

Esame Es.20240403 – Prova scritta del 03 aprile 2024

Si vuole progettare e realizzare *TimeBank*, un sistema che fornisca i servizi di una "banca del tempo". Tale sistema sarà utilizzato tramite client quali applicazioni per smartphone e browser web.

Gli utenti di *TimeBank* possono prendere in prestito ed offrire ore del loro tempo libero ad altri utenti, per soddisfare i loro bisogni reciproci. Lo scopo è quello di supportare forme di scambio e aiuto reciproco instaurando una rete di relazioni sociali.

In breve, tutti gli utenti dichiarano l'intenzione di volersi mettere a disposizione per un certo insieme di prestazioni (ad es., "riparazioni domestiche", "corsi di inglese", ecc.). Queste vengono chiamate le *abilità* messe a disposizione dai singoli utenti. Tali abilità possono essere richieste dagli altri utenti per risolvere delle loro necessità.

Ad esempio, l'utente Alice (esperta di corsi di inglese e babysitting), lamenta una perdita d'acqua in casa; utilizzando *TimeBank*, si mette in contatto con l'utente Biagio –che sa effettuare riparazioni domestiche. I due utenti si accordano per una prestazione della durata di 2 ore in un certo giorno.

Queste interazioni tra utenti vengono chiamate "scambi di tempo". Al momento dello scambio di tempo, Alice (l'utente richiedente lo scambio) "pagherà" Biagio (utente offerente) un importo per il servizio richiesto. Pertanto il "saldo-tempo" di Alice subirà un addebito pari al valore dello scambio, mentre quello di Biagio subirà un accredito dello stesso importo. D'altro canto, Alice potrà aumentare il proprio "saldo-tempo" offrendo ore per insegnare l'inglese agli altri utenti di *TimeBank*, ricevendo da questi i rispettivi pagamenti.

Si osservi che il valore di uno scambio di tempo si basa sulla sua durata, e non sulla tipologia del servizio offerto. Infine, si noti che *TimeBank* non prevede, in nessun caso, scambio di denaro tra gli utenti.

Degli utenti dell'applicazione sono di interesse il nome, il cognome, l'indirizzo email e il numero di telefono. Inoltre, per ognuno interessa conoscere le abilità messe a disposizione (di cui interessa il nome), e gli scambi di tempo che intercorrono tra *coppie* di utenti (non è prevista la possibilità di scambi di tempo che coinvolgano più di due utenti). Degli scambi interessa la data e l'ora di inizio, la durata (in ore), e la particolare abilità richiesta all'utente offerente.

TimeBank deve permettere ad ogni utente che richieda una prestazione di esprimere un giudizio (*feedback*) da 1 a 10 sulla qualità del servizio ricevuto.

Il valore del tempo scambiato non è sempre lo stesso, ma dipende dalla qualità dei feedback ricevuti dall'utente offerente. In particolare, deve essere possibile calcolare, per ogni utente e per ogni sua abilità, il suo *moltiplicatore*; il valore di uno scambio sarà quindi dato dal prodotto della sua durata e del moltiplicatore associato all'utente rispetto all'abilità oggetto dello scambio.

Più precisamente, sia d la durata dello scambio, t l'istante in cui avviene lo scambio, u l'utente offerente, a l'abilità richiesta, m la media dei feedback ottenuti da u per tutti gli scambi avvenuti prima di t in cui ha offerto la propria abilità a , il valore dello scambio è calcolato come:

$$v(u, a, t, d) = d(1 + \arctan(0.05 * m)).$$

Al momento di inizio di uno scambio di tempo, il saldo-tempo dell'utente offerente viene accreditato di un importo pari al valore dello scambio, mentre quello dell'utente richiedente viene addebitato dello stesso importo.

In ogni momento, per gli utenti di *TimeBank* si deve poter calcolare il saldo-tempo, che è pari alla somma dei valori degli scambi offerti meno i valori di quelli richiesti.

