



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Ανάπτυξη Λογισμικού για Δίκτυα και Τηλεπικοινωνίες

ΕΡΓΑΣΙΑ Νο 3

Network Monitoring application

Παναγιώτα Πουλοπούλου 1115201100110

Γρηγόριος Δημόπουλος 1115201100198

Κεφάλαιο 1

1 Κεφάλαιο 1.1

ΠΕΡΙΓΡΑΦΗ ΚΛΑΣΕΩΝ ΚΑΙ ΤΩΝ ΣΥΝΑΡΤΗΣΕΩΝ ΤΟΥΣ

android.database package

⤴ AvailableNodes.java

1. AvailableNodes.class

Συναρτήσεις:

public AvailableNodes()	Καλείται κατά την δημιουργία της κλάσης και αρχικοποιεί τη μεταβλητή nodes
public ArrayList<String> getNodes()	Επιστρέφει τη δομή nodes
public void setNodes(ArrayList<String> nodes)	Θέτει τη δομή nodes
public void insert(String a)	Εισάγει έναν κόμβο a στη λίστα nodes
public Object getProperty(int arg0)	Επιστρέφει την ιδιότητα arg0
public int getPropertyInfo (int arg0, Hashtable arg1, PropertyInfo arg2)	Επιστρέφει στο arg2 τα χαρακτηριστικά της ιδιότητας arg0
public int getPropertyCount()	Επιστρέφει τον αριθμό των ιδιοτήτων της κλάσης
public void setProperty(int arg0, Object arg1)	Θέτει την ιδιότητα arg0 με την τιμή arg1

⤴ StatisticalReport.java

1. StatisticalReport.class

Συναρτήσεις:

<code>public StatisticalReport()</code>	Κενός constructor
<code>public StatisticalReport(Integer np, Integer nip)</code>	Constructor με ορίσματα Integer np, Integer nip
<code>public StatisticalReport(String pcUUID, Integer np, Integer nip)</code>	Constructor με ορίσματα String pcUUID, Integer np, Integer nip
<code>public StatisticalReport(String UUID, String [][] MP, String [][] MIP)</code>	Constructor με ορίσματα String UUID, String [][] MP, String [][] MIP
<code>public String[][] getMPreport()</code>	Επιστρέφει τη δομή MPreport
<code>public void setMPreport(String[][] mPreport)</code>	Θέτει τη δομή MPreport
<code>public String[][] getMIPreport()</code>	Επιστρέφει τη δομή MIPreport
<code>public void setMIPreport(String[][] mIPreport)</code>	Θέτει τη δομή mIPreport
<code>public String getPcUUID()</code>	Επιστρέφει τη μεταβλητή pcUUID
<code>public void setPcUUID(String pcUUID)</code>	Θέτει τη μεταβλητή pcUUID
<code>public void ADDtoMPreport(String[] MP, Integer i)</code>	Ορίζει τη σειρά i του πίνακα MP
<code>public void ADDtoMIPreport(String[] MIP, Integer i)</code>	Ορίζει τη σειρά i του πίνακα MIP
<code>public String toString()</code>	Επιστρέφει σε ένα String τα δεδομένα της κλάσης
<code>public Object getProperty(int arg0)</code>	Επιστρέφει την ιδιότητα arg0
<code>public int getPropertyNameCount()</code>	Επιστρέφει τον αριθμό των ιδιοτήτων της κλάσης

public void getPpropertyInfo(int arg0,
Hashtable arg1, PropertyInfo arg2)

Επιστρέφει στο arg2 τα χαρακτηριστικά
της ιδιότητας arg0

public void setProperty(int arg0, Object
arg1)

Θέτει την ιδιότητα arg0 με την τιμή arg1

▲ smarthphoneDB.java

1. smarthphoneDB.class

Περιγραφή: Η κλάση αυτή χρησιμοποιείται για να γίνεται διαχείριση της βάσης της εφαρμογής σε SQLite.

Συναρτήσεις:

smarthphoneDB(Context)

Ο βασικός constructor της βάσης

addfromAvailableNodes(AvailableNodes,
String)

Εισαγωγή στοιχείων απο δεδομένα τύπου
AvailableNodes.

addfromStatisticalReport(StatisticalReport)

Εισαγωγή στοιχείων απο δεδομένα τύπου
StatisticalReport

addRecordtoAssociated_Devices(String,
String)

Εισαγωγή στοιχείου στον πίνακα
Associated_Devices.

addRecordtoClient_Interfaces(String,
String, String)

Εισαγωγή στοιχείου στον πίνακα
Client_Interfaces

addRecordtoMalicious_Ip_History(String,
String, String, String, String)

Εισαγωγή στοιχείου στον πίνακα
Malicious_Ip_History

addRecordtoMalicious_Pattern_History(Str
ing, String, String, String, String)

Εισαγωγή στοιχείου στον πίνακα
Malicious_Pattern_History

addRecordtoTask_table(String, String)

Εισαγωγή στοιχείου στον πίνακα
Task_table

deletefromTask_table(String)

Διαγραφή στοιχείου απο τον πίνακα
Task_table

<code>deleteTerminal(String)</code>	Διαγραφή τερματικού απο ολούς του πίνακες της βάσης.
<code>drop(Context)</code>	Διαγραφή βάσης.
<code>futuredelete()</code>	Συνάρτηση που βρίσκει τα τερματικά που προκείμε να διαγραφούν σύμφωνα με τις εγγραφές του Task_table
<code>onCreate(SQLiteDatabase)</code>	Καλείτε κατά την δημιουργία της βάσης και δημιουργεί τους πίνακες.
<code>onUpgrade(SQLiteDatabase, int, int)</code>	Καλείτε κατά την ανανέωση της βάσης.
<code>selectfromAssociated_Devices(String)</code>	Επέλεξε από τον πίνακα Associated_Devices αυτές τις εγγραφές που ανοίκουν στον user που δίνεται σαν όρισμα.
<code>selectfromAssociated_Devicesall()</code>	Επέλεξε από τον πίνακα Associated_Devices όλες τις εγγραφές.
<code>selectfromAssociated_DevicesUuid(String)</code>	Επέλεξε από τον πίνακα Associated_Devices όλες τις εγγραφές με uuid αυτό που ορίζεται στο όρισμα.
<code>selectfromClient_Interfaces(String, String)</code>	Επέλεξε απο τον πίνακα Client_Interfaces αυτή την εγγραφή με interface ip και uuid αυτά που δίνονται ως ορίσματα.
<code>selectfromClient_Interfacesall(String)</code>	Επέλεξε απο τον πίνακα Client_Interfaces όλες τις εγγραφές
<code>selectfromMalicious_Ip_History(String, String)</code>	Επέλεξε απο τον πίνακα Malicious_Ip_History τις εγγραφές που έχουν το δοσμένο Recency και rcuuid.
<code>selectfromMalicious_Ip_Historywithip(String, String)</code>	Επέλεξε απο τον πίνακα Malicious_Ip_History τις εγγραφές που έχουν το δοσμένο interface ip και rcuuid.

