

# Progetto 3 – IoT

## Analisi

Nella prima fase del progetto abbiamo cercato di focalizzarci e analizzare tutti gli attori facenti parte del sistema. Quindi descrivendo le funzionalità di questi e suddividendoli come moduli definendo le modalità di interazione tra loro. Questo lavoro iniziale ci ha permesso di dividerci efficacemente i compiti dei sottosistemi, poiché questi indipendenti tra loro.

## Design

Nella progettazione delle singole soluzioni per ogni sottosistema ci siamo basati sulle specifiche richieste dal progetto dove specificato, e sviluppato delle nostre dove no.

Infatti, nel sistema *Controller* è stata adottata la logica in termini di macchina a stati finiti sincrona che si basa su un task principale che si occupa di passare da uno stato all'altro in base alle comunicazioni BT con il sistema *Mobile App*.

In quest'ultimo ci siamo dedicati nello sviluppare una soluzione che potesse rispecchiare la realtà del sistema nella modalità più fedele, e quindi senza l'utilizzo di emulazioni lato seriale o BT.

Per quanto riguarda lo sviluppo del sistema *Service* ci siamo avvalsi dell'utilizzo di un server Java che potesse intercettare richieste POST e GET mantenendo lo stato del sistema a runtime. Questo è stato possibile utilizzando il servizio offerto dal sito *ngrok* il quale permette di reindirizzare su un calcolatore nascosto da un NAT le richieste provenienti da Internet.

Gli ultimi due sottosistemi *Edge* e *DashBoard* si basano rispettivamente su una macchina a stati finiti semplice e tecnologie web client-site.

## Sviluppo

Le scelte implementative più rilevanti che attuate ci hanno portate alla soluzione preposta ed eventuali complicità sono definite di seguito.

### Aggiornamento UI e Interazione BT/Http SD-App

Nel sottosistema *Mobile App*, in cui l'architettura ad eventi è preponderante, è stata fatta particolare attenzione nell'aspetto riguardante l'interazione con l'utilizzatore. Infatti, il sottosistema in questione è composto di una Activity principale ed una di supporto. Quest'ultima prende spazio al momento della selezione del tipo di rifiuto che si intende depositare e permette di gestire la funzione prettamente di selezione per il deposito, l'annullamento e l'estensione del tempo, come anche la sua visualizzazione del conto alla rovescia. L'interfacciamento utente è sviluppato in modo tale da richiedere le connessioni e autorizzazioni necessarie prima di essere abilitati ad effettuare operazioni di deposito. In particolare, in prima battuta il dispositivo deve essere già associato con il modulo BT del sottosistema *Controller* così da poterne poi concedere la connessione e inoltre richiedere un token virtuale che autorizza il dispositivo ad effettuare il deposito. Quest'ultimo viene richiesto per ogni successivo

deposito visto che il sottosistema *Service* potrebbe diventare non-disponibile tra un deposito e l'altro.

Nel caso la comunicazione di deposito venga interrotta o avvenisse in modo errato (tra la *Mobile App* e il *Controller*) lo scadere di un timer di controllo, che inizia dal momento dell'invio richiesta deposito al *Controller*, reimposterà la UI della applicazione permettendo all'utente di procedere con un altro tentativo.

Infine, la comunicazione http con il sottosistema *Service* avviene solamente dopo espressa conferma di esecuzione del comando.

