



SAPIENZA
UNIVERSITÀ DI ROMA

Basi di Dati, Modulo 2

Sapienza Università di Roma

Facoltà di Ing. dell'Informazione, Informatica e Statistica

Laurea in Informatica

Prof. Toni Mancini

<http://tmancini.di.uniroma1.it>

Progetto 20070703 (P.20070703)

DormoDaTe

Versione 2016-02-28

Indice

Indice	1
1 Introduzione	3
2 Specifica dei Requisiti	5
A Analisi Concettuale	7
A.1 Raffinamento dei Requisiti	9
A.1.1 Testo	9
A.1.2 Soluzione	10
A.2 Diagramma ER	13
A.2.1 Testo	13
A.2.2 Soluzione	14
A.3 Estensione del Diagramma ER	19
A.3.1 Testo	19
A.3.2 Soluzione	20
A.4 Identificazione delle Entità e Definizione Informale di Vincoli Esterni	25
A.4.1 Testo	25
A.4.2 Soluzione	26
A.5 Diagramma UML degli Use-Case	33
A.5.1 Testo	33
A.5.2 Soluzione	34

A.6	Specifiche Concettuali Informali degli Use-Case	35
A.6.1	Testo	35
A.6.2	Soluzione	36
A.7	Vincoli Esterni	43
A.7.1	Testo	43
A.7.2	Soluzione	44
A.8	Specifiche Concettuali degli Use-Case	51
A.8.1	Testo	51
A.8.2	Soluzione	52

1

Introduzione

Si vuole progettare e realizzare *DormoDaTe.com*, un sistema per un portale web che consenta agli iscritti di chiedere ospitalità, durante i loro viaggi, a casa di altri iscritti. Lo scopo, oltre che quello banale di ridurre le spese di viaggio, è soprattutto quello di incentivare scambi sociali e senso di ospitalità, e di favorire nuove conoscenze.



2

Specifica dei Requisiti

Il sistema *DormoDaTe.com*¹ deve consentire agli utenti di iscriversi al portale web, dichiarando il loro profilo, ovvero nome, cognome, sesso, età, città di residenza, oltre che informazioni circa la tipologia di ospitalità che sono disposti ad offrire.

In particolare, è necessario che gli iscritti possano inserire nel loro profilo le seguenti informazioni:

- Distanza della loro abitazione dal centro città;
- Distanza dalla stazione autobus/metro/treno più vicina;
- Numero di membri della famiglia (numero di adulti e di bambini);
- Numero di persone che sono disposti ad ospitare contemporaneamente (posti letto); dei posti letto va specificato (nel modo più preciso e strutturato possibile) se si tratta di letti (singoli/doppi) in camere separate, divani in stanze comuni, ecc. (ad esempio, un iscritto può dichiarare di avere un posto in letto singolo più due posti in letto doppio in una camera, ed un posto in soggiorno sul divano).

In ogni momento ogni iscritto ha la facoltà di aggiornare le informazioni circa le sue disponibilità ad ospitare altre persone, sulla base dei suoi impegni. In particolare, il sistema deve permettere ad ogni iscritto la possibilità di dichiarare delle date o dei periodi nei quali non è disponibile a ricevere ospiti, ad es. perché fuori (la ragione non è di interesse per il sistema).

I viaggiatori che desiderano organizzare un viaggio usufruendo del servizio devono poter interrogare il sistema per ottenere l'insieme di tutti gli iscritti della città desiderata che sono disposti a ricevere ospiti nel periodo richiesto e hanno, per le date richieste, posti sufficienti disponibili.

¹Il sistema vuole mettersi in diretta concorrenza con il ben più famoso *couchsurfing.com*.

Una volta scelta una soluzione di alloggio, i viaggiatori possono effettuare la relativa richiesta di prenotazione, specificando tutte le informazioni necessarie (date, persone, scelta dei posti letto, ecc.). Si noti che è possibile per un viaggiatore prenotare anche per eventuali accompagnatori: tuttavia il sistema impone che tutti (chi prenota ed eventuali accompagnatori) siano iscritti al sistema (questo per monitorare la qualità e l'affidabilità degli ospiti, cfr. seguito). Il relativo ospitante (membro padrone di casa) può decidere liberamente se accettare la richiesta o rifiutarla (in quest'ultimo caso deve specificare una motivazione). Tutte le richieste di prenotazioni, quelle accettate e quelle rifiutate, devono essere mantenute in modo persistente dal sistema.