Si noti che, in ogni momento, il saldo di tutti gli utenti deve essere maggiore o uguale a zero (cioè gli utenti non possono andare in "rosso"). Pertanto, un utente non può richiedere uno scambio di tempo se si prevede che, al momento di inizio di tale scambio, non abbia credito sufficiente a pagarlo. Tuttavia, al fine di garantire il buon funzionamento del sistema nelle sue fasi iniziali, tutti i correntisti ricevono, al momento della loro iscrizione, un accredito bonus di un determinato numero x di ore (pertanto il saldo-tempo di un utente, al momento dell'iscrizione, è pari a x).

Il sistema deve offrire le seguenti funzionalità agli utenti:

1. registrarsi con un accredito iniziale di x ore bonus;
2. trovare gli utenti con una certa abilità e con una media di feedback maggiore o uguale a un valore dato;
3. calcolare il proprio moltiplicatore corrente rispetto a una data abilità
4. ottenere le abilità di un dato utente;
5. modificare le proprie abilità;
6. dare un feedback per uno scambio in cui ha partecipato come richiedente;
7. calcolare il proprio saldo corrente.

1 Analisi concettuale

Domanda 1 (10 minuti) Raffinare la specifica dei requisiti eliminando inconsistenze, omissioni e ridondanze e producendo un elenco numerato di requisiti il meno ambiguo possibile. (La risposta a questa domanda non sarà valutata, ma si consiglia di svolgere accuratamente questo passo, in quanto può facilitare di molto le attività di progetto.)

Risposta

Utente:

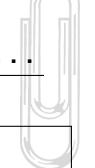
- nome: String
- cognome: String
- mail: Email
- tel: Tel
- ist^{gsc}: Database
- associato alle Abilità

Somlio:

- inizioSca: DataOra
- durataOre: Reale > 0
- associato agli Utenti che dormono e mancano
- associato alle Abilità
- associato all'Utente che tiene traccia Feedback:
 - range: 1..10

Abilità:

- nome: String
- associato agli Utenti
 - (con media (t : Data): Reale ≥ 0)



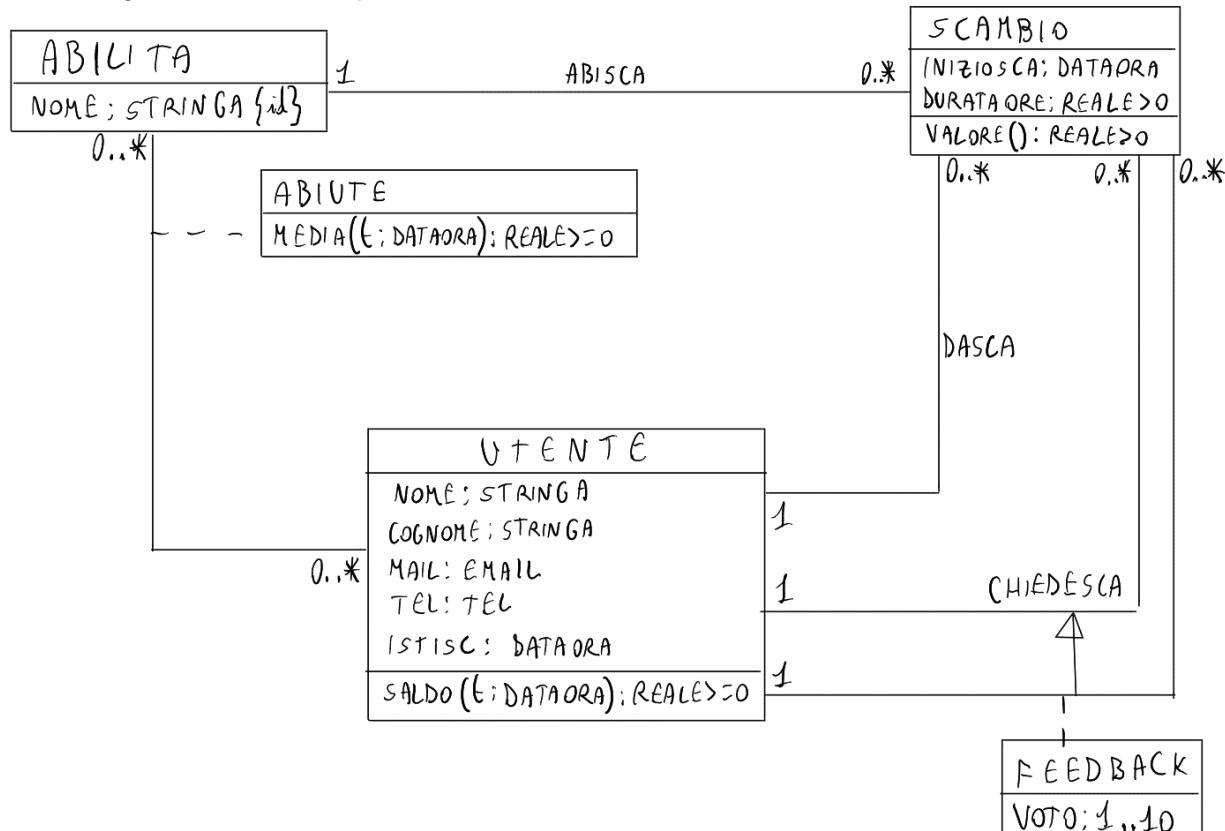
Risposta alla Domanda 1 (segue)