<code>selectfromMalicious_Pattern_History(String, String)</code>	Επέλεξε απο τον πίνακα <code>Malicious_Pattern_History</code> τις εγγραφές που έχουν το δοσμένο <code>Recency</code> και <code>rcuid</code> .
<code>selectfromMalicious_Pattern_Historywithip(String, String)</code>	Επέλεξε απο τον πίνακα <code>Malicious_Ip_History</code> τις εγγραφές που έχουν το δοσμένο <code>interface ip</code> και <code>rcuid</code> .
<code>selectfromTask_tableall()</code>	Επέλεξε απο τον πίνακα <code>Task_table</code> όλες τις εγγραφές.
<code>takemostrecent(ArrayList<ArrayList<String>>)</code>	Συνάρτηση που επιστρέφει τα πιο πρόσφατα στοιχεία του πίνακα <code>Malicious_Ip_History</code> ή <code>Malicious_Pattern_History</code> ανάλογα με το τι θα πάρει σαν ορίσμα σαν <code>ArrayList<ArrayList<String>></code> .
<code>takemostrecentRecordfromTask_table()</code>	Επιστροφή του πιο παλιού στοιχείου που μπήκε στο <code>Task_table</code> .

tabs package

⌘ Add_Malicious_Fragment.java

1. Add_Malicious_Fragment.class

Περιγραφή: Αυτό είναι το fragment το οποίο μπορούμε να δούμε τα malicious pattern/ip που έχουν ορισθεί στον server αλλά και να προσθέσουμε άλλα δικά μας.

Συναρτήσεις:

<code>Add_Malicious_Fragment(TabActivity)</code>	Ο βασικός constructor της κλάσης.
<code>onClick(View)</code>	Η συνάρτηση που καλείται για διαχείριση των κουμπιών όταν κάποιο απο αυτά πατιθεί.

onCreate(Bundle)	Η συνάρτηση που καλείται κατά την δημιουργία του fragment.
onCreateView(LayoutInflater, ViewGroup, Bundle)	Η συνάρτηση που καλείται κατά την δημιουργία της οθόνης του fragment.
newInstance(int, TabActivity)	Συνάρτηση που χρησιμοποιείται για την δημιουργία νέου fragment.

2. addMalicious.class extends AsyncTask<Void, Void, Void>

Περιγραφή: Η κλάση αυτή χρησιμοποιείται για αποστολή νέων malicious στον server και λήψη απάντησης αν αυτή υπάρχει.

Συναρτήσεις:

doInBackground(Void...)	Χρήση για επικοινωνία με τον αθροίστη.
onPostExecute(Void)	Τυπώμα καταλληλων μηνυμάτων αναλόγα με την έκβαση των αποτελεσμάτων της doInBackground(Void...)

3. takeMalicious.class extends AsyncTask<Void, Void, Void>

Περιγραφή: Η κλάση αυτή χρησιμοποιείται για λήψη malicious απο τον server

Συναρτήσεις:

doInBackground(Void...)	Χρήση για επικοινωνία με τον αθροίστη.
onPostExecute(Void)	Ενημέρωση δεδομένων στις λίστες με τα malicious ip/pattern του tab.

^ Interfaces Fragment.java

1. Interfaces Fragment.class

Περιγραφή: Η κλάση αυτή υλοποιεί το δεύτερο fragment/tab που φιλοξενεί τα εκάστοτε interfaces του τερματικού που έχει επιλεχθεί στο πρώτο tab.

Συναρτήσεις:

<code>newInstance(int, TabActivity)</code>	Συνάρτηση που χρησιμοποιείται για την δημιουργία νέου fragment.
<code>Interfaces_Fragment(TabActivity)</code>	Ο βασικός constructor της κλάσης.
<code>onCreate(Bundle)</code>	Η συνάρτηση που καλείται κατά την δημιουργία του fragment.
<code>onListItemClick(ListView, View, int, long)</code>	Ενέργεια που εκτελείται κατά το κλικάρισμα σε ένα στοιχείο της λίστας.

✎ Malicious.java

1. Malicious.class

Περιγραφή: Η κλάση αυτή φιλοξενεί το fragment που παρουσιάζει τα malicious pattern και τα malicious ip κάποιου συγκεκριμένου interface.

Συναρτήσεις:

<code>addCurrentmalicious(String)</code>	Εισαγωγή στοιχείου με ενημέρωση στην λίστα Currentmalicious
<code>addMIP(ArrayList<String>)</code>	Εισαγωγή στοιχείου λίστα MIP
<code>addMP(ArrayList<String>)</code>	Εισαγωγή στοιχείου λίστα MP
<code>clearCurrentmalicious()</code>	Καθάρισε την λίστα Currentmalicious με ενημέρωση του adapter.
<code>clearMIP()</code>	Καθάρισε την λίστα MIP.
<code>clearMP()</code>	Καθάρισε την λίστα MP.
<code>getAdapter_Currentmalicious()</code>	Επέστρεψε τον adapter που διαχειρίζεται την λίστα Currentmalicious.
<code>getCurrentmalicious()</code>	Επέστρεψε την λίστα Currentmalicious.

<code>getCurrentmalicious(Integer)</code>	Επέστρεψε το στοιχείο της λίστας <code>Currentmalicious</code> που βρίσκεται στην θέση που δείχνει ο <code>integer</code>
<code>getDB()</code>	Επέστρεψε την βάση
<code>getInterface_ip()</code>	Επέστρεψε την μεταβλητή <code>interface_ip</code>
<code>getInterface_name()</code>	Επέστρεψε την μεταβλητή <code>interface_name</code>
<code>getMIP()</code>	Επέστρεψε την λίστα <code>MIP</code>
<code>getMIP(Integer)</code>	Επέστρεψε το στοιχείο που βρίσκεται στην θέση που δηλώνει ο <code>integer</code> απο την λίστα <code>MIP</code> .
<code>getMP()</code>	Επέστρεψε την λίστα <code>MP</code>
<code>getMP(Integer)</code>	Επέστρεψε το στοιχείο που βρίσκεται στην θέση που δηλώνει ο <code>integer</code> απο την λίστα <code>MP</code> .
<code>getmSectionsPagerAdapter()</code>	Επιστροφή του <code>adapter</code> για το <code>FragmentMalicious</code> .
<code>getmViewPager()</code>	Επιστροφή του <code>ViewPager</code> του <code>activity</code> .
<code>getPc_Uuid()</code>	Επιστροφή του <code>Pc_Uuid</code>
<code>getSelected()</code>	Επιστροφή της μεταβλητής <code>selected</code> .
<code>onCreate(Bundle)</code>	Καλείται κατά την δημιουργία του <code>activity</code> .
<code>onCreateOptionsMenu(Menu)</code>	Καλείται κατά την δημιουργία του μενού.
<code>onKeyDown(int, KeyEvent)</code>	Καλείται κατά το πάτημα ενός κουμπιού και γίνεται διαχείριση της ενέργειας.
<code>onOptionsItemSelected(MenuItem)</code>	Καλείται κατά την επιλογή απο το <code>menu</code>

onPause()	Καλείται κατά την παύση του activity
onStop()	Καλείται κατά την διακοπή του activity
removeCurrentmalicious(String)	Διέγραψε το στοιχείο που βρίσκεται στο string απο την λίστα Currentmalicious
removeMIP(ArrayList<String>)	Διέγραψε το στοιχείο που βρίσκεται στο argument απο την λίστα MIP
removeMP(ArrayList<String>)	Διέγραψε το στοιχείο που βρίσκεται στο argument απο την λίστα MP
setAdapter_Currentmalicious(ArrayAdapte r<String>)	Όρισε τον adapter για τα Currentmalicious.
setCurrentmalicious(ArrayList<String>)	Όρισε τα Currentmalicious
setDB(smartphoneDB)	Όρισε τη βάση
setInterface_ip(String)	Όρισε το interface ip
setInterface_name(String)	Όρισε το interface name
setMIP(ArrayList<ArrayList<String>>)	Όρισε τα MIP
setMP(ArrayList<ArrayList<String>>)	Όρισε τα MP
setmSectionsPagerAdapter(Page_Adapter)	Όρισε τον adapter του MaliciousFragment
setmViewPager(ViewPager)	Όρισε το ViewPager του activity
setPc_Uuid(String)	Όρισε το Pc_Uuid
setSelected(String)	Όρισε την μεταβλητή selected

✎ MaliciousFragment.java

1. MaliciousFragment.class

Περιγραφή: Η κλάση αυτή υλοποιεί το fragment που φιλοξενεί τα εκάστοτε malicious pattern/ip ανάλογα την επιλογή απο το μενού.