La difficoltà maggiore che si riscontra in sistemi del genere è quella relativa alla sicurezza, visto che, di fatto, gli iscritti offrono ospitalità a casa propria a perfetti estranei (e, dualmente, un viaggiatore va a dormire a casa di sconosciuti). Per questo motivo, sono di grande importanza quei meccanismi che permettono di avere maggiore confidenza circa la qualità e l'affidabilità dell'ospitante e degli ospitati. In particolare, *DormoDaTe.com* deve permettere dei meccanismi di feedback per valutare la qualità e l'affidabilità sia dei padroni di casa che dei viaggiatori. In dettaglio:

- Ogni viaggiatore, dopo aver alloggiato presso un iscritto ospitante, deve necessariamente esprimere una valutazione su quest'ultimo, in termini di un intero da 0 (scarso) a 5 (ottimo);
- Ogni ospitante, all'atto della partenza di un ospite, può esprimere una valutazione su quest'ultimo, anche qui in termini di un intero da 0 a 5.

Si noti che, la valutazione di un ospitante su un ospitato è facoltativa, mentre quella opposta è obbligatoria. In particolare, sebbene quest'ultima possa essere effettuata anche in un tempo successivo, il sistema deve fare in modo che un viaggiatore non possa prenotare più alcuna nuova sistemazione se non ha valutato tutti gli ospitanti presso cui ha alloggiato.

Parte A

Analisi Concettuale

A.1

Raffinamento dei Requisiti

A.1.1 Testo

Raffinare la specifica dei requisiti eliminando inconsistenze, omissioni o ridondanze e producendo un elenco numerato di requisiti il meno ambiguo possibile.

A.1.2 Soluzione

1. Degli utenti iscritti interessa:
 - 1.1. nome
 - 1.2. cognome
 - 1.3. sesso
 - 1.4. anno di nascita
 - 1.5. città di residenza
 - 1.6. abitazioni messe a disposizione ([Req. 2.](#))
2. Delle abitazioni degli utenti interessa:
 - 2.1. la città
 - 2.2. la distanza dal centro città
 - 2.3. la distanza dalla stazione autobus/metro/treno più vicina
 - 2.4. il numero di membri della famiglia (numero di adulti e di bambini)
 - 2.5. le sistemazioni offerte ([Req. 3.](#) e [Req. 4.](#))
3. Delle stanze in cui sono offerte sistemazioni interessa:
 - 3.1. l'abitazione in cui sono collocate ([Req. 2.](#))
 - 3.2. il tipo (privata o comune)
4. Delle sistemazioni offerte da un utente interessa:
 - 4.1. la stanza in cui sono collocate ([Req. 3.](#))
 - 4.2. la tipologia (letto o divano)
 - 4.3. il numero di posti
5. Il sistema deve poter rappresentare, per ogni utente, i periodi di tempo in cui non è disponibile a ricevere ospiti nelle sue diverse abitazioni
6. Il sistema deve poter rappresentare le prenotazioni effettuate.
 - 6.1. Delle prenotazioni interessa:
 - 6.1.1. il periodo (date di arrivo e ripartenza)
 - 6.1.2. le sistemazioni scelte
 - 6.1.3. l'utente che effettua la prenotazione
 - 6.1.4. gli utenti accompagnatori
 - 6.2. Delle prenotazioni rifiutate interessa la motivazione del rifiuto (testo libero).

7. Il sistema deve poter rappresentare feedback, sotto forma di interi tra 0 e 5. Questi possono essere:
 - 7.1. feedback espressi da utenti ospitati (sia in qualità di utenti che hanno effettuato una prenotazione che di utenti accompagnatori) riguardanti la qualità dell'ospitalità ricevuta
 - 7.2. feedback espressi da utenti ospitanti riguardanti la qualità di un utente ospitato
8. Il sistema deve fornire le seguenti funzionalità:
 - 8.1. gli utenti devono potersi iscrivere, inserire e aggiornare le sistemazioni offerte e i loro periodi di indisponibilità ([Req. 5.](#))
 - 8.2. gli utenti devono poter cercare, data una città c , un periodo p , ed un numero di posti n , tutte le abitazioni che offrono sistemazioni nella città c , che sono disponibili nel periodo p a ricevere ospiti e che hanno, in ogni giorno in p , almeno n posti disponibili
 - 8.3. gli utenti devono poter effettuare prenotazioni specificando: date di arrivo e ripartenza, insieme di accompagnatori, sistemazioni scelte. Una prenotazione deve essere rifiutata se l'utente non ha espresso feedback verso utenti ospitanti relativi alle sue precedenti visite (anche se in qualità di accompagnatore)
 - 8.4. gli utenti devono poter accettare o rifiutare le richieste di prenotazione delle sistemazioni che offrono. In caso di rifiuto deve essere specificata una motivazione (testo libero)
 - 8.5. gli utenti, dopo aver alloggiato presso un utente ospitante, devono poter esprimere un feedback su quest'ultimo.
 - 8.6. gli utenti, dopo aver ospitato un utente, devono poter esprimere un feedback su quest'ultimo.