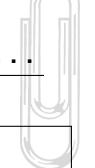
Domanda 2 (45 minuti; 75 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML concettuale delle classi per l'applicazione, le specifiche di classi, associazioni, tipi di dato e vincoli esterni.

Una risposta soddisfacente a questa domanda è condizione *necessaria* (ma non sufficiente) per superare la prova.

Diagramma UML concettuale delle classi

Produrre un diagramma UML concettuale delle classi per l'applicazione in termini di classi, associazioni, attributi, generalizzazioni, operazioni di classe.





Risposta alla Domanda 2 (segue)

Specifiche delle classi o associazioni Per ogni classe o associazione del diagramma **con** operazioni o vincoli:

- Definire la specifica formale di eventuali operazioni necessarie a modellare i requisiti contrassegnati dalla barra laterale, ed eventuali vincoli esterni. Usare la logica del primo ordine estesa con teoria degli insiemi e semantica di mondo reale vista nel corso, usando il seguente alfabeto:
 - Un simbolo di predicato $C/1$ per ogni classe C .
Semantica di $C(x)$: x è una istanza di C .
 - Un simbolo di predicato $T/1$ per ogni tipo di dato T .
Semantica di $T(x)$: x è un valore di T .
 - Un simbolo di predicato $\text{assoc}/2$ per ogni associazione binaria assoc.
Semantica di $\text{assoc}(c_1, c_2)$: (c_1, c_2) è una istanza di assoc.
 - Un simbolo di predicato $\text{attr}/2$ per ogni attributo attr di entità
Semantica di $\text{attr}(c, v)$: uno dei valori dell'attributo attr dell'istanza c è v .
 - Un simbolo di predicato $\text{attr}/3$ per ogni attributo attr di associazione binaria.
Semantica di $\text{attr}(c_1, c_2, v)$: uno dei valori dell'attr. attr del link (c_1, c_2) è v .
 - Un simbolo di predicato $\text{op}/(n+2)$ per ogni operazione di classe ad n argomenti.
Semantica di $\text{op}(c, \arg_1, \dots, \arg_n, v)$: uno dei valori di ritorno di op, quando invocata sull'istanza c e con argomenti \arg_1, \dots, \arg_n è v .
 - Il simbolo di $=/2$ (la cui interpretazione è la relazione che lega ogni elemento del dominio di interpretazione solo con se stesso) e opportuni simboli di predicato e di funzione, soggetti a semantica di modo reale, per relazioni e funzioni standard tra elementi dei tipi di dato, tra cui $\text{adesso}/0$, interpretato come il valore del dominio DataOra che rappresenta l'istante corrente.

Risposta

<p><input type="checkbox"/> Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ...<u>V.TENTE</u>.....</p> <p>Operazioni, vincoli:</p> <p><u>V. NO SALDO NEGATIVO</u></p> <p>$\forall u, sc, v, sa, i [CHIedesca(u, sc) \wedge VALORE(sc, v) \wedge SALDO(u, i, sa) \wedge INIZIOSCA(sc, i) \rightarrow sa >= v]$</p>	<p><input type="checkbox"/> Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ...<u>V.TENTE</u>.....</p> <p>Operazioni, vincoli:</p> <p><u>V. SCAMBI DISGIUNTI</u></p> <p>$\forall u, sc, i, d [CHIedesca(u, sc) \vee DASCA(u, sc)] \wedge INIZIOSCA(sc, i) \wedge DURATA(sc, d) \rightarrow$</p> <p>$\exists sc', i', d', t [CHIedesca(sc', u) \wedge DASCA(sc', u)] \wedge INIZIOSCA(sc', i') \wedge DATAORE(sc', d') \wedge DATAORA(t) \wedge i' <= t <= i + d' \text{ ORE} \wedge i' <= t <= i + d' \text{ ORE}$</p>
---	--

