Blue text on a black background

Description automatically generated

***Πολυτεχνική Σχολή***

***Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών***

***Τεχνολογία και Προγραμματισμός Κινητών Συσκευών***

***Εργασία Εξαμήνου – Lab Control***

*ΕΥΦΡΟΣΥΝΗ ΒΑΡΣΟΥ 21390021*

*ΙΩΑΝΝΗΣ ΓΟΥΒΙΑΝΑΚΗΣ 21390044*

*ΑΓΓΕΛΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ ΜΕΝΤΖΕΛΟΣ 21390132*

# Εξήγηση υλοποίησης

1. Server

O Server τρέχει σε κάθε έναν από τους υπολογιστές που πρέπει να ελέγχονται από το πρόγραμμα. Ο κάθε server ακούει σε ένα συγκεκριμένο port πάνω στο οποίο θα περιμένει tcp εντολές για να συνδεθεί με τον client του προγράμματος. Χρησιμοποιώντας sockets, συνδέεται με τον client και ξεκινά να παίρνει τις εντολές του client. Οι εντολές και οι απαντήσεις ανάμεσα στον client και στον server στέλνονται μέσω κειμένου, οι εντολές που μπορεί να πάρει ο server είναι οι εξής: echo, restart, shutdown και restore. Για το echo, ο server στέλνει σαν απάντηση το όνομα του υπολογιστή στον οποίο τρέχει ο server καθώς και το λειτουργικό του σύστημα. Για το restart, ο server στέλνει σαν απάντηση το όνομα του υπολογιστή στον οποίο τρέχει ο server καθώς το κείμενο rebooting, αντίστοιχα και για το shutdown. Για το restore, ο server στέλνει σαν απάντηση τα κείμενα, restoring και restored.

1. TCPClient

Η βασική λειτουργία του TCPClient είναι να στέλνει μια εντολή μέσω TCP στον server και να διαβάζει την απάντηση που έλαβε. Η μέθοδος **sendCommand()** παίρνει ως παραμέτρους το hostname, τo port, την εντολή προς αποστολή, το activity, ένα TextView για εμφάνιση απαντήσεων και έναν handler για επεξεργασία της απάντησης. Δημιουργεί σύνδεση TCP με τον server (Socket), στέλνει την εντολή και διαβάζει γραμμή-γραμμή την απάντηση. Για κάθε γραμμή απάντησης (εκτός από τη γραμμή "END"), ενημερώνει το UI (στο TextView) και καλεί τον handler. Αν προκύψει Exception, εμφανίζει το μήνυμα λάθους στο UI και το περνάει στον handler. Συνεπώς, το αρχείο αυτό υλοποιεί έναν απλό TCP client για αποστολή εντολών και λήψη απαντήσεων σε Android περιβάλλον.

1. Android Εφαρμογή

### Main Activity

Ο κώδικας της MainActivity είναι η βασική δραστηριότητα της εφαρμογής και οργανώνει όλη τη λειτουργικότητα του client για τον έλεγχο των υπολογιστών του εργαστηρίου.

Αρχικά, ορίζονται οι βασικές μεταβλητές που χρησιμοποιούνται για τη λειτουργία της εφαρμογής, όπως τα στοιχεία του UI (commandSpinner, computerListView, sendButton, wolButton, checkOnlineButton, responseTextView) και οι πίνακες που περιέχουν τα ονόματα, τις IP διευθύνσεις, τα MAC addresses, το λειτουργικό σύστημα και την κατάσταση (online/offline) των υπολογιστών του εργαστηρίου.

Στη μέθοδο **onCreate()** γίνεται η αρχικοποίηση των στοιχείων του UI με τη χρήση της setContentView και της findViewById για κάθε στοιχείο. Επίσης, γεμίζουν οι πίνακες με τα στοιχεία των υπολογιστών και συνδέονται οι adapters για το ListView και το Spinner. Τέλος, ορίζονται τα click listeners για τα κουμπιά, ώστε όταν ο χρήστης πατήσει κάποιο από αυτά να εκτελείται η αντίστοιχη λειτουργία.

Η μέθοδος **sendServerCommand**() διαβάζει την εντολή που έχει επιλεγεί από το Spinner και τους υπολογιστές που έχουν επιλεγεί από το ListView. Για κάθε επιλεγμένο υπολογιστή, δημιουργεί ένα νέο thread που καλεί τη μέθοδο TcpClient.sendCommand για να στείλει την εντολή στον αντίστοιχο server μέσω TCP. Ανάλογα με την απάντηση που λαμβάνει, ενημερώνει την κατάσταση του υπολογιστή (online/offline) και το λειτουργικό του σύστημα.

Η μέθοδος **checkOnline()** ενημερώνει τα ονόματα των υπολογιστών στη λίστα, προσθέτοντας το λειτουργικό σύστημα δίπλα στο όνομα, και ανανεώνει το UI ώστε να εμφανίζονται οι αλλαγές με χρήση του Computer List Adapter. Η μέθοδος **scrollResp()** κάνει responseTextView να κάνει scroll στο τέλος κάθε φορά που προστίθεται νέα απάντηση από τον server.

