

# **Esami di stato per l'abilitazione all'esercizio della professione di Ingegnere II Sessione 2014**

## **Sezione A Prova progettuale – Settore Informatica**

*Al candidato è richiesto di svolgere uno dei seguenti temi proposti.*

### **TEMA N. 1**

Si progetti un sistema informativo per la gestione di un centro di vaccinazioni. Ogni vaccino è identificato da un codice, ed ha associato un numero intero che rappresenta il livello di rischio della sua somministrazione e l'anno in cui è stato introdotto nel sistema sanitario. Ogni vaccino viene somministrato per prevenire una o più malattie, ed è di interesse conoscere tali malattie. Per ogni vaccino è di interesse conoscere la sua importanza data da una funzione che calcola con la media del numero di persone del centro che hanno contratto le corrispondenti malattie. Di ogni malattia interessa il codice identificativo ed il tipo (contagiosa, ereditaria, ecc.), mentre non è interessante risalire agli eventuali vaccini che la prevengono. Per una malattia può esistere un numero qualunque di vaccini che la prevengono. Una vaccinazione rappresenta la somministrazione di un vaccino ad una persona (una persona viene sottoposta al massimo ad una somministrazione di ogni vaccino). Di ogni vaccinazione interessa la data in cui è avvenuta. Di ogni persona interessa il nome, il cognome, la data di nascita ed altre informazioni utili (un recapito telefonico, indirizzo di residenza, ecc..) ed è di interesse risalire alla sue vaccinazioni. Dato un vaccino è di interesse anche risalire alle vaccinazioni che hanno avuto per oggetto quel vaccino. Infine, di ogni malattia interessano le persone che le hanno contratte (mentre non è di interesse risalire dalle persone alle malattie che hanno contratto).

I servizi che il sistema deve consentire sono le normali operazioni gestionali di inserimento/cancellazione/modifica di dati su vaccini, persone, malattie e vaccinazioni, e di ispezione (ricerca) dei valori. Tali operazioni devono poter essere eseguite in modalità desktop dagli operatori del centro vaccinazioni. Nell'accezione più ampia, alcune informazioni (dati personali, vaccinazioni effettuate, ecc..) devono poter essere visualizzate anche dalle persone oggetto delle vaccinazioni, quindi viene definita un'interfaccia tipo web service per accedervi.

Si svolgano i seguenti punti.

- Progettare l'architettura del sistema tramite la definizione di un diagramma delle componenti UML o un diagramma a blocchi facendo tutte le ulteriori ipotesi necessarie per la modellizzazione delle varie entità/dispositivi coinvolti. Fornire se si ritiene utile anche altri

diagrammi in UML o notazioni simili (macchine di stato, reti di Petri, ecc.) per modellare il comportamento di una o più componenti critiche del sistema e le loro interazioni.

- Si effettui la progettazione concettuale dei dati che il sistema deve gestire mediante un modello Entità-Relazione o mediante un diagramma delle classi UML.
- Si definisca lo schema logico relazionale del database di supporto del sistema, usando il linguaggio SQL per definire le tabelle e i vincoli. Scrivere, in particolare, le seguenti interrogazioni in algebra relazionale e/o SQL:
  - a. Dato un vaccino, trovare le persone alle quali è stato somministrato in un certo periodo (intervallo tra due date).
  - b. Data una malattia, contare le persone che l'hanno contratta in un certo periodo (intervallo tra due date).
  - c. Data una persona, trovare le vaccinazioni che ha effettuato.
  - d. Dati come parametri una data  $d$ , un numero intero  $n$ , ed un insieme  $v$  di vaccini, trovare le persone alle quali è stato somministrato prima di  $d$  una vaccinazione di un vaccino presente in  $v$  la cui importanza è maggiore di  $n$ .
- Si definiscano i messaggi essenziali che devono essere scambiati tra i vari client e server (si definiscano sia i messaggi di richiesta che quelli di risposta) per l'esecuzione di funzioni essenziali di gestione da parte degli operatori del centro e di visualizzazione di un certo numero di informazioni da parte dell'utenza finale. Usare a tale scopo, un IDL (Interface Description Language) di propria scelta, ad esempio: RSDL (RESTful Service Description Language) per REST web services, WSDL, XML DTD o schema, Java interface, UML interface e sequence diagram, ecc.
- Scrivere il codice Java-like o pseudocodice del lato server che gestisce i gruppi di messaggi (interagendo con il database per prelevare/aggiornare i dati) per l'esecuzione di alcuni servizi essenziali.