Συναρτήσεις:

newInstance(int, Malicious)	Συνάρτηση που χρησιμοποιείται για την δημιουργία νέου fragment.
MaliciousFragment(Malicious)	Ο βασικός constructor της κλάσης.
onCreate(Bundle)	Η συνάρτηση που καλείται κατά την δημιουργία του fragment.
onListItemClick(ListView, View, int, long)	Η συνάρτηση που καλείται για διαχείριση των στοιχείων της λίστας όταν κάποιο απο αυτά πατιθεί.

⤴ Page Adapter.java

1. Page Adapter.class

Περιγραφή: Κλάση που ορίζει τον FragmentPagerAdapter για το fragment που έχει να κάνει με την αναλυτική παρουσίαση των malicious pattern/ip.

Συναρτήσεις:

Page_Adapter(FragmentManager, Malicious)	Ο βασικός constructor της κλάσης.
getCount()	Επιστροφή του αριθμού των fragment για διαχείριση.
getItem(int)	Επιστροφή αντικειμένου fragment ανάλογα την θέση
getPageTitle(int)	Επιστροφή του τίτλου της σελίδας.

⤴

⤴ Pages Adapter.java

1. Pages Adapter.class

Περιγραφή: Κλάση που ορίζει τον `FragmentManagerAdapter` για τα `fragments` του `swipe view`.

Συναρτήσεις:

<code>Pages_Adapter(FragmentManager, TabActivity)</code>	Ο βασικός constructor της κλάσης
<code>getCount()</code>	Επιστροφή του αριθμού των <code>fragment</code> για διαχείριση.
<code>getItem(int)</code>	Επιστροφή αντικειμένου <code>fragment</code> ανάλογα την θέση
<code>getItemPosition(Object)</code>	Καλείται όταν η <code>host view</code> προσπαθεί να καθορίσει εάν η θέση ενός στοιχείου έχει αλλάξει.
<code>getPageTitle(int)</code>	Επιστροφή του τίτλου της σελίδας.

⤴ Remove_Terminals_Fragments.java

1. Remove_Terminals_Fragments.class

Περιγραφή: Αυτό είναι το `fragment` το οποίο μπορούμε να διαγράψουμε ενα τερματικό απο το δίκτυο.

Συναρτήσεις:

<code>newInstance(int, TabActivity)</code>	Συνάρτηση που χρησιμοποιείται για την δημιουργία νέου <code>fragment</code> .
<code>Remove_Terminals_Fragment(TabActivity)</code>	Ο βασικός constructor της κλάσης.
<code>onCreate(Bundle)</code>	Η συνάρτηση που καλείται κατα την δημιουργία του <code>fragment</code> .
<code>onListItemClick(ListView, View, int, long)</code>	Ενέργεια που εκτελείται κατά το κλικάρισμα σε ένα στοιχείο της λίστας.

2. removeUUID.class extends AsyncTask<Void, Void, Void>

Περιγραφή: Η κλάση αυτή χρησιμοποιείται για αποστολή των αιτήματων για διαγραφή τερματικού στον server.

Συναρτήσεις:

doInBackground(Void...)	Χρήση για επικοινωνία με τον αθροίστη.
onPostExecute(Void)	Τυπώμα καταλληλων μηνυμάτων αναλόγα με την έκβαση των αποτελεσμάτων της doInBackground(Void...)

⤴ TabActivity.java

1. logOut.class extends AsyncTask<Void, Void, Void>

Περιγραφή: Χρησέ της κλάσης για επικοινωνία με τον αθροιστή και Logout απο το σύστημα.

Συναρτήσεις:

doInBackground(Void...)	Χρήση για επικοινωνία με τον αθροίστη.
onPostExecute(Void)	Τυπώμα καταλληλων μηνυμάτων αναλόγα με την έκβαση των αποτελεσμάτων της doInBackground(Void...).Και τελικό log out αν υπήρξε έγκριση απο τον server.

2. TabActivity.class

Περιγραφή: Η βασική κλάση/activity του πακέτου χρησιμοποιείται ως διαυλός επικοινωνίας μεταξύ των tabs/fragments και τα δημιουργεί.

Συναρτήσεις:

addAlluuids(String)	Εισαγωγή στοιχείου λίστα Alluuids με ενημέρωση του adapter.
---------------------	---

addInterafacesList(String)	Εισαγωγή στοιχείου λίστα InterafacesList με ενημέρωση του adapter.
addUuidsList(String)	Εισαγωγή στοιχείου λίστα Uuids με ενημέρωση του adapter.
clearAlluuids()	Καθάρισε την λίστα Alluuids με ενημέρωση του adapter.
clearInterafacesList()	Καθάρισε την λίστα InterafacesList με ενημέρωση του adapter.
clearUuidsList()	Καθάρισε την λίστα Uuids με ενημέρωση του adapter..
getAdapter_interfaces()	Επέστρεψε τον adapter που διαχειρίζεται την λίστα InterafacesList
getAdapter_uuid()	Επέστρεψε τον adapter που διαχειρίζεται την λίστα Uuids
getAdapter_uuid_remove()	Επέστρεψε τον adapter που διαχειρίζεται την λίστα Alluuids.
Επέστρεψε την λίστα MIP	Επέστρεψε την λίστα Alluuids
getAlluuids(Integer)	Επέστρεψε το στοιχείο που βρίσκεται στην θέση που δηλώνει ο integer απο την λίστα Alluuids
getClicked_UUID()	Επέστρεψε την μεταβλητή Clicked_UUID
getConnection()	Επέστρεψε την μεταβλητή Connection
getDB()	Επέστρεψε την βάση
getInterafacesList()	Επέστρεψε την λίστα InterafacesList
getInterafacesList(Integer)	Επέστρεψε το στοιχείο που βρίσκεται στην θέση που δηλώνει ο integer απο την λίστα