<p>3 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ...SCAMBIO.....</p> <p>Operazioni, vincoli:</p> <p>V. ABILITA POSSEDUTA</p> $\forall u, sc, a \ DASCA(u, sc) \wedge ABISCA(sc, a) \rightarrow ABIVTE(a, u)$ <p>V. DA NON CHIEDE</p> $\forall u, u', sc \ DASCA(u, sc) \wedge CHIEDESCA(u', sc) \rightarrow u \neq u'$ <p>V. DOPO ISCRIZIONE</p> $\forall sc, u, u', i, i', is \ DASCA(u, sc) \wedge CHIEDESCA(u', sc) \wedge ISTISC(u, i) \wedge ISTISC(u', i') \wedge INIZIOSCA(sc, is) \rightarrow i < is \wedge i' < is$	<p>6 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ...SCAMBIO.....</p> <p>Operazioni, vincoli:</p> <p>VALORE(): REALE >= 0</p> <ul style="list-style-type: none"> - pre; - post; <p>SIANO d, m TALI CHE:</p> $\text{DURATA ORE}(this, d) \wedge \exists a, u, i \ DASCA(this, u) \wedge ABISCA(this, a) \wedge \text{MEDIA}(a, u, i, m) \wedge \text{INIZIOSCA}(this, i)$ $\text{RESULT} = d(1 + \text{ARCTAN}(0.05 * m))$
<p>4 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ...ABIVTE.....</p> <p>Operazioni, vincoli:</p> <p>MEDIA(t; DATA ORA); REALE >= 0</p> <ul style="list-style-type: none"> - pre; - post; $F = \left\{ (s, u) \mid \begin{array}{l} \exists u. \text{FEEDBACK}(u, s) \wedge \text{VOTO}(u, s, v) \\ \wedge DASCA(s, this, u) \wedge ABISCA(s, this, a) \\ \wedge \exists i. \text{INIZIOSCA}(i, s) \wedge i < t \end{array} \right\}$ $(F =0 \rightarrow \text{RESULT}=0) \wedge$ $(F >0 \rightarrow \text{RESULT} = \sum_{(s, v) \in F} v / F)$	<p>7 Tipo: Classe Associazione (cerchiare)</p> <p>Nome:</p> <p>Operazioni, vincoli:</p>
<p>5 Tipo: Classe Associazione (cerchiare)</p> <p>Nome:</p> <p>Operazioni, vincoli:</p>	<p>8 Tipo: Classe Associazione (cerchiare)</p> <p>Nome:</p> <p>Operazioni, vincoli:</p>

Specifiche dei tipi di dato, specifiche di ulteriori vincoli esterni ed altre specifiche

EMAIL: STRINGA SECONDO STANDARD

TEL: STRINGA SECONDO STANDARD

UTENTE.SALDO (t : DATAORA): REALE ≥ 0

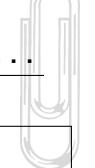
- pre: $\exists i \text{ ISTISC}(ths, i) \wedge t \geq i$

- post:

$$S = \left\{ (s, v) \mid \begin{array}{l} \text{SCAMBIO}(s) \wedge \text{CHIEDESCA}(ths, s) \wedge \text{VALORE}(s, v) \wedge \\ \exists is \text{ INIZIOSCA}(is, s) \wedge is \leq t \end{array} \right\}$$

