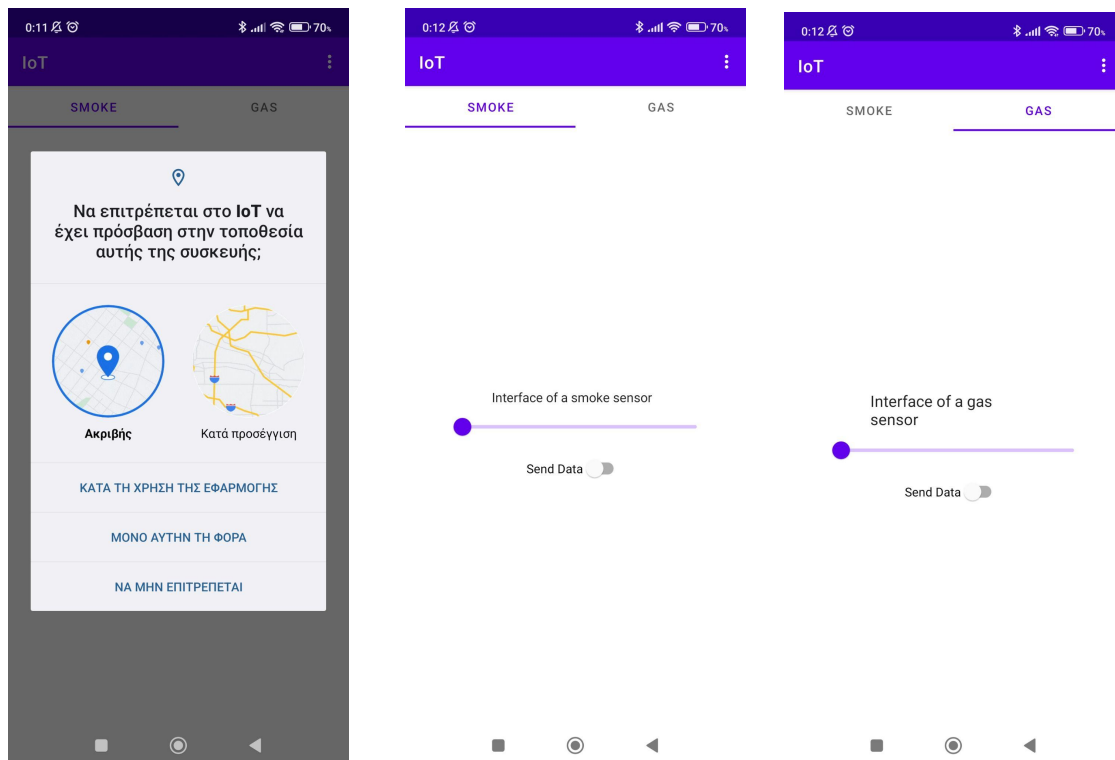


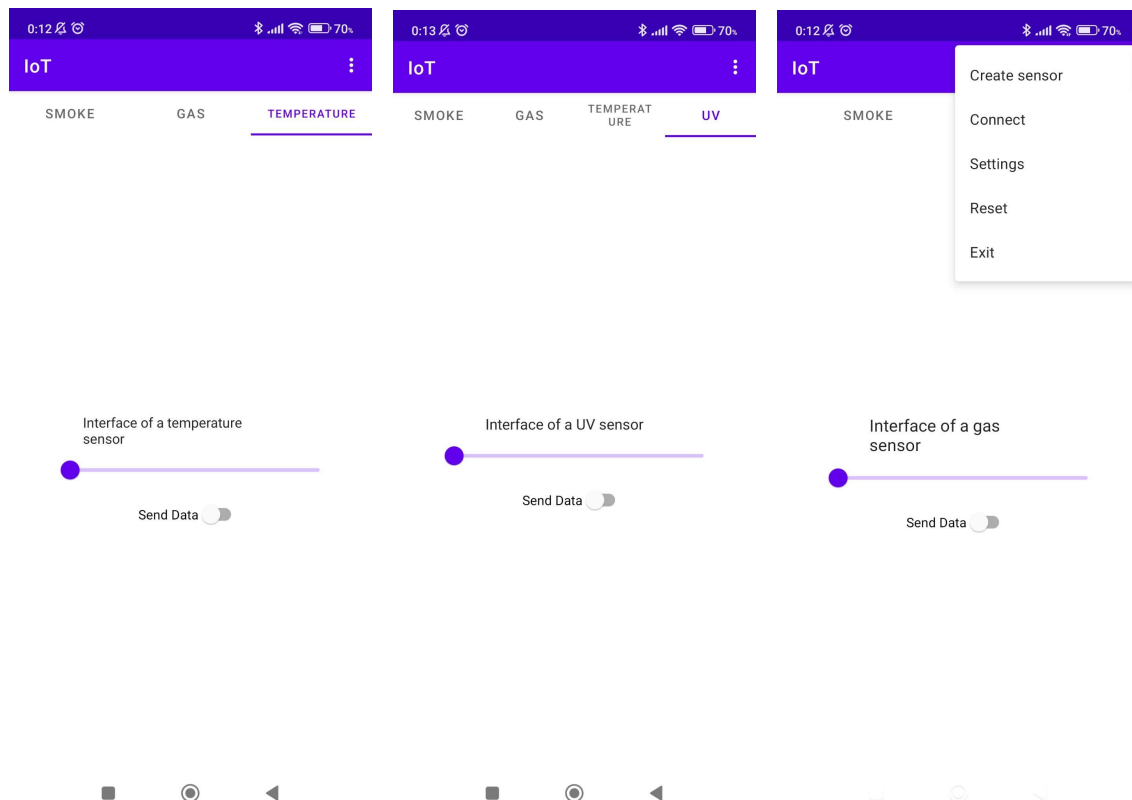
IoT Συσκευές

Για την υλοποίηση της IoT εφαρμογής στο Android Studio έχουν υλοποιηθεί τα αρχεία java: MainActivity.java, SettingsActivity.java, NewSensorActivity.java, IoT.java, ViewPager2Adapter.java, SmokeFragment.java, GasFragment.java, TemperatureFragment.java, UVFragment.java.

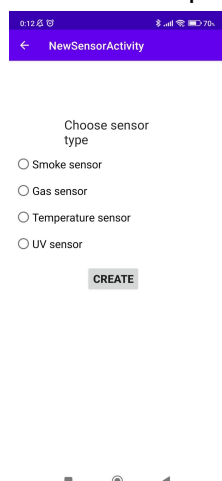
Στο αρχείο MainActivity.java στη συνάρτηση onCreate δημιουργούνται τα tabs των 2 προεγκατεστημένων αισθητήρων καθώς όπως αναφέρεται ρητά στην εκφώνηση της εργασίας οι 2 πρώτοι αισθητήρες (Smoke και Gas) είναι προεγκατεστημένοι στην εφαρμογή. Κάθε ένας αισθητήρας δημιουργείται μέσω της συνάρτησης createTab(string Tabname) που δέχεται ως όρισμα τον τύπο του αισθητήρα. Στη συνάρτηση createTab εκτός από τη δημιουργία του tab του αισθητήρα πραγματοποιείται και η ενδοεπικοινωνία της MainActivity με το fragment του αισθητήρα, ώστε να ενημερώνονται οι τιμές του slider σε κάθε αισθητήρα, καθώς επίσης και αν εκείνος θα στέλνει τα δεδομένα του στον EdgeServer (αυτό επιτυγχάνεται με το SwitchButton send data).



