION FOR REMOTE FUNCTION EXECUTION **/

public String getRunningFunctionInfo(String ip)
throws SAXException, IOException
{
    return functionfile.getExistingWorks(ip);
}

```

```

/** Start executing the function with the given application url and
inputs ( and the name we want for the outputs ) and return. */
public String startFunctionExecution(String mobileip,
String applicationUrl,fileInput[] inputfiles,fileInput[] outputfiles)
throws SAXException, IOException, ParserConfigurationException
{
    String[] inputfiless = new String[inputfiles.length];
    for(int i=0;i<inputfiles.length;i++)
        inputfiless[i] = inputfiles[i].getFileName();

    String[] outputfiless= new String[outputfiles.length];
    for(int i=0;i<outputfiles.length;i++)
        outputfiless[i] = outputfiles[i].getFileName();

    String convId = functionfile.getFunctionConversationId(mobileip,
        applicationUrl, inputfiless);
    if (convId == null)
    {
        convId = startFunction(mobileip,applicationUrl,inputfiles,outputfiles);
        functionfile.add(mobileip, applicationUrl, inputfiless, outputfiless,
            "functionStarted",""+System.currentTimeMillis(),"0", convId);
        return "Work started...Please wait.";
    }
    else
    {
        return "Work already exist.";
    }
}

private String getConvId(String mobileip,String filename)
throws IOException, SAXException, ParserConfigurationException
{
    XmlFile1 xmlfile = type(filename);

    String convId = xmlfile.getConvId(mobileip, filename);

    return convId;
}

```

```
private String updateDataDatabase(String mobileip, String filename,
String convid)
throws IOException, SAXException, ParserConfigurationException
{
    XmlFile1 xmlfile = type(filename);
    xmlfile.add(mobileip, filename, convid);
    return OK;
}
}
```

Ask Mobile Phone

➤ **Mobile.java**

```
package packagel;

import java.io.BufferedReader;
import java.io.BufferedOutputStream;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.InetAddress;
import java.net.Socket;
import java.util.Iterator;

public class Mobile
{
    public static String searchForFile(String filename) throws IOException
    {
        Iterator<String> it = (GLOBAL.arrayip).iterator();
        while(it.hasNext())
        {
            String iip = it.next();
            String anwer = askQuestion(iip,"search",filename);
            if(anwer.equals("ok"))
            {
                return iip;
            }
        }

        return "Nothing found.";
    }

    public static String askQuestion(String ip, String action,
String filename) throws IOException
    {
        InetAddress serverAddr = InetAddress.getByName(ip);
        Socket socket = new Socket (serverAddr, 4444);
```

```

OutputStream os                = socket.getOutputStream ();
BufferedOutputStream bos       = new BufferedOutputStream (os);
DataOutputStream dos           = new DataOutputStream (bos);

dos.writeUTF(action+": "+filename);
dos.flush();

InputStream          is        = socket.getInputStream();
BufferedInputStream bis       = new BufferedInputStream(is);
DataInputStream      dis       = new DataInputStream(bis);

String response = dis.readUTF();

dos.close();
dis.close();

socket.close();

return response;
}
}

```

Other

➤ **fileInput.java**

```

package packagel;

public class fileInput
{
    private String filename;
    private boolean flag;

    public fileInput(String name, String from)
    {
        filename = name;
        if (from.equals("gria"))
        {
            flag = false;
        }
    }
}

```



```

        }
        else
        {
            flag = true;
        }
    }
    public String getFileName()
    {
        return filename;
    }
    public boolean newDataStager()
    {
        return flag;
    }
}

```

➤ **GLOBAL.java**

```

package packagel;

import java.util.ArrayList;

public class GLOBAL
{
    public static ArrayList<String> arrayip    = null;
    public static boolean      flag          = false;
    public static String[]     functionName = null;
    public static String[]     functionUrl  = null;
    public static String       functionname = null;
    public static int          index        = 0;
}

```

➤ **MyStringTokenizer.java**