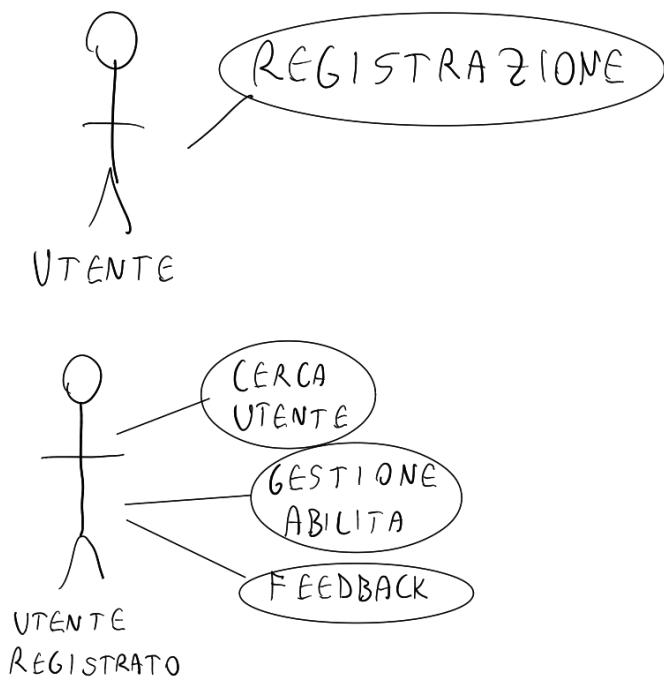
$$G = \left\{ (s, v) \mid \begin{array}{l} \text{SCAMBIO}(s) \wedge \text{DASCA}(ths, s) \wedge \text{VALORE}(s, v) \wedge \\ \exists is \text{ INIZIOSCA}(is, s) \wedge is \leq t \end{array} \right\}$$

$$\text{RESULT} \in X + \sum_{(s, v) \in G} v + \sum_{(s, v) \in S} v$$



Risposta alla Domanda 2 (segue)

Domanda 3 (5 minuti; 10 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML degli use-case che definisca ad alto livello tutte le funzionalità richieste al sistema.

Risposta

Domanda 4 (10 minuti) Proseguire la fase di Analisi Concettuale dei requisiti definendo la **segnatura** delle operazioni in ogni use-case.

Risposta

REGISTRAZIONE;

- NEW UTENTE ($n: \text{STRINGA}, c: \text{STRINGA}, e: \text{EMAIL}, t: \text{TEL}$): UTENTE

CERCA UTENTE;

- CERCA_ABILITA ($a: \text{ABILITA}, m: \text{REALC} \geq 0$): UTENTE [0..*]
- ABILITA_UTENTE ($u: \text{UTENTE}$): ABILITA [0..*]

GESTIONE ABILITA;

- NEW_ABILITA ($u: \text{UTENTE}, a: \text{ABILITA}$)
- REMOVE_ABILITA ($u: \text{UTENTE}, a: \text{ABILITA}$)
- MOLTA_BI ($u: \text{UTENTE}, a: \text{ABILITA}$): REALC >= 0

FEEDBACK;

- NEW_FEEDBACK ($u: \text{UTENTE}, s: \text{SCAMBIO}, v: 1..10$)

Domanda 5 (30 minuti; 60 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti producendo le specifiche concettuali per le operazioni di use-case, **limitandosi** a quelle necessarie a modellare i requisiti contrassegnati dalla barra laterale (come quella qui a sinistra), ed includendo eventuali operazioni ausiliarie. In particolare, per ogni operazione, definire segnatura, precondizioni e postcondizioni utilizzando il linguaggio della logica del primo ordine. Si assuma lo stesso vocabolario definito alla [Domanda 2](#).

Una risposta soddisfacente a questa domanda è condizione *necessaria* (ma non sufficiente) per superare la prova.

Risposta

CERCA MEDIA ($a: ABILITA, m: REALE \geq 0$): UTENTE [$0..*$]

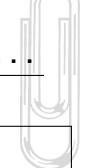
- pre:
- post:

RESULT = $\left\{ u \mid \begin{array}{l} \text{UTENTE}(u) \wedge \text{ABIVTE}(u, a) \wedge \exists m' \text{ MEDIA}(a, u, m') \\ \wedge m' \geq m \end{array} \right\}$

MOLTAZI ($u: UTENTE, a: ABILITA$): REALE ≥ 0

- pre: ABIVTE(u, a)
- post: SIA m TALE CHE: MEDIA($a, u, ADESSO, m$)