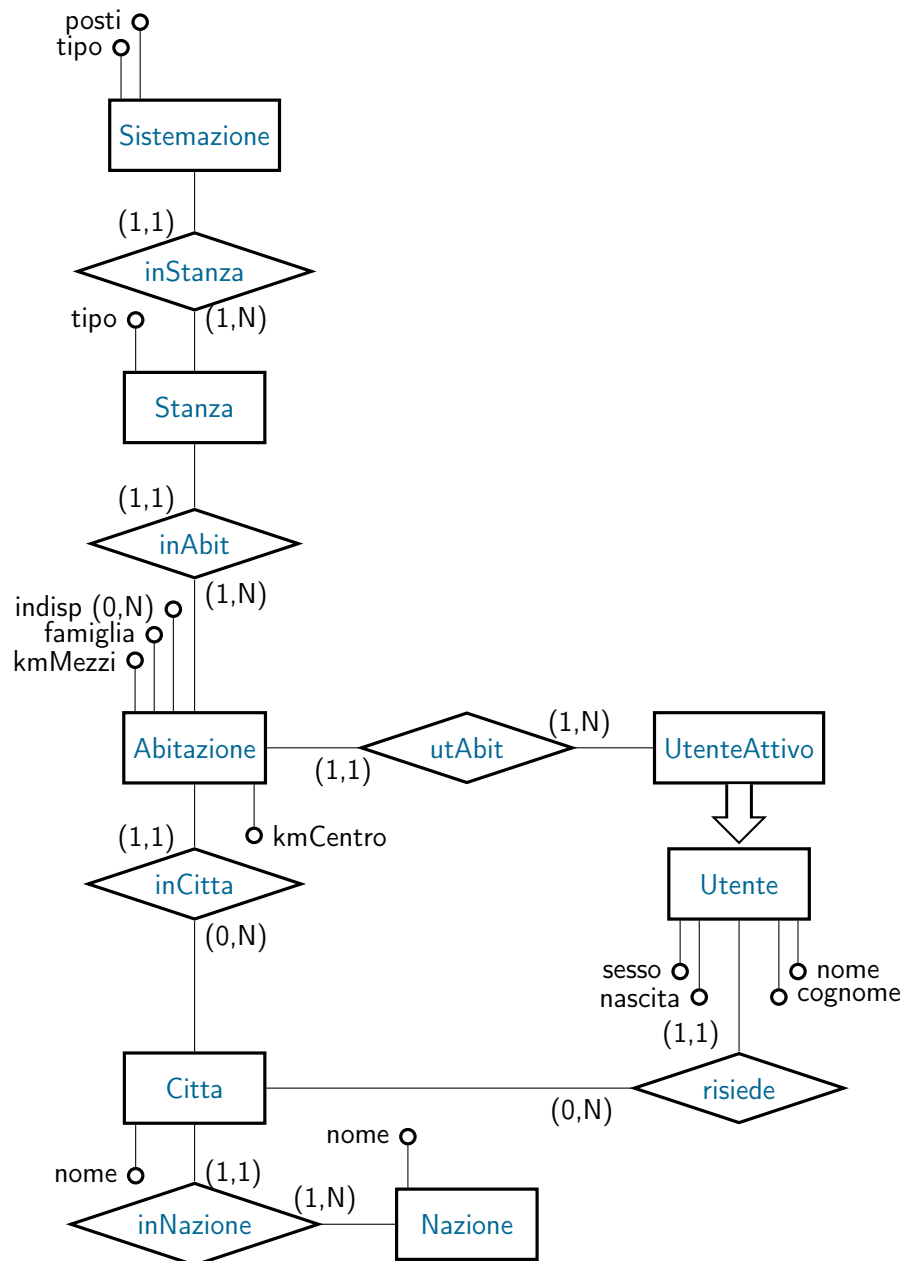
A.2

Diagramma ER

A.2.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti a partire dall'output della fase di raffinamento effettuata al passo A.1. In particolare, produrre il diagramma ER concettuale per l'applicazione ed il dizionario dei dati per modellare i seguenti requisiti: [Req. 1.](#), [Req. 2.](#), [Req. 3.](#), [Req. 4.](#), [Req. 5.](#)

A.2.2 Soluzione



Specifiche dei Dati

Entità **Utente**

Ogni istanza di questa entità rappresenta un utente registrato ([Req. 1.](#)). Le istanze di questa entità potrebbero non essere coinvolte in istanze della relationship **utAbit**

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome dell'utente (Req. 1.1.)
cognome	stringa		Il cognome dell'utente (Req. 1.2.)
sex	{M,F}		Il sesso dell'utente (Req. 1.3.)
nascita	intero > 0		L'anno di nascita dell'utente (Req. 1.4.)