```

package packagel;

import java.util.ArrayList;

public class MyStringTokenizer
{
    private String    s;

```

```

private char[]    array;
private char[]    mystring;
private int       index;
private ArrayList<String> arrayindex;
private String[]  myindex;
private int       total;

public MyStringTokenizer(String s1, String s2)
{
    s = s1;
    array = s1.toCharArray();
    mystring = s2.toCharArray();
    arrayindex = new ArrayList<String>();

    total = 0;
    for(int i=0;i<array.length;i++)
    {
        boolean flag;
        if(i+mystring.length<=array.length)
        {
            flag = true;
        }
        else
        {
            flag = false;
        }

        for (int j=i; (j<i+mystring.length)&&(flag);j++)
        {
            if (array[j] != mystring[j-i])
            {
                flag = false;
                break;
            }
        }
        if (flag)
        {
            total++;
            arrayindex.add((String) ""+i);
        }
    }
}

```

```

        if (total!=0)
        {
            myindex = (String[]) arrayindex.toArray(
                new String[arrayindex.size()]);
            index    = 0;
        }
    }

    public int getNumber()
    {
        return total;
    }

    public boolean hasMoreTokens()
    {
        if (index < total)
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public String nextToken()
    {
        int begin = Integer.parseInt(myindex[index]);
        int end;
        if(index+1<total)
        {
            end    = Integer.parseInt(myindex[index+1]);
        }
        else
        {
            end    = s.length();
        }
        index++;
        return s.substring(begin, end);
    }
}

```

10.3 AskMobilePhone

➤ main.java

```
package package1;
import javax.swing.JFrame;
public class main
{
    public static void main(String[] args)
    {
        myFrame f= new myFrame();
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setVisible(true);
    }
}
```

➤ myFrame.java

```
package package1;
import java.awt.Component;
import java.awt.Dimension;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.util.StringTokenizer;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
```

```

import javax.swing.JTextArea;
import javax.swing.JTextField;

public class myFrame extends JFrame
{
    private GridBagLayout        layout;
    private GridBagConstraints  constraints;
    private JPanel                panel, helppanel;
    private JLabel               label1, label2, label3;
    private JButton              find, execute;
    private JTextField            ip;
    private JTextArea            area, area2;
    private JComboBox            list;

    public myFrame()
    {
        super("Ask Mobile");
        int width      = 350;
        int height     = 380;
        Dimension screen = Toolkit.getDefaultToolkit().getScreenSize();
        int x = (screen.width - width) / 2;
        int y = (screen.height - height) / 2;
        setBounds(x, y, width, height);

        label1 = new JLabel ("Supported functions from an Android
Mobile", JLabel.CENTER);

        label2 = new JLabel ("Mobile IP: ", JLabel.RIGHT);
        label3 = new JLabel ("Functions: ", JLabel.RIGHT);

        find    = new JButton("Ask mobile for supported Functions.");
        execute = new JButton("E X E C U T E");

        ip      = new JTextField(15);
        ip.setText("127.0.0.1");
    }
}

```

```

area    = new JTextArea(7,25);
area.setEditable(false);

    JScrollPane scroller = new JScrollPane(area);
area2   = new JTextArea(3,25);
    area2.setEditable(false);
    JScrollPane scroller2 = new JScrollPane(area2);

list = new JComboBox();
list.setMaximumRowCount(5);

find.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent e) {

            String mobileip = ip.getText();

            list.removeAllItems();
            String response;
            try
            {
                response =
Mobile.askQuestion(mobileip,"supportedfunctions","");
            }
            catch(Exception ex)
            {
                response = "Error";
            }

            StringTokenizer tokens = new
StringTokenizer(response);

            while(tokens.hasMoreTokens())
            {
                list.addItem(tokens.nextToken());
            }
        }
    }
);

```