RESULT = m



Risposta alla Domanda 5 (segue)

2 Progettazione della base dati e delle funzionalità

Domanda 6 (20 minuti; 30 minuti al massimo) Iniziare la fase di progettazione logica della base di dati decidendo il DBMS da utilizzare e ristrutturando lo schema UML delle classi concettuale, il dizionario dei dati e i vincoli esterni. In particolare:

- progettare una corrispondenza tra i tipi di dato concettuali ed opportuni domini SQL (domini base o utente, oppure realizzati mediante relazioni aggiuntive) supportati dal DBMS scelto
- eliminare attributi multivale o composti
- eliminare relazioni is-a e generalizzazioni
- definire un identificatore primario per ogni classe
- ristrutturare i vincoli esterni per renderli consistenti con la struttura del nuovo diagramma.

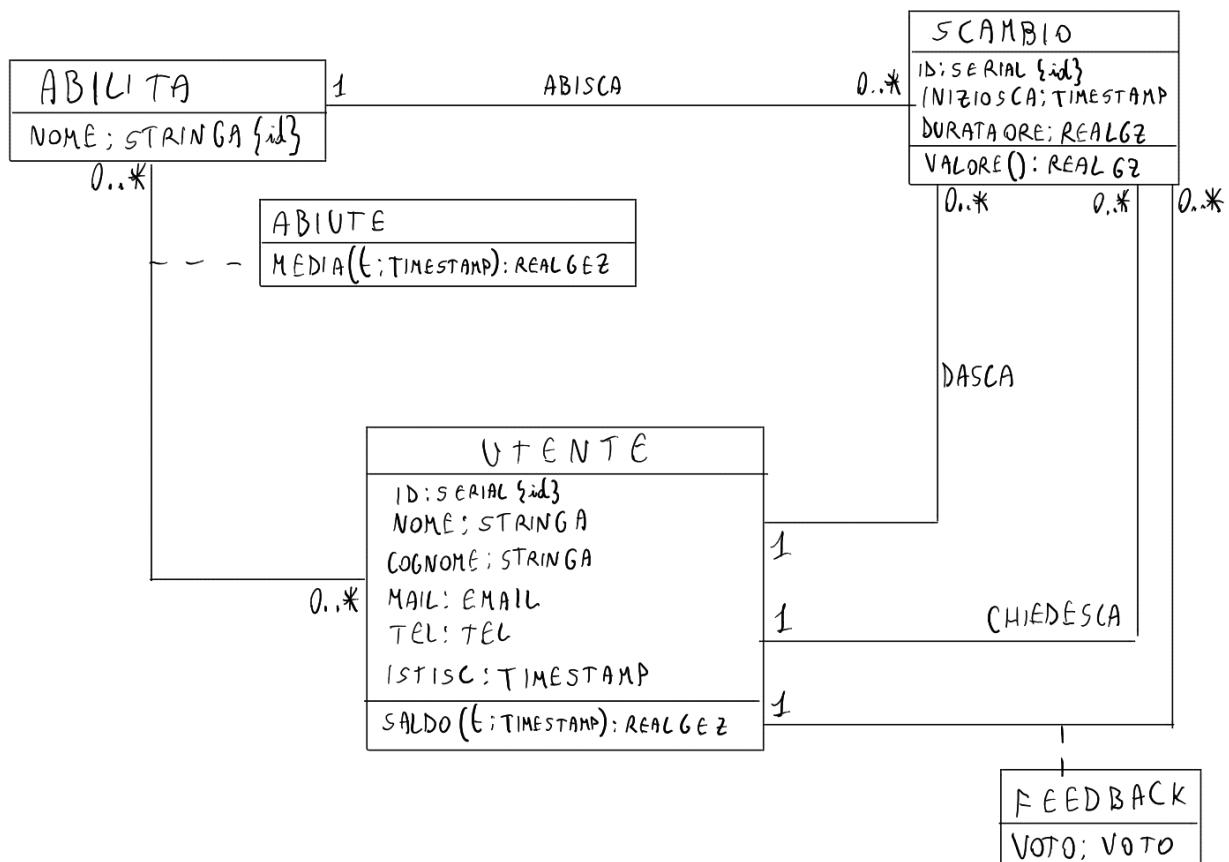
Descrivere brevemente le principali scelte effettuate.