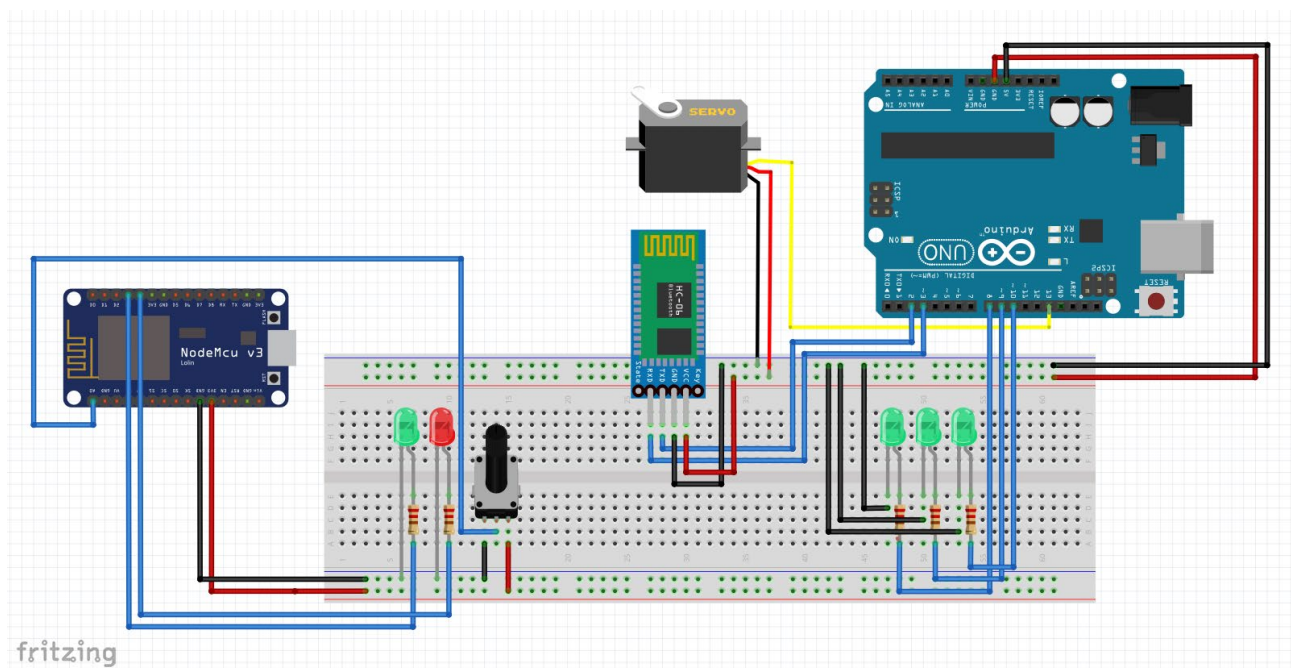
### Gestione comandi BT SD-Controller

Nel sottosistema del *Controller* è stata fatta particolare attenzione nello stato "TIMING" del task principale in cui vengono eseguiti i controlli sui comandi impartiti per mezzo BT. In particolare, si è strutturata una classe di interfacciamento (GUI) con l'interfaccia BT che permette di controllare un messaggio ricevuto in parti diversi di codice e consumarlo non rendendolo più disponibile una volta riconosciuto.