## TEMA N. 2

Si consideri un'applicazione web (portale web) pre-esistente responsabile dell'erogazione di servizi di *e-government* a utenti finali della quale non si conoscono i dettagli interni di funzionamento. Si consideri di estendere tale applicazione con funzionalità di controllo d'accesso progettando un opportuno sistema *Single Sign-On* (SSO, traducibile come autenticazione unica o identificazione unica) con funzionalità di *autenticazione e autorizzazione* degli utenti. L'obiettivo è discriminare l'accesso ai singoli servizi applicativi forniti dal portale, mantenendo al tempo stesso trasparenza e non invasività rispetto ai servizi pre-esistenti. Il portale potrà pertanto erogare i propri servizi soltanto a utenti registrati. La nuova infrastruttura di controllo è responsabile del mantenimento del profilo di registrazione degli utenti e dell'erogazione di un apposito servizio di registrazione e di autenticazione. Deve, inoltre, essere in grado di intercettare le richieste di servizio effettuate dall'utente durante la navigazione del portale, autorizzando o meno, l'utente all'accesso e successivamente trasferire il controllo al servizio applicativo.

Un utente registrato è caratterizzato da un identificativo (username) che lo identifica univocamente nel sistema, da un insieme di informazioni anagrafiche (nome, cognome, ecc.) e da credenziali opportune utilizzate dall'utente durante la fase di autenticazione a seconda dello specifico servizio richiesto. In particolare, si distingue tra credenziali "deboli" e "forti" a seconda della garanzia di identificazione del soggetto associata alla specifica credenziale utilizzata. Per le credenziali di tipo "forte" è importante prevedere un processo di consegna riservato e sicuro all'utente.

Un utente è anche caratterizzato da un insieme di qualifiche (o ruoli) noti al portale erogatore dei servizi. Ogni servizio erogato dal portale è caratterizzato da un insieme di qualifiche autorizzate all'accesso del servizio stesso. Una volta completata con successo l'autenticazione, il portale erogatore di servizi deve poter consentire all'utente richiedente l'accesso al servizio richiesto soltanto se l'utente è in possesso di almeno una delle qualifiche associate al servizio.

Il sistema da realizzare è dunque responsabile della gestione delle nuove funzionalità di registrazione degli utenti, di autenticazione degli utenti e di accreditamento (per associare ad un utente registrato le qualifiche necessarie per l'accesso ai servizi erogati dal portale).

Si consideri la comunicazione client-server che tali nuove funzionalità comportano e si svolgano i seguenti punti:

- Si effettui la progettazione concettuale dei nuovi dati da gestire (quelli relativi agli utenti, credenziali e qualifiche) mediante un modello Entità-Relazione o mediante un diagramma delle classi UML.
- Si definisca una macro-architettura software del sistema mediante diagramma a blocchi e/o diagramma UML delle componenti, individuando in particolare le componenti e/o sottosistemi per l'intercettazione e controllo delle richieste di servizio, per l'accesso al profilo di autenticazione e autorizzazione di un servizio.

- Si definisca, poi, uno scenario di accesso ad un servizio del portale da parte di un utente registrato che mostri il comportamento (esecuzione normale) delle componenti del sistema individuate al punto precedente per il controllo degli accessi.
- Per i servizi principali, si definiscano i messaggi che devono essere scambiati tra il client e i server (si definiscano sia i messaggi di richiesta che quelli di risposta) usando un IDL (Interface Description Language) di propria scelta, ad esempio: RSDL (RESTful Service Description Language) per REST web services, WSDL, XML DTD o schema, Java interface, UML interface e diagramma di sequenza UML, ecc.
- Scrivere il codice Java-like o pseudocodice del lato server che gestisce i gruppi di messaggi (interagendo con il database per prelevare/aggiornare i dati) scambiati tra il client e i server per accedere ad un servizio del portale.