Οι επόμενοι 2 αισθητήρες (Temperature, UV) δημιουργούνται από τον χρήστη και συγκεκριμένα από το menu της εφαρμογής (3 τελείες πάνω δεξιά) και επιλέγοντας "Create Sensor".



Τότε εμφανίζονται 4 radioButtons, ένα για κάθε αισθητήρα. Ο κώδικας που αφορά στο σημείο αυτό έχει υλοποιηθεί στο αρχείο NewSensorActivity.java και η MainActivity ενημερώνεται κατάλληλα για τον τύπο του αισθητήρα που έχει επιλέξει να δημιουργήσει ο χρήστης μέσω ενός Intent, ώστε να δημιουργηθεί στην αρχική οθόνη της εφαρμογής το νέο tab του αισθητήρα. Η εφαρμογή λειτουργεί για κάθε έναν τύπο του αισθητήρα, επομένως αφού έχουν δημιουργηθεί ήδη οι αισθητήρες smoke και gas, ο χρήστης θα πρέπει να δημιουργήσει τον temperature sensor ή/και τον uv sensor. Κάθε αισθητήρας διαθέτει έναν slider στον οποίον το εύρος των τιμών κυμαίνεται από τις δοθείσες τιμές της εκφώνησης. Επίσης, ο καθένας διαθέτει ένα SwitchButton send data, το οποίο όταν είναι ενεργοποιημένο ο αισθητήρας κάνει publish την τιμή του slider στον EdgeServer. Επιπλέον, στον κώδικα κάθε αισθητήρας διαθέτει το δικό του fragment που περιέχει συναρτήσεις get και set, καθώς επίσης γίνονται οι αλλαγές στον Slider και στο SwitchButton για τον καθένα.