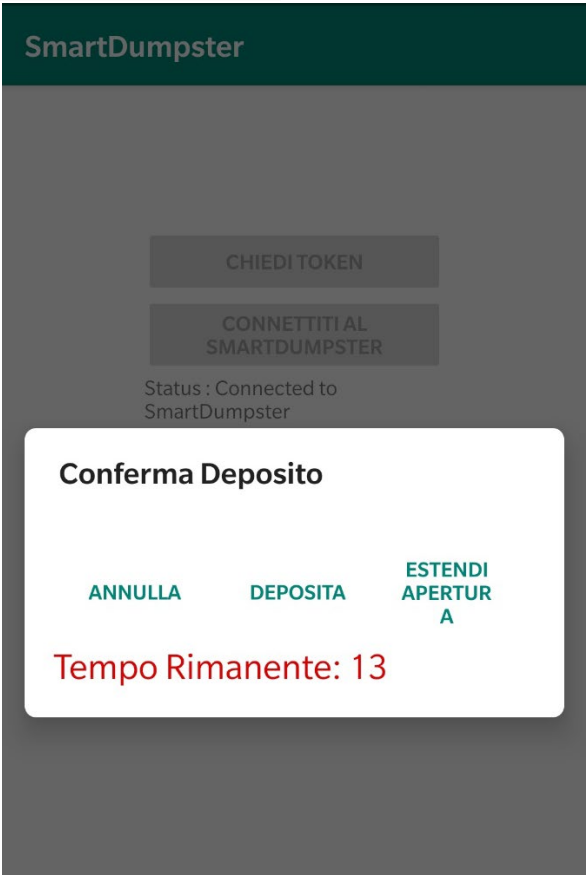
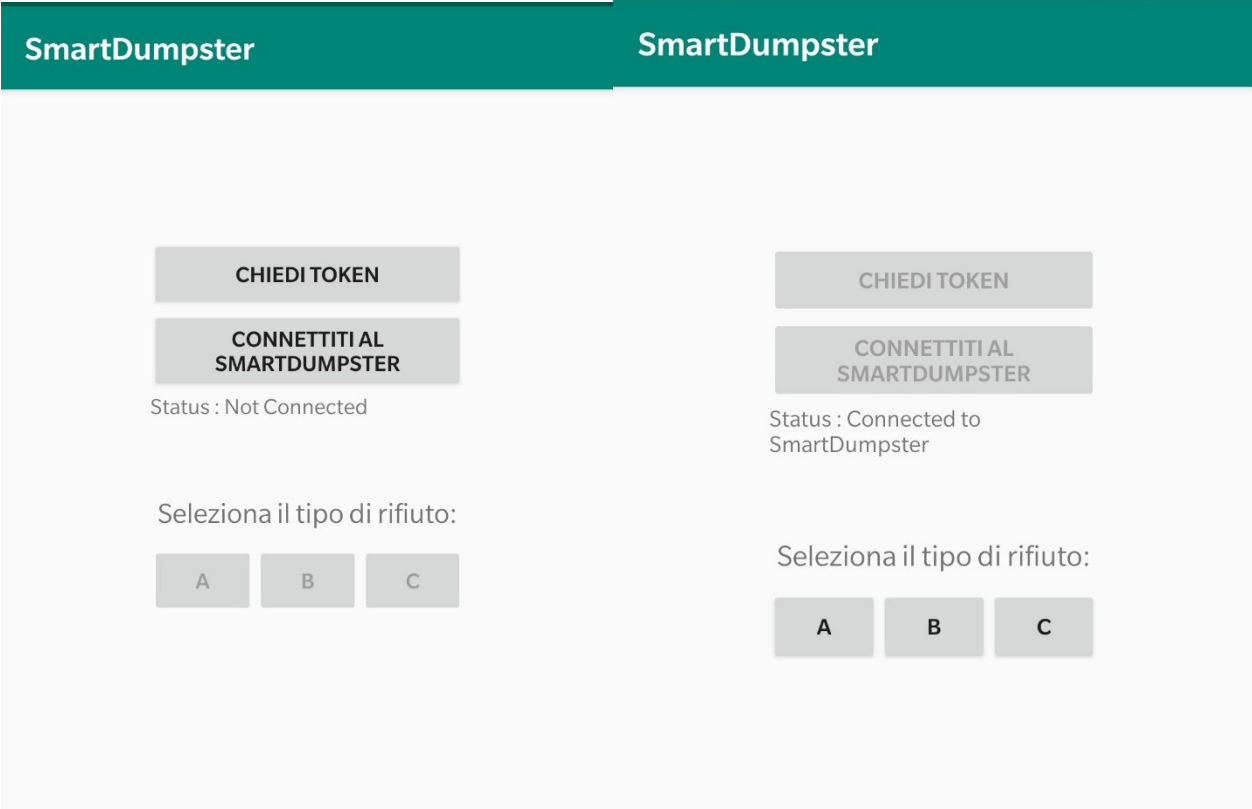
### Comunicazioni tra sottosistemi

Il principale mezzo trasmissivo tra i vari sottosistemi è via Internet, questo può essere un vantaggio visto che appunto non rende obbligatoria la presenza di tutte le componenti nello stesso luogo e momento. L'unico svantaggio che limita la soluzione finale sta nel fatto che il servizio non ci riserva staticamente un indirizzo URL, bensì questo cambia ogni volta che viene riavviato, il che costringe la modifica delle stringhe hard-coded nel sottosistema *Edge*, *Mobile App*, *DashBoard*.

## Schema Fritzing



# Screen App Mobile



## Machine a Stati Finiti SD-Controller