DBMS da utilizzare ... POSTGRESQL.....

Corrispondenza tra tipi di dato concettuali e domini supportati dal DBMS

```
CREATE DOMAIN STRINGA AS VARCHAR NOT NULL;  
CREATE DOMAIN REALGZ AS REAL CHECK(VALUE>0);  
CREATE DOMAIN REALGEZ AS REAL CHECK(VALUE>=0);  
CREATE DOMAIN EMAIL AS VARCHAR;  
CREATE DOMAIN TEL AS VARCHAR;  
CREATE DOMAIN VOTO AS INTEGER CHECK(VALUE>=1 AND VALUE<=10);
```

Diagramma UML delle classi ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

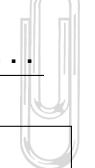
ELIMINATA GENERALIZZAZIONE SU FEEDBACK

Vincoli esterni introdotti o modificati durante la fase di ristrutturazione

(si omettano i vincoli esterni la cui formulazione è rimasta identica a seguito della ristrutturazione)

$\forall \text{FEEDBACK}, \text{SECHIEDE}$

$\forall u, s \text{ FEEDBACK}(u, s) \rightarrow \text{CHIEDESCA}(u, s)$



Risposta alla Domanda 6 (segue)

Domanda 7 (30 minuti; 60 minuti al massimo) Proseguire la fase di progettazione logica della base di dati producendo lo schema relazionale della base dati e i relativi vincoli a partire dallo schema UML delle classi ristrutturato.

Una risposta soddisfacente a questa domanda è condizione *necessaria* (ma non sufficiente) per superare la prova.

1 Relazione .ABILITA.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>NOME</u>	
Domini STRINGA	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

2 Relazione .UTENTE.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>ID</u> NOME COGNOME MAIL TEL <u>ISTISC</u>	
Domini SERIAL STRINGA STRINGA EMAIL TEL TIMESTAMP	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

3 Relazione .ABI_UT.F.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>ABILITA</u> UTENTE	
Domini STRINGA INTEGER	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

FK(ABILITA) REF ABILITA(NOME)

FK(UTENTE) REF UTENTE(ID)

La relazione accorda le relazioni che implementano le seguenti associazioni:

4 Relazione .SCAMBIO.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>ID</u> INIZIOSCA DURATAORE DA CHIEDE VOTO * ABILITA	
Domini SERIAL TIMESTAMP REAL GZ INTEGER INTEGER VOTO STRINGA	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

FK(DA) REF UTENTE(ID) FK(ABILITA) REF ABILITA(NOME) CHECK(DA > CHIEDE)

FK(CHIEDE) REF UTENTE(ID) (DA, ABILITA) ⊑ ABI_UT.F(UTENTE, ABILITA)

La relazione accorda le relazioni che implementano le seguenti associazioni: .ABISCA., DASCA., CHIEDESCA..

5 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

6 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

7 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

8 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

9 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

10 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

11 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

12 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

13 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

14 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

15 Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

16	Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi		
Domini		

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

17	Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi		
Domini		

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

18	Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi		
Domini		

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

19	Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi		
Domini		

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

20	Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi		
Domini		

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

Ulteriori vincoli esterni

Per ogni ulteriore vincolo esterno (non ancora espresso perché non definibile mediante vincoli di chiave, foreign key, ennupla, dominio, inclusione), progettare un trigger che lo implementi, definendo: (a) gli eventi da intercettare (inserimento, modifica, eliminazione di ennupple); (b) quando intercettare tali eventi (appena prima o subito dopo l'evento intercettato); (c) la relativa funzione in pseudo-codice con SQL immerso che implementa il controllo del vincolo.