	InterafacesList
getmSectionsPagerAdapter()	Επιστροφή του adapter για τα Fragments
getmViewPager()	Επιστροφή του ViewPager του activity.
getPass()	Επιστροφή του Password
getUser()	Επιστροφή του User
getUuidsList()	Επέστρεψε την λίστα UuidsList
getUuidsList(Integer)	Επέστρεψε το στοιχείο που βρίσκεται στην θέση που δηλώνει ο integer απο την λίστα UuidsList
isAdmin()	Επέστρεψε αν ο user είναι admin ή όχι
isEnabledfiltering()	Επέστρεψε αν είναι ενεργοποιημένο το φιλτράρισμα
onCreate(Bundle)	Καλείται κατά την δημιουργία του activity
onCreateOptionsMenu(Menu)	Καλείται κατά την δημιουργία του menu του activity
onKeyDown(int, KeyEvent)	Καλείται κατά το πάτημα ενός κουμπιού και γίνεται διαχείριση της ενέργειας.
onOptionsItemSelected(MenuItem)	Καλείται κατά την επιλογή απο το menu
removeAlluuids(String)	Διέγραψε απο την λίστα Alluuids το στοιχείο στο argument
removeInterafacesList(String)	Διέγραψε απο την λίστα InterafacesList το στοιχείο στο argument
removeUuidsList(String)	Διέγραψε απο την λίστα UuidsLis το στοιχείο στο argument

setAdapter_interfaces(ArrayAdapter<String>)	Όρισε τον adapter του InterfacesList
setAdapter_uuid(ArrayAdapter<String>)	Όρισε τον adapter του UuidsList
setAdapter_uuid_remove(ArrayAdapter<String>)	Όρισε τον adapter του Alluuids
setADMIN(boolean)	Όρισε αν είναι ή όχι admin ο user
setAlluuids(ArrayList<String>)	Όρισε την λίστα Alluuids
setClicked_UUID(String)	Όρισε την μεταβλητή clicked_UUID
setConnection(Connectivity)	Όρισε την μεταβλητή Connection
setDB(smartphoneDB)	Όρισε την βάση
setEnablefiltering(boolean)	Όρισε αν θα υπάρχει filtering ή όχι.
setInterfacesList(ArrayList<String>)	Όρισε την λίστα InterfacesList
setmSectionsPagerAdapter(Pages_Adapter)	Όρισε τον adapter των fragments
setmViewPager(ViewPager)	Όρισε το ViewPager του activity
setPass(String)	Όρισε τον password
setUser(String)	Όρισε τον user
setUuidsList(ArrayList<String>)	Όρισε την λίστα UuidsList

✎ Uuids_Fragment.java

1. Uuids_Fragment.class

Περιγραφή: Αυτό είναι το fragment/tab στο οποίο μπορούμε να δούμε τα τερματικά.

Συναρτήσεις:

<code>newInstance(int, TabActivity)</code>	Η συνάρτηση που καλείται κατά την δημιουργία του fragment.
<code>Uuids_Fragment(TabActivity)</code>	Ο βασικός constructor της κλάσης.
<code>onCreate(Bundle)</code>	Η συνάρτηση που καλείται κατά την δημιουργία του fragment.
<code>onListItemClick(ListView, View, int, long)</code>	Ενέργεια που εκτελείται κατά το κλικάρισμα σε ένα στοιχείο της λίστας.

main package

MainActivity.java

Συναρτήσεις:

<code>public void onCreate(Bundle savedInstanceState)</code>	Καλείται κατά τη δημιουργία της κλάσης και ανάλογα με τα αποθηκευμένα Preferences καλεί το αντίστοιχο activity
<code>public void onCreateOptionsMenu(Menu menu)</code>	Δημιουργεί το menu
<code>public boolean onOptionsItemSelected(MenuItem item)</code>	Εκτελεί τις ενέργειες που πρέπει να γίνουν όταν επιλεγεί ένα κουμπί του menu

Connectivity.java

Συναρτήσεις:

<code>public boolean isURLReachable(Context context)</code>	Επιστρέφει true αν το URL του σέρβερ είναι προσβάσιμο, false σε κάθε άλλη περίπτωση
<code>public boolean isConnected(Context context)</code>	Επιστρέφει true αν τη συσκευή είναι συνδεδεμένη στο διαδίκτυο, false σε κάθε

άλλη περίπτωση

public String findServerURL()

Βρίσκει σε ποια διεύθυνση έχει ανέβει ο server

DataService.java

Συναρτήσεις:

public IBinder onBind (Intent intent)

Επιστρέφει το κανάλι επικοινωνίας με την υπηρεσία

public void onCreate()

Καλείται κατά τη δημιουργία της υπηρεσίας

public void onStart(Intent intent, int startId)

Καλείται στο ξεκίνημα της υπηρεσίας και εκκινεί τα βασικά νήματα

public void onDestroy()

Καλείται κατά την καταστροφή της υπηρεσίας

public smartphoneDB getDB()

Επιστρέφει την smartphoneDB

public void setDB(smartphoneDB dB)

Ορίζει την smartphoneDB

Dataservice.TaskProcessingThread

public void run()

Εκτελεί όλες τις εντολές που είχανε μπει στην λίστα αναμονής όσο δεν υπήρχε σύνδεση με τον server

Dataservice.TaskProcessingThread.addMalicious

protected Void doInBackground(Void.. params)

Καλεί την συνάρτηση insertMaliciousPatterns του server

Dataservice.TaskProcessingThread.removeUUID

protected Void doInBackground (Void.. params)	Καλεί τη συνάρτηση deletePC του server
protected void onPostExecute (Void result)	Καλείται μετά την ολοκλήρωση της doInBackground και σβήνει τον client από τη βάση του κινητού

DataService.UpdaterThread

public void run()	Ζητάει περιοδικά δεδομένα από τον server αν υπάρχει σύνδεση με αυτόν
-------------------	--

DataService.UpdaterThread.getfromServer

protected Void doInBackground(Void... params)	Καλεί τη συνάρτηση retrieveStatistics του server
protected void onPostExecute (Void result)	Αποθηκεύει τα αποτελέσματα που πήρε στη τοπική βάση του κινητού

MyPreferences.java

Συναρτήσεις:

static SharedPreferences getSharedPreferences(Context ctx)	Επιστρέφει ένα SharedPreferences instance που χρησιμοποιείται από την εφαρμογή
public static void setUserIp(Context ctx, String userIp)	Θέτει το UserIp
public static String getUserIp(Context ctx)	Επιστρέφει το UserIp
public static clearUserIp(Context ctx)	Διαγράφει το τρέχον UserIp από τα preferences του χρήστη

<code>public static void setUserPass(Context ctx, String userPass)</code>	Θέτει το UserPass
<code>public static void getUserPass(Context ctx)</code>	Επιστρέφει το UserPass
<code>public static void clearUserPass(Context ctx)</code>	Διαγράφει το τρέχον UserPass από τα preferences του χρήστη
<code>public static void setUsername(Context ctx, String userName)</code>	Θέτει το UserName
<code>public static void getUsername(Context ctx)</code>	Επιστρέφει το UserName
<code>public static void clearUserName(Context ctx)</code>	Διαγράφει το τρέχον UserName από τα preferences του χρήστη
<code>public static void setUserAdmin(Context ctx, String userAdmin)</code>	Θέτει το UserAdmin
<code>public static void getUserAdmin(Context ctx)</code>	Επιστρέφει το UserAdmin
<code>public static void clearUserAdmin(Context ctx)</code>	Διαγράφει το τρέχον UserAdmin από τα preferences του χρήστη

Login package

LoginActivity.java

Συναρτήσεις:

<code>public Boolean onCreate(Bundle savedInstanceState)</code>	Καλείται κατά τη δημιουργία του activity
<code>public Boolean onCreateOptionsMenu(Menu menu)</code>	Καλείται κατά τη δημιουργία του menu
<code>public Boolean onOptionsItemSelected(MenuItem item)</code>	Εκτελεί τις απαραίτητες ενέργειες ανάλογα με την επιλογή κουμπιού που έγινε στο menu
<code>public void onClick(View v)</code>	Εκτελεί τις αντίστοιχες ενέργειες όταν

```
public Connectivity getConnection()  
public void setConnection(Connectivity  
connection)
```

