



Scenario

Si progetti e implementi l'architettura software di un sistema di waste management di una smart city. I cassonetti smart – simulati da un nodo software – permettono il conferimento di rifiuti di vario tipo previa identificazione del cittadino. Il nodo comunica in tempo reale i dati del conferimento includendo varie informazioni fra cui ad esempio identità del cittadino, tipo di rifiuto, data, ora e luogo.

Lato backend, i sistemi interessati ai dati comunicati dai cassonetti sono quelli dell'azienda di gestione rifiuti, e l'ufficio del comune che calcola le tasse.

L'azienda di gestione rifiuti può:

- Ricevere allarmi di sovrabbondanza di conferimenti
- Monitorare su mappa lo stato dei cassonetti
- Istanziare percorsi di pulizia dei cassonetti

L'ufficio del comune può:

- Ogni anno, emettere l'importo da pagare per ogni cittadino
- Identificare cittadini da sensibilizzare
- Controllare lo stato dei pagamenti
- Analizzare le performance di separazione dei rifiuti del Comune

Il cittadino può:

- Controllare i suoi pagamenti ed eventualmente pagare
- Analizzare le proprie performance di separazione dei rifiuti

Insieme alla documentazione tecnica (architettura software, modello concettuale del database), al manuale dei casi d'uso e agli sprint backlog e burndown chart, deve essere consegnato un report contenente i design pattern adottati (GoF, microservizi), gli indici di complessità ciclomatica, e le metriche di sostenibilità dell'architettura secondo le specifiche SCI.

Contenuto del database

Il database deve contenere almeno le seguenti entità:

- 1 amministratore
- 20 cassonetti
- 3 cittadini

Raccomandazioni

Il Sistema deve essere implementato mediante framework Spring Boot per il backend, e Angular per il front-end. Il sistema finale deve essere:

- scalabile rispetto a numero di cassonetti smart e cittadini
- deployato su Cloud AWS o altro di preferenza dello studente
- dockerizzato
- versionato su server Git

Le specifiche architetturali devono essere preferibilmente definite con tecniche di Infrastructure as Code.

Gli studenti devono adottare il processo Scrum e documentarne l'applicazione durante il progetto. Il progetto deve essere sviluppato in coppia (due studenti è il numero massimo consentito). Per esigenze individuali si prega di contattare il docente.

48 ore prima dell'esame orale, gli studenti devono inviare al docente la documentazione del progetto in formato PDF e il link al repository Git. Il sorgente del progetto consegnato verrà analizzato automaticamente utilizzando DIFF o strumenti simili.

Il giorno dell'esame orale gli studenti: (1) presenteranno l'architettura software del sistema, (2) presenteranno i design pattern adottati, (3) presenteranno gli unit test (insieme alle metriche di copertura e complessità), (4) dimostreranno in modo interattivo il progetto, (5) dimostreranno la capacità di evolvere l'architettura software partendo da nuovi requisiti forniti dal docente. La presentazione verrà eseguita direttamente sul sistema in esecuzione, senza supporti di presentazione statici (ad esempio, PowerPoint).

La documentazione deve essere breve e composta da: (1) analisi dei casi d'uso, (2) progettazione UML dell'architettura software con dettagli sui design pattern adottati, (3) progettazione del database (E/R oppure non relazionale), (4) descrizione degli unit tests con report dettagliato sulle metriche, (5) Scrum "sprint backlog" e "grafico di burndown", (6) sezione sulle metriche di sostenibilità dell'architettura software.