T. UTENTE. NO SALDO NEGATIVO

- INSERT OR UPDATE SCAMBIO

VALID = EXISTS (SELECT * FROM SCAMBIO
WHERE VALORE (ID) < SALDO (DA, INIZIOSCA) AND ID = NEW.ID)

IF VALID: COMMIT

ELSE: ERRORE

T. UTENTE. SCAMBI DISGIUNTI

- INSERT OR UPDATE SCAMBIO

VALID = NOT EXISTS (SELECT * FROM SCAMBIO
WHERE [DA=NEW.DA OR DA=NEW.CHIODE OR CHIODE=NEW.CHIODE OR CHIODE=NEW.DA]
AND ID <> NEW.ID AND (INIZIOSCA, INTERVAL '1 HOUR' * DURATAORE) OVERLAPS
(NEW, INIZIOSCA, INTERVAL '1 HOUR' * NEW.DURATAORE))

IF VALID: COMMIT

ELSE: ERRORE

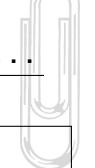
T. SCAMBIO. DOPO ISCRIZIONE

- INSERT OR UPDATE SCAMBIO

VALID = EXISTS (SELECT * FROM UTENTE U1, UTENTE U2
WHERE U1.ID = NEW.DA AND U2.ID = NEW.CHIODE
AND U1.ISTISC <= NEW.INIZIOSCA AND U2.ISTISC <= NEW.INIZIOSCA)

IF VALID: COMMIT

ELSE: ERRORE



Risposta alla Domanda 7 (segue)

Domanda 8 (30 minuti; 45 minuti al massimo) Proseguire la fase di progettazione dell'applicazione producendo le specifiche realizzative delle operazioni di classe e/o use-case definite per modellare i requisiti contrassegnati dalla barra laterale della specifica dei requisiti.

In particolare, per ogni operazione definire la segnatura, in termini di nome dell'operazione, nomi e dominio SQL degli argomenti, dominio SQL dell'eventuale valore di ritorno, e un algoritmo in pseudo-codice con SQL immerso che verifichi le precondizioni e garantisca il raggiungimento delle postcondizioni definite in fase di Analisi. Specificare, per ogni operazione, se debba essere implementata nel DBMS o nel *back-end*.

Una risposta soddisfacente a questa domanda è condizione *necessaria* (ma non sufficiente) per superare la prova.

Risposta

CERCAMEDIA ($a: \text{STRINGA}$, $m: \text{REALGEZ}$): $\text{INTEGER}[0..\ast]$

$R = \text{SELECT } u.\text{ID} \text{ FROM UTENTE } u, \text{ABILITE } au$
 $\text{WHERE } au.\text{UTENTE} = u.\text{ID} \text{ AND } au.\text{ABILITA} = a$
 $\text{AND } \text{MEDIA}(u.\text{ID}, a) \geq m$

RETURN R

MOLTABI ($u: \text{INTEGER}$, $a: \text{STRINGA}$): REALGEZ

ERRORS $\equiv \text{NOT EXISTS}(\text{SELECT } * \text{ FROM ABILUTE}$
 $\text{WHERE UTENTE} = u \text{ AND ABILITA} = a)$

IF ERROR; ERROR

$R = \text{SELECT MEDIA}(u, a, \text{NOW}())$

RETURN R

CREATE FUNCTION MEDIA($u: \text{INTEGER}$, $a: \text{STRINGA}$, $t: \text{TIMESTAMP}$): REALGEZ

$R = \text{WITH F AS}(\text{SELECT AVG(VOTO)} m \text{ FROM SCAMBIO}$
 $\text{WHERE DA} = u \text{ AND ABILITA} = a \text{ AND INIZIOSCAL} = t)$

SELECT CASE

WHEN m IS NULL THEN 0

ELSE m

END

FROM F

RETURN R

Risposta alla Domanda 8 (segue)

CREATE FUNCTION VALORE (S: INTEGER): REALGEZ

R=SELECT DURATAORE (1+ARCTAN (0.05*MEDIA (DA, ABILITA, INIZIOSCA)))
FROM SCAMBIO
WHERE ID=S

RETURN R

CREATE FUNCTION SALDO (U: INTEGER, t; TIMESTAMP): REALGEZ

IF t < (SELECT ISTISC FROM UTENTE
WHERE ID=U): ERRORE

R=WITH S AS (SELECT SUM(VALORE(ID)) S FROM SCAMBIO
WHERE CHIEDE=U AND INIZIOSCA<=t),

G AS (SELECT SUM(VALORE(ID)) G FROM SCAMBIO
WHERE DA=U AND INIZIOSCA<=t)

SELECT X + G, S - G FROM S, G

RETURN R