πατηθεί το κουμπί που έχει συνδεθεί με
έναν `onClickListener`

Επιστρέφει τη δομή `connection`

Ορίζει τη δομή `connection`

Login.attemptLogin

```
protected Void doInBackground(Void...  
params)  
protected void onPostExecute(Void result)
```

Καλεί την συνάρτηση `login` του `server`

Περνάει τα αποτελέσματα στη τοπική βάση
του κινητού και είτε καλεί το επόμενο
`Activity` είτε πετάει μήνυμα λάθους

Register package

Register.java

Συναρτήσεις:

```
protected void onCreate(Bundle  
savedInstanceState)  
public Boolean  
onCreateOptionsMenu(Menu menu)  
public Boolean  
onOptionsItemSelected(MenuItem item)  
public void onClick(View v)  
  
public smartphoneDB getDB()  
public void setDB(smartphone dB)
```

Καλείται κατά τη δημιουργία του `Activity`

Καλείται κατά τη δημιουργία του `menu`

Εκτελεί τις απαραίτητες ενέργειες ανάλογα
με την επιλογή κουμπιού που έγινε στο
`menu`

Εκτελεί τις αντίστοιχες ενέργειες όταν
πατηθεί το κουμπί που έχει συνδεθεί με
έναν `onClickListener`

Επιστρέφει τη δομή `smartphoneDB`

Θέτει τη δομή `smartphoneDB`

Register.sendNodes

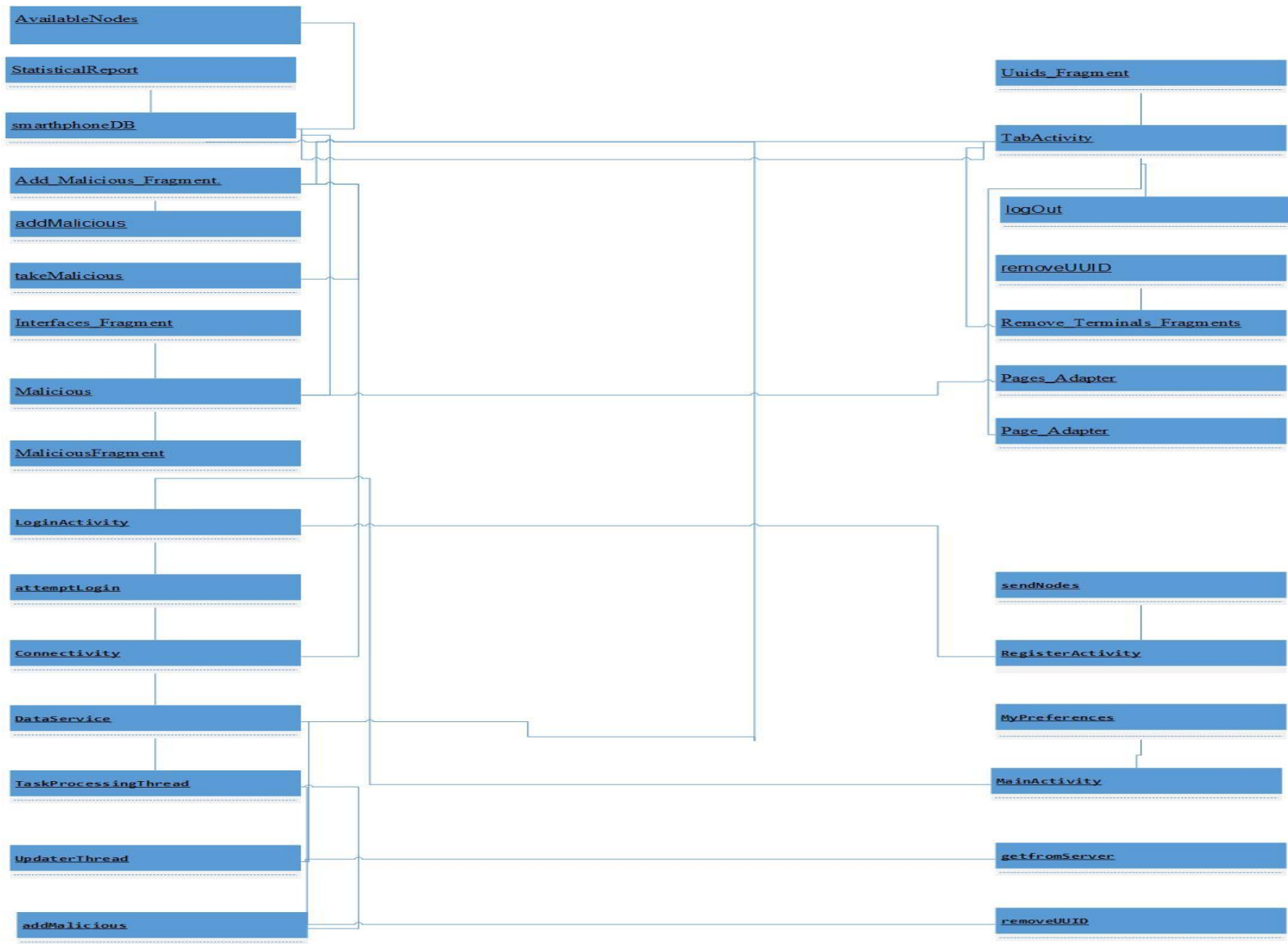
```
doInBackground(Void... params)  
protected void onPostExecute(Void result)
```

Καλεί τη συνάρτηση `register_android` του
`server`

Περνάει τα αποτελέσματα στη τοπική βάση
του κινητού και είτε καλεί το επόμενο

Activity είτε πετάει μήνυμα λάθους

Class Diagram



Περιγραφή του σχήματος της SQLite βάσης

Η βάση της εφαρμογής διαχειρίζεται από την κλάση `smarhphoneDB.class` και τις συναρτήσεις της. Η κλάση αυτή κάνει **extends** `SQLiteOpenHelper`. Έτσι η δημιουργία της βάσης γίνεται αν δεν υπάρχει στην πρώτη κλήση του constructor της κλάσης όπου αυτή γίνει αλλιώς γίνεται απλό ανοιγμά της βάσης.

Πίνακες της Βάσης:

✧ **Associated Devices** (primary key :`PC_Uuid`)

Περιγραφή: Το συγκεκριμένο table περιέχει πληροφορίες σχετικά με τα τερματικά και τους ιδιοκτήτες τους. Αν δεν γνωρίζουμε τον ιδιοκτήτη κάποιου τερματικού υπάρχει κέννη εγγραφή στο πεδίο `Ownership`.

Διαχείριση: Από τις συναρτήσεις της κλάσης `smarhphoneDB.class` που αναφέρονται στον συγκεκριμένο πίνακα της βάσης.

Συσχέτιση με άλλα tables: Ο πίνακας αυτός χρησιμοποιείται για να περνάμε τα `Uuid` όλα η ενός χρήστη και να τα δίνουμε ως ορίσματα στις συναρτήσεις των πινάκων `Malicious_Pattern_History` και `Malicious_Pattern_History` όπως και `Client_Interfaces` για να επιλέγουμε συγκεκριμένες εγγραφές.