Η μέθοδος **doWOL()** είναι υπεύθυνη για την αποστολή των WOL πακέτων. Αρχικά, ενημερώνει τον χρήστη μέσω του TextView ότι ξεκινά η διαδικασία. Στη συνέχεια, για κάθε MAC διεύθυνση στον πίνακα computers\_mac, δημιουργείται και ξεκινά ένα νέο Thread, το οποίο καλεί τη μέθοδο sendWOLPacket() ώστε να σταλεί το αντίστοιχο WOL πακέτο.

Η μέθοδος **sendWOLPacket()** δημιουργεί και στέλνει το magic packet σε μια συγκεκριμένη MAC διεύθυνση. Αρχικά μετατρέπει την MAC σε Bytes: Η συμβολοσειρά της MAC διεύθυνσης σπάει και κάθε κομμάτι της μετατρέπεται σε αντίστοιχο byte. Αμέσως μετά δημιουργεί το Magic Packet όπου δημιουργεί έναν πίνακα με bytes που περιέχει: a) 6 φορές το byte 0xFF στην αρχή και β) 16 επαναλήψεις της MAC διεύθυνσης διότι αυτή είναι η μορφή που ορίζεται από το πρωτόκολλο Wake-on-LAN. Στην συνέχεια γίνεται η αποστολή UDP πακέτου όπου το πακέτο αποστέλλεται μέσω UDP στην broadcast διεύθυνση του δικτύου και στην θύρα 9, η οποία χρησιμοποιείται συνήθως για WOL. Για να μπορέσει να γίνει broadcast χρησιμοποιείται το DatagramSocket. Τέλος, εάν προκύψει οποιοδήποτε σφάλμα, το μήνυμα εμφανίζεται στο περιβάλλον του χρήστη μέσω του runOnUiThread().

### Computer List Adapter

Για την δημιουργία και αλλαγή χρωμάτων ώστε να εμφανίζονται οι ενεργοί και ανενεργοί υπολογιστές , δημιουργήθηκε ένας custom list adapter ο οποίος συνδέει τις λέξεις που θέλουμε μαζί με το χρώμα που χρειαζόμαστε. O adapter παίρνει τους υπολογιστές που εμφανίζονται στο listview του προγράμματος και κοιτάει για κάθε ένας από τους υπολογιστές την κατάστασή τους (offline / online). Αφού λοιπόν βρεθούν οι καταστάσεις των υπολογιστών κάνοντας override στο αρχικό get\_view, μπορεί ο adapter να αλλάξει τα χρώματα του κειμένου και την πρώτη φορά που ενεργοποιείται το πρόγραμμα και κάθε φορά που ανανεώνεται η κατάσταση κάποιου υπολογιστή.

### AndroidManifest.xml & activity\_main.xml

Το αρχείο AndroidManifest.xml, είναι το βασικό αρχείο ρυθμίσεων κάθε Android εφαρμογής. Περιγράφει δικαιώματα, δραστηριότητες και άλλες βασικές πληροφορίες της εφαρμογής. Εδώ, βάλαμε το δικαίωμα INTERNET (android.permission.INTERNET) καθώς είναι απαραίτητο για να μπορεί η εφαρμογή να κάνει TCP σύνδεση μεταξύ server και TCP client, δηλαδή να επικοινωνεί μέσω δικτύου. Χωρίς αυτό το permission, η εφαρμογή δεν θα μπορεί να στείλει ή να λάβει δεδομένα μέσω Internet ή τοπικού δικτύου. Επίσης, το permission ACCESS\_NETWORK\_STATE ελέγχει αν υπάρχει σύνδεση στο δίκτυο (WiFi ή mobile data) και να γνωρίζει την κατάσταση του δικτύου. Είναι χρήσιμο αν θέλεις να ελέγχεις αν υπάρχει διαθέσιμη σύνδεση πριν προσπαθήσεις να επικοινωνήσεις με τον server.

Το αρχείο activity\_main.xml, ορίζει τη διάταξη της κύριας οθόνης της εφαρμογής και περιλαμβάνει τα εξής βασικά components και ρυθμίσεις:

* **commandSpinner:** Spinner στο πάνω μέρος, για να επιλέγεις ποια εντολή (π.χ. Echo, Restart, Shutdown, Restore) θα σταλεί στον server.
* **computerListView:** Λίστα με όλους τους υπολογιστές του εργαστηρίου, ώστε να μπορείς να τσεκάρεις σε ποιους υπολογιστές θέλεις να στείλεις την εντολή (PC 01, … PC27).
* **sendCommandButton:** Κουμπί που στέλνει την επιλεγμένη εντολή στους υπολογιστές που έχεις επιλέξει στη λίστα.
* **WOLButton:** Κουμπί που καλεί τη λειτουργία Wake-on-LAN για να ενεργοποιήσει απομακρυσμένα τους υπολογιστές.
* **CheckOnline:** Κουμπί που ελέγχει και ενημερώνει την κατάσταση (online/offline) των υπολογιστών.
* **NestedScrollView:** Περιέχει το responseTextView, όπου εμφανίζονται οι απαντήσεις που επιστρέφει ο server για κάθε εντολή που στέλνεις.

Όλα τα παραπάνω είναι τοποθετημένα με ConstraintLayout.

1. Προεπισκόπηση Εφαρμογής