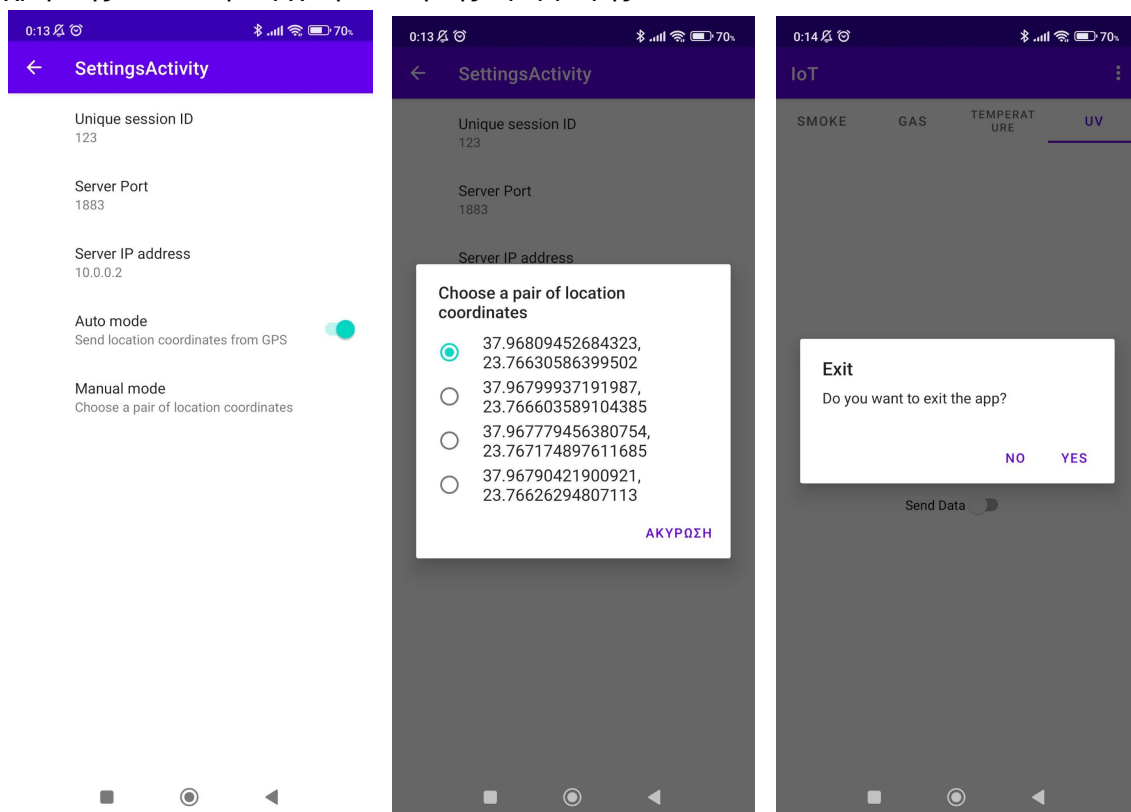
Στο menu της εφαρμογής υπάρχει η επιλογή των settings που μπορεί να αλλάξει ο χρήστης. Συγκεκριμένα, τα settings αυτά αποθηκεύονται στα sharedpreferences ώστε να αποθηκευτούν και μετά το κλείσιμο της εφαρμογής. Για να αποθηκευτούν τα sharedpreferences χρησιμοποιείται ένας sharedpreferences.editor στον οποίο αποθηκεύονται οι τιμές και στη συνέχεια χωρίς ο χρήστης να τις εισάγει ξανά, μπορούν να χρησιμοποιηθούν από την εφαρμογή.

Οι επιλογές για τα settings έχουν δημιουργηθεί στο αρχείο root_preferences.xml.

Αναλυτικότερα, ο χρήστης μπορεί από τα settings να εισάγει το device id, το port number στο οποίο "ακούει" το πρωτόκολλο MQTT, δηλαδή στο port 1883, καθώς επίσης και τη διεύθυνση IP του server (για να τεστάρουμε την εφαρμογή η server IP που χρησιμοποιήθηκε ήταν η 10.0.2.2).

Επίσης, ο χρήστης έχει τη δυνατότητα να επιλέξει εάν οι συντεταγμένες θα εισάγονται από το GPS ή θα είναι 1 από τις 4 δοθείσες από την εκφώνηση (auto mode or manual mode).

Αφού έχουν εισαχθεί οι ρυθμίσεις τότε για να γίνει το connect με τον server ο χρήστης μέσω του menu επιλέγει την παράμετρο Connect που υπάρχει, ώστε να πραγματοποιηθεί το connect με τον edge Server μέσω του πρωτοκόλλου Mqtt. Επίσης, στο menu υπάρχει η επιλογή reset settings και exit (με προειδοποιητικό μήνυμα για έξοδο από την εφαρμογή). Προειδοποιητικό μήνυμα από την εφαρμογή υπάρχει και όταν πατηθεί το back key όταν ο χρήστης είναι στην αρχική οθόνη της εφαρμογής.



Εφόσον γίνει connect στον server, τότε γίνονται publish στον server κάθε 1 second οι συντεταγμένες (latitude, longitude), το ποσοστό της μπαταρίας και οι τιμές των αισθητήρων που έχουν ενεργοποιημένο το SwitchButton send Data. Το publish κάθε 1 second γίνεται μέσω της συνάρτησης onResume στην οποία υπάρχει ένας handler και δημιουργείται

ένα Runnable(). Μέσα στη συνάρτηση αυτή καλείται η συνάρτηση publish εφόσον υπάρχει σύνδεση στον server (όταν η boolean μεταβλητή connect_status είναι true). Η συνάρτηση publish() εισάγει σε ένα JSONObject τις συντεταγμένες και το ποσοστό της μπαταρίας (publishObject), σε ένα JSONObject για κάθε αισθητήρα που έχει ενεργοποιημένο το SwitchButton send data, την τιμή του slider και τον τύπο του αισθητήρα. Τέλος, κάνει publish τις πληροφορίες αυτές στον edgeServer μέσω του topic `iot`.