`PC_Uuid` text

`Ownership` text

✧ **Malicious Pattern History**

(primary key: `PC_Uuid, Interface_ip, Malicious_Pattern, Recency`)

Περιγραφή: Το συγκεκριμένο table περιέχει πληροφορίες σχετικά με τα `malicious patterns` που έχουν εντοπιστεί σε κάθε terminal σε κάθε interface από αυτά που είμαστε ενήμεροι από τον server. Επιπλέον υπάρχει και το πεδίο `Recency` για να μπορεί να γίνει διαλογή του πιο πρόσφατου `StatisticalReport` από αυτόν τον πίνακα.

Διαχείριση: Από τις συναρτήσεις της κλάσης `smarhphoneDB.class` που αναφέρονται στον συγκεκριμένο πίνακα της βάσης.

`PC_Uuid`
text

`Interface_ip`
text

`Malicious_Pattern`
text

`Frequency` text

`Recency` text

✧ **Malicious Ip History**

(primary key: `PC_Uuid, Interface_ip, Malicious_Ip, Recency`)

Περιγραφή: Το συγκεκριμένο table περιέχει πληροφορίες σχετικά με τα `malicious ips` που έχουν εντοπιστεί σε κάθε terminal σε κάθε interface από αυτά που είμαστε ενήμεροι από τον server. Επιπλέον υπάρχει και το πεδίο `Recency` για να μπορεί να γίνει διαλογή του πιο πρόσφατου `StatisticalReport` από αυτόν τον πίνακα.

Διαχείριση: Από τις συναρτήσεις της κλάσης `smarhphoneDB.class` που αναφέρονται στον συγκεκριμένο πίνακα της βάσης.

`PC_Uuid`
text

`Interface_ip`
text

`Malicious_Ip` text

`Frequency` text

`Recency` text

⤴ Client Interfaces

(primary key:PC_Uuid,Interface_ip)

Περιγραφή:Το συγκεκριμένο table περιέχει πληροφορίες σχετικά με τα ονόματα των interfaces με Interface_ip στο τερματικό με Pc_Uuid

Διαχείριση:Απο τις συναρτήσεις τις κλάσης smarthphoneDB.class που αναφέρονται στον συγκεκριμένο πίνακα της βάσης.

PC_Uuid text

Interface_ip text

Interface_name text

⤴ Task table

(primary key:Time)

Περιγραφή:Το συγκεκριμένο table περιέχει πληροφορίες σχετικά με τα αιτήματα που εκκρεμούν προς τον server.Το Waiting_task υποδηλώνει τον τύπο του αιτήματος ενώ στο argument βρίσκονται τα arguments του αιτήματος.Επίσης υπάρχει το πεδίο Time για να βρίσκουμε τα πιο παλιά αιτήματα.

Διαχείριση:Απο τις συναρτήσεις τις κλάσης smarthphoneDB.class που αναφέρονται στον συγκεκριμένο πίνακα της βάσης.

Waiting_task text

Argument text

Time text

Περιγραφή Service

Η κλάση DataService.java κάνει extend το Service. Κατά την εκκίνηση του TabActivity καλείται η συνάρτηση startService η οποία ξεκινάει το αντίστοιχο service. Κατά την εκκίνηση του service καλείται η μέθοδος onStartCommand η οποία ξεκινάει δύο threads.

Το TaskprocessThread ελέγχει περιοδικά αν υπάρχει σύνδεση με τον server και το διαδίκτυο. Σε περίπτωση που υπάρχει εκτελεί ένα ένα όλα τα ετοιμάτα που είχε ορίσει ο χρήστης όσο η εφαρμογή δεν ήτανε συνδεδεμένη.

Το updateThread ελέγχει περιοδικά τη σύνδεση με τον server και το διαδίκτυο. Σε περίπτωση που υπάρχει ζητάει από τον server να του στείλει τα πιο πρόσφατα StatisticalReports που είναι αποθηκευμένα στη βάση του. Στη συνέχεια τα περνάει στην τοπική βάση της εφαρμογής

Το DataService τερματίζει όταν ο χρήστης επιλέξει να αποσυνδεθεί απο την εφαρμογή. Τα δύο threads τερματίζονται μέσα στο σώμα της onDestroy.

Παραδοχές

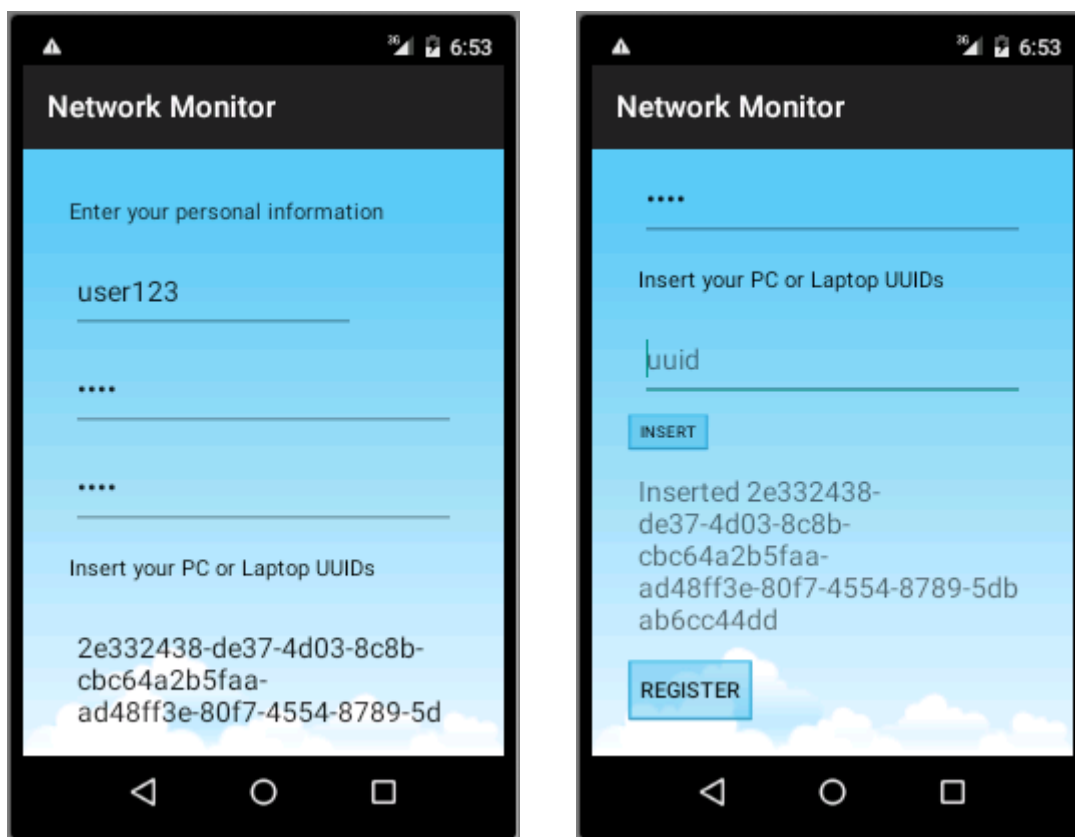
Στο TabActivity υπάρχει η επιλογή Refresh στο menu. Όταν πατηθεί η επιλογή Refresh τότε ανανεώνεται η λίστα με τα UUID που είναι περασμένα στη βάση.

Για να μπορέσει η εφαρμογή να συνδεθεί με τον server πρέπει πριν το login ή το register να οριστεί από το menu η IP στην οποία είναι ανεβασμένος ο server.

