1. **Utilizzeremo un server centrale**: si era valutata un’architettura con server locali ed un server centrale, ma si sarebbero create diverse problematiche sull’aggiornamento e la comunicazione dei dati raccolti.   
   Abbiamo quindi optato per la gestione del sistema con un singolo server centrale, questo ci permetterà di evitare problematiche di aggiornamento dei dati. La cosa però più importante è quello di rendere scalabile il sistema.
2. **Realizzeremo una webApp:** si renderà quindi possibile l’accesso da un qualunque dispositivo, verrà implementata utilizzando il framework Ionic. Il client ha quindi la necessità di essere molto leggero, quindi la struttura del sistema e la gestione dei dati deve favorire questa necessità.
3. **Utilizzeremo un Database Relazionale.** Nei precedenti documenti avevamo descritto la decisione di utilizzare una base di dati non relazionale, analizzando in maniera più approfondita la struttura dei dati che abbiamo progettato non si riscontrata la necessità di dover utilizzare un dabatase non relazionale. Inoltre in fase di sviluppo prototipo, dove continuavamo a pensare ad un DB non relazionale, abbiamo riscontrato problematiche nell’aggiornamento dei dati, i tempi di aggiornamento erano eccessivi.
4. **Forniremo feedback ai sensori sul loro stato di anomalia (regolazione della frequenza di invio).** Ogni sensore ha un parametro di tempo, questo parametro è normalmente impostato per inviare il dato calcolato minuto per minuto. In caso di dati non congruenti o dati che dimostrano la presenza di problematiche verrà modificata questa frequenza di invio.
5. **Il dato del singolo sensore non verrà visualizzato da tutti i livelli di gestione,** l’unico gestore che potrà visualizzare il dato sarà il gestore di edificio, quindi ai gestori dei livelli superiori (distretto e città) verrà semplicemente comunicato che nel livello immediatamente sottostante vi è un errore/anomalia.Dopo lunghe discussioni si è ragionevolmente arrivati alla conclusione che fornire i dati di tutti i sensori presenti nel sistema a tutti i gestori di qualsiasi livello avrebbe appesantito notevolmente il sistema, creando, inoltre, problemi di usabilità all’utente finale; in più sarebbe stato superfluo far ricevere i dati di un determinato distretto alla dashboard del gestore di un distretto diverso da quello da lui gestito.