```

        }
    };

    list.addItemListener(
        new ItemListener()
        {
            @Override
            public void itemStateChanged(ItemEvent event)
            {
                if (event.getStateChange() ==
ItemEvent.SELECTED)
                {
                    String mobileip = ip.getText();
                    int index = list.getSelectedIndex();
                    String response;
                    try
                    {
                        response =
Mobile.askQuestion(mobileip, "description", ""+index);
                    }
                    catch (Exception ex)
                    {
                        response = "Error";
                    }
                    area2.setText(response);
                }
            }
        }
    );

    execute.addActionListener(
        new ActionListener() {
            public void actionPerformed(ActionEvent e) {

```

```

String mobileip = ip.getText();
int index = list.getSelectedIndex();

String response;
try
{
    response =
Mobile.askQuestion(mobileip,"execute",""+index);
}
catch(Exception ex)
{
    response = "Error 2";
}
area.setText(response);

}
}
);

```

```

layout = new GridBagLayout();
panel = new JPanel();
panel.setLayout(layout);
constraints = new GridBagConstraints();

constraints.fill=GridBagConstraints.HORIZONTAL;
constraints.anchor=GridBagConstraints.CENTER;

addComponent(label1 , 0, 0, 4, 2, 100, 30);

addComponent(new JLabel("") , 2, 0, 1, 1, 20, 10);
addComponent(label2 , 2, 1, 1, 1, 30, 10);
addComponent(ip , 2, 2, 1, 1, 30, 10);
addComponent(new JLabel("") , 2, 3, 1, 1, 20, 10);

```



```

        addComponent(new JLabel("")) , 3, 0, 1, 1, 20, 10);
        addComponent(find , 3, 1, 2, 1, 60, 10);
        addComponent(new JLabel("")) , 3, 3, 1, 1, 20, 10);

        addComponent(new JLabel("")) , 4, 0, 1, 1, 20, 10);
        addComponent(label3 , 4, 1, 1, 1, 30, 10);
        addComponent(list , 4, 2, 1, 1, 30, 10);
        addComponent(new JLabel("")) , 4, 3, 1, 1, 20, 10);

        addComponent(area2 , 5, 0, 4, 3, 100, 15);

        addComponent(new JLabel("")) , 8, 0, 1, 1, 20, 10);
        addComponent(execute , 8, 1, 2, 1, 60, 10);
        addComponent(new JLabel("")) , 8, 3, 1, 1, 20, 10);

        addComponent(area , 9, 0, 4, 6, 100, 15);

        setContentPane(panel);
    }

    private void addComponent(Component com, int row, int column, int
width, int height, int wx, int wy)
    {
        constraints.gridx=column;
        constraints.gridy=row;
        constraints.gridwidth=width;
        constraints.gridheight=height;
        constraints.weightx=wx;
        constraints.weighty=wy;

        layout.setConstraints(com,constraints);
        panel.add(com);
    }
}

```

➤ **Mobile.java**

```
package packagel;

import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.InetAddress;
import java.net.Socket;

public class Mobile
{

    public static String askQuestion(String ip, String action, String
filename) throws IOException
    {

        InetAddress serverAddr      = InetAddress.getByName(ip);
        Socket socket                = new Socket (serverAddr,
4444);

        OutputStream os              = socket.getOutputStream ();
        BufferedOutputStream bos     = new   BufferedOutputStream
(os);
        DataOutputStream dos         = new DataOutputStream (bos);

        dos.writeUTF(action+": "+filename);
        dos.flush();

        InputStream is               = socket.getInputStream();
        BufferedInputStream bis      = new
BufferedInputStream(is);
        DataInputStream dis          = new DataInputStream(bis);
```

```
String response = dis.readUTF();

dos.close();
dis.close();

socket.close();

return response;
}
}
```

(Υπογραφή)

.....

ΧΟΝΔΡΟΓΙΑΝΝΗΣ ΕΥΘΥΜΙΟΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Ευθύμιος Χονδρογιάννης

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