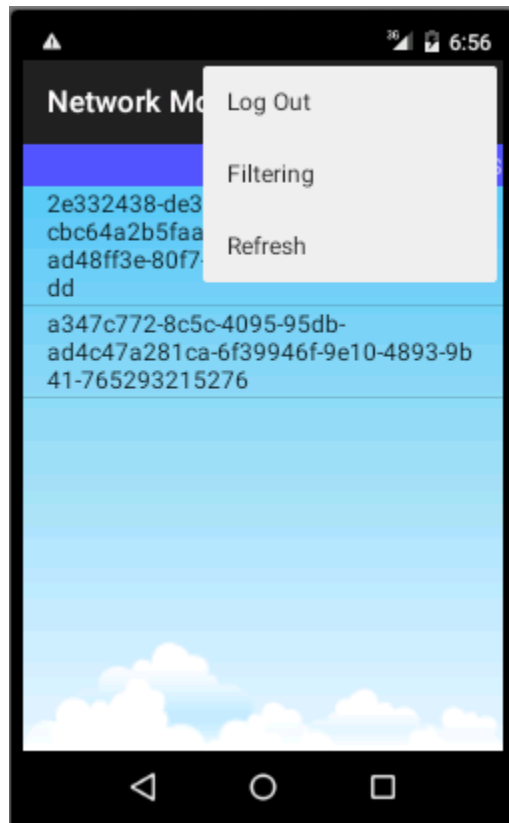
Κεφάλαιο 2

Σενάριο 1^ο

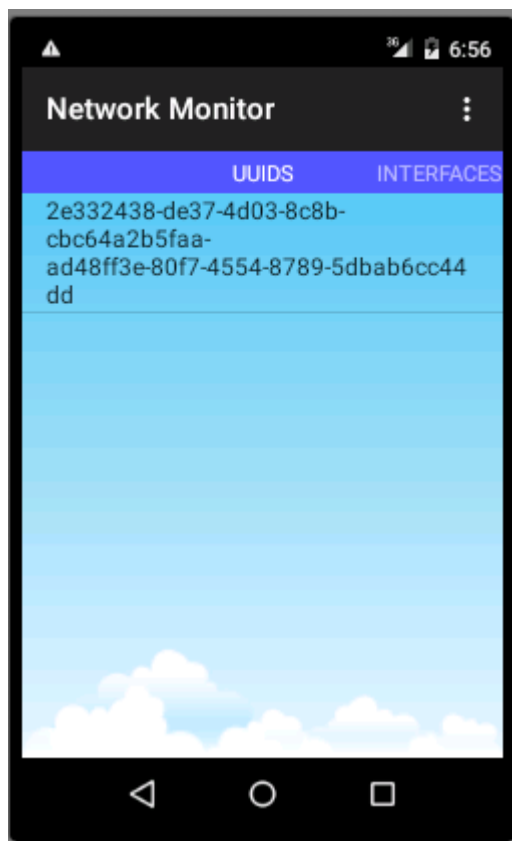
Απλός χρήστης κάνει (επιτυχώς) register και (επιτυχώς) login.



Στην εφαρμογή εμφανίζονται όλα τα διαθέσιμα pc-laptop (τουλάχιστον 3 με τουλάχιστον 2 διαφορετικούς χρήστες).



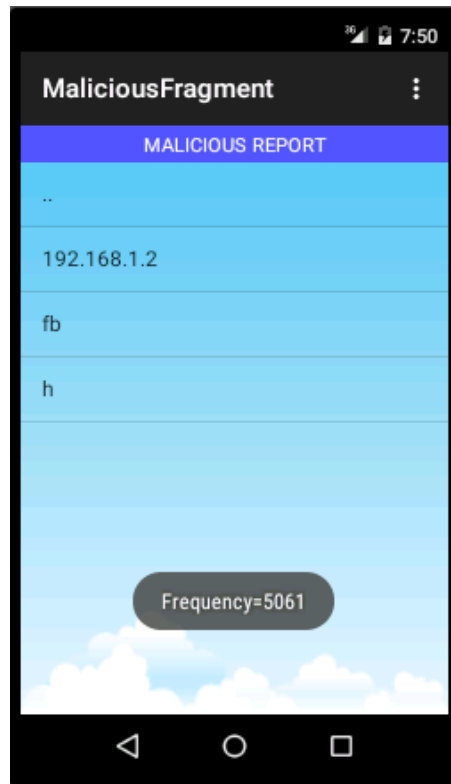
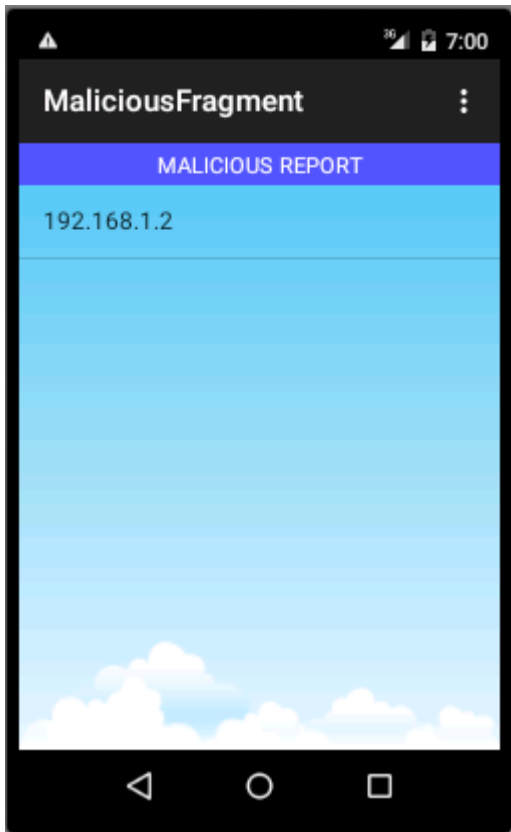
Ο χρήστης επιλέγει να φιλτράρει τα αποτελέσματα και εμφανίζονται μόνο τα δικά του



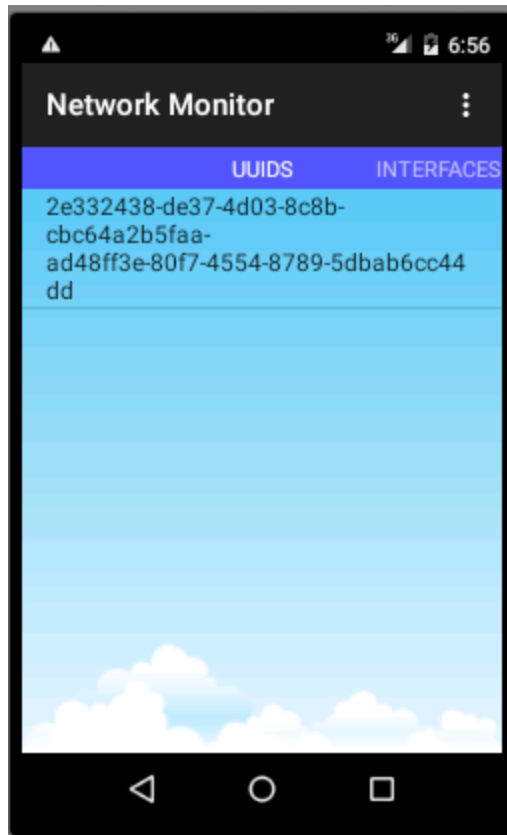
Ο χρήστης επιλέγει ένα από τα pc-laptop και εμφανίζονται στο 2ο tab τα interfaces του pc/laptop που επέλεξε



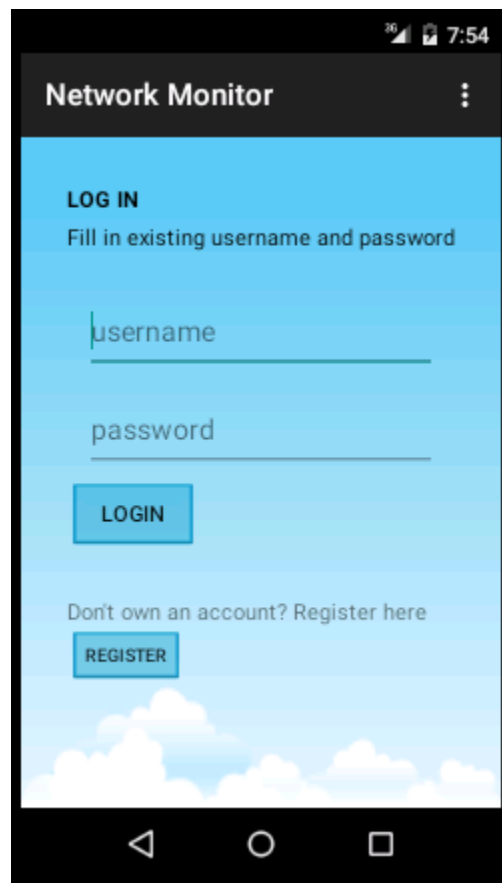
Ο χρήστης επιλέγει ένα από τα interfaces και εμφανίζονται πληροφορίες για το interface.



Ο χρήστης πατάει το κουμπί home και μετά ξανανοίγει την εφαρμογή (screenshot της οθόνης που εμφανίζεται).



Ο χρήστης κάνει logout και ξαναοίγει την εφαρμογή (screenshot της οθόνης που εμφανίζεται).



Σεναριο 2ο

Απλός χρήστης κάνει (ανεπιτυχώς) register

Εμφανίζεται μήνυμα λάθους στο χρήστη

Network Monitor

Enter your personal information

user345

...

.....

Insert your PC or Laptop UUIDs

uuid

INSERT

Network Monitor

user345

...

.....

Insert your PC or Laptop UUIDs

uuid

INSERT

Passwords don't match

REGISTER

Σεναριο 3ο

Admin χρήστης κάνει (επιτυχώς) login

Network Monitor

LOG IN

Fill in existing username and password

admin123

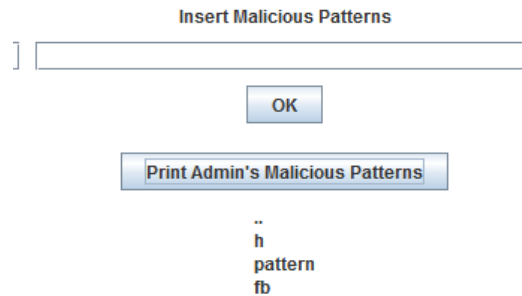
...

LOGIN

Don't own an account? Register here

REGISTER

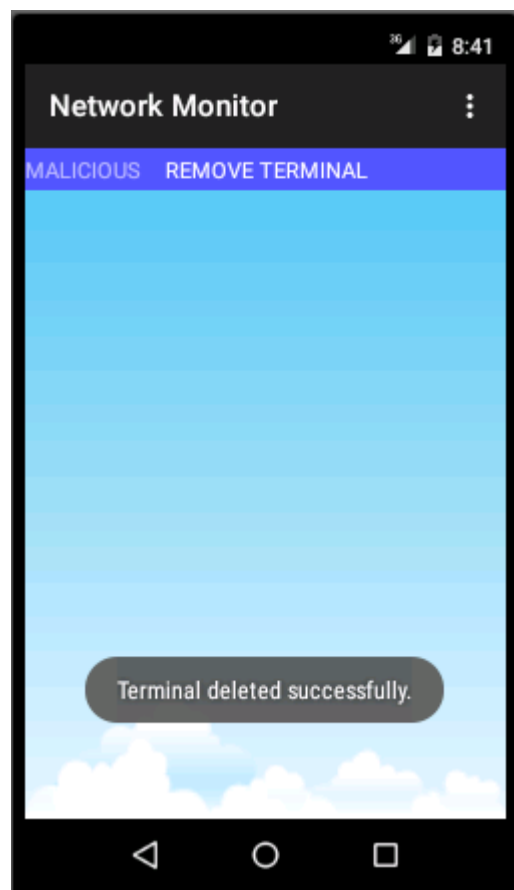
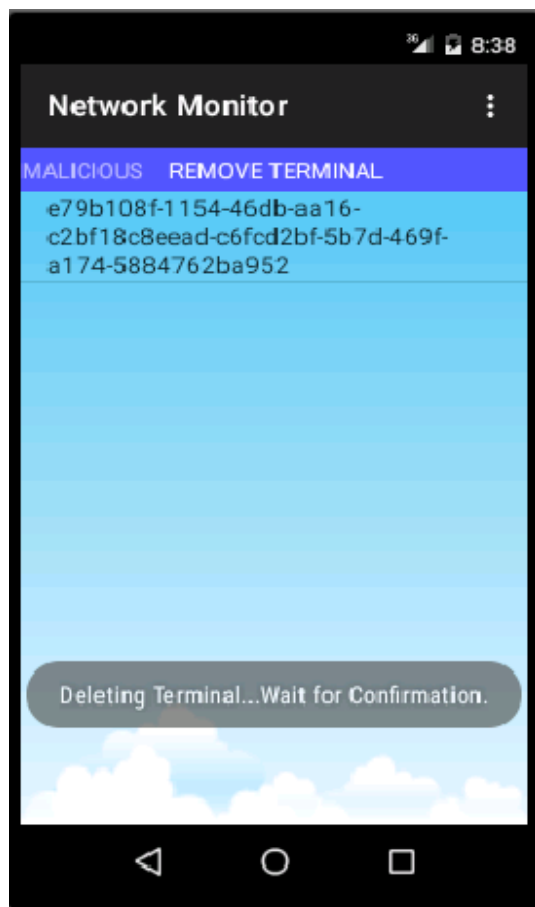
Ο admin μεταβαίνει στο 3ο tab και ορίζει ένα malicious pattern το οποίο δεν υπάρχει στα εμφανιζόμενα patterns, οπότε προστίθεται στον server



Ο admin ορίζει ένα malicious pattern το οποίο υπάρχει στα εμφανιζόμενα patterns, οπότε εμφανίζεται μήνυμα λάθους στον admin



Ο admin μεταβαίνει στο 4ο tab και επιλέγει ένα pc-laptop για διαγραφή. Ο κόμβος διαγράφεται από τον server και εμφανίζεται αντίστοιχο μήνυμα στο pc-laptop που διεγράφη



Enter the server http address:

<http://0.0.0.0:9999/Add/>

```
|-----|
| Starting Network Monitoring Application... |
|-----|
```

|Found new active interfaces...|

|Device\NPF_{3FD59BCA-3A29-475C-8ACE-A2380EB162F2}: Device opened successfully!

|Shutting down application...|

Closing device: \Device\NPF_{3FD59BCA-3A29-475C-8ACE-A2380EB162F2}

```
|-----|
| End |
|-----|
```