Entità **UtenteAttivo**

Ogni istanza di questa entità rappresenta un utente attivo, che ha reso disponibile almeno una sua **Abitazione**, e che quindi può effettuare e ricevere prenotazioni ([Req. 1.](#))

Attributi: Nessuno

Entità **Abitazione**

Ogni istanza di questa entità rappresenta una abitazione censita dal sistema ([Req. 2.](#))

attributo	dominio	molteplicità	descrizione
kmCentro	reale ≥ 0		La distanza dal centro città (in Km) (Req. 2.2.)
kmMezzi	reale ≥ 0		La distanza dalla stazione/fermata più vicina del sistema di trasporto pubblico (in Km) (Req. 2.3.)
famiglia	Famiglia		Il numero di adulti e bambini che vivono stabilmente nell'abitazione (Req. 2.4.)
indisp	Periodo	(0,N)	I periodi di indisponibilità del proprietario di questa abitazione (Req. 5.)

Entità Stanza

Ogni istanza di questa entità rappresenta una stanza di una abitazione censita dal sistema (Req. 3.)

attributo	dominio	molteplicità	descrizione
tipo	{privata, comune}		Il tipo della stanza (Req. 3.2.)

Entità Sistemazione

Ogni istanza di questa entità rappresenta una sistemazione censita dal sistema (Req. 4.)

attributo	dominio	molteplicità	descrizione
tipo	{divano, letto}		La tipologia della sistemazione (Req. 4.2.)
posti	intero > 0		Il numero di posti della sistemazione (Req. 4.3.)

Entità Città

Ogni istanza di questa entità rappresenta una città

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della città

Entità Nazione

Ogni istanza di questa entità rappresenta una nazione

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della nazione

Relationship inNazione

Ogni istanza di questa relationship lega una Città alla Nazione a cui appartiene

Attributi: Nessuno

Relationship inCittà

Ogni istanza di questa relationship lega una Abitazione alla Città in cui è sita (Req. 2.1.)

Attributi: Nessuno

Relationship risiede

Ogni istanza di questa relationship lega un **Utente** alla **Città** in cui risiede (**Req. 1.5.**)

Attributi: Nessuno

Relationship utAbit

Ogni istanza di questa relationship lega un **Utente Attivo** ad una **Abitazione** che mette a disposizione (**Req. 1.6.**)

Attributi: Nessuno

Relationship inStanza

Ogni istanza di questa relationship lega una **Sistemazione** alla **Stanza** a cui appartiene (**Req. 4.1.**)

Attributi: Nessuno

Relationship inAbit

Ogni istanza di questa relationship lega una **Stanza** all'**Abitazione** a cui appartiene (**Req. 2.5.**)

Attributi: Nessuno

Dominio Famiglia

Il dominio è un record composto dai seguenti campi:

- adulti: intero > 0
- bambini: intero ≥ 0

Dominio Periodo

Il dominio è di tipo record, le cui istanze sono composte dai seguenti campi:

- da: data
- a: data



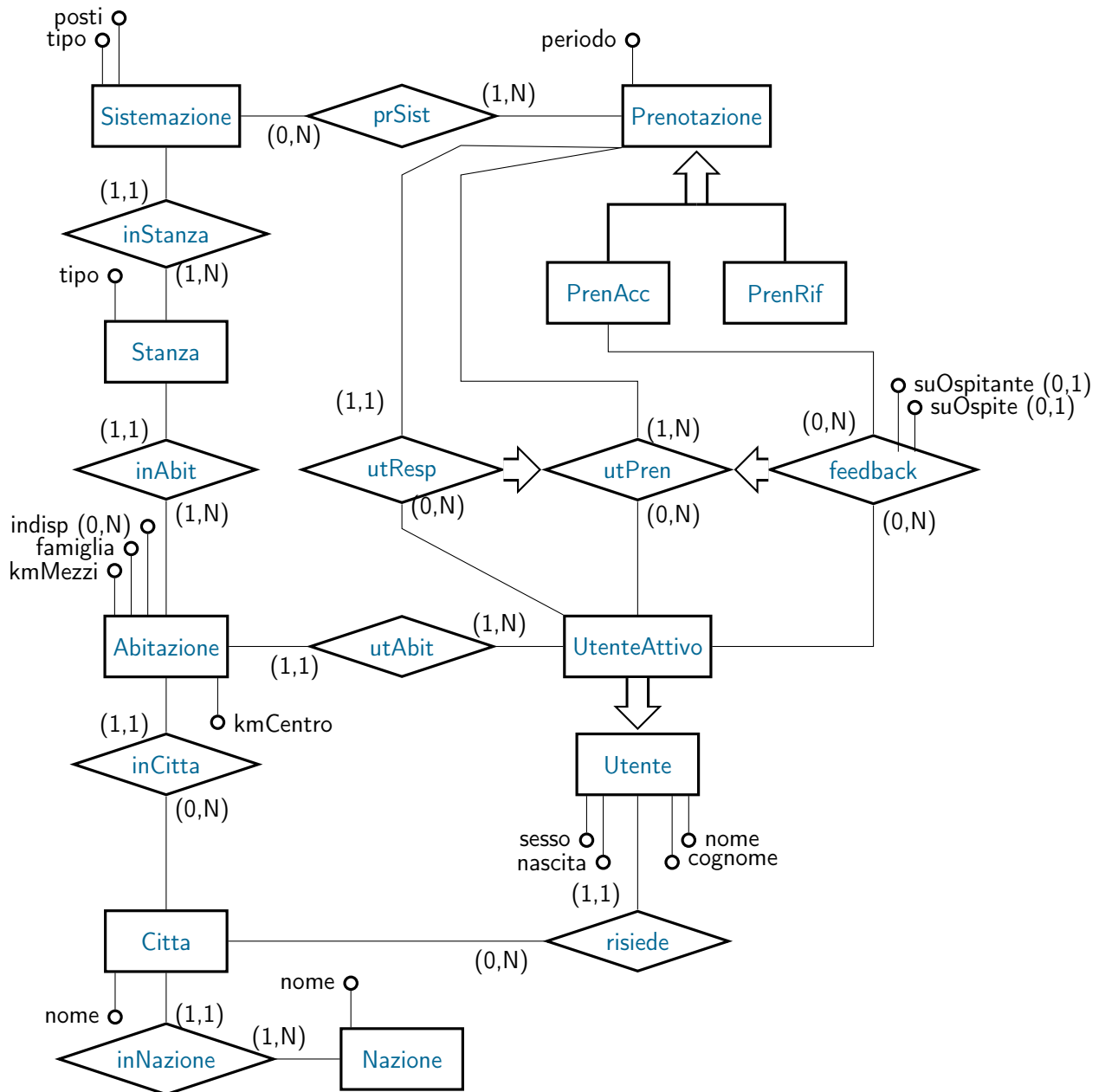
A.3

Estensione del Diagramma ER

A.3.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti estendendo il diagramma ER concettuale per l'applicazione ed il dizionario dei dati per modellare anche i seguenti requisiti: [Req. 6.](#), [Req. 7.](#)

A.3.2 Soluzione



Specifiche dei Dati

Entità **Utente**

Ogni istanza di questa entità rappresenta un utente registrato ([Req. 1.](#)). Le istanze di questa entità potrebbero non essere coinvolte in istanze della relationship **utAbit**

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome dell'utente (Req. 1.1.)
cognome	stringa		Il cognome dell'utente (Req. 1.2.)
sex	{M,F}		Il sesso dell'utente (Req. 1.3.)
nascita	intero > 0		L'anno di nascita dell'utente (Req. 1.4.)

Entità **UtenteAttivo**

Ogni istanza di questa entità rappresenta un utente attivo, che ha reso disponibile almeno una sua **Abitazione**, e che quindi può effettuare e ricevere prenotazioni ([Req. 1.](#))

Attributi: Nessuno

Entità **Abitazione**

Ogni istanza di questa entità rappresenta una abitazione censita dal sistema ([Req. 2.](#))

attributo	dominio	molteplicità	descrizione
kmCentro	reale ≥ 0		La distanza dal centro città (in Km) (Req. 2.2.)
kmMezzi	reale ≥ 0		La distanza dalla stazione/fermata più vicina del sistema di trasporto pubblico (in Km) (Req. 2.3.)
famiglia	Famiglia		Il numero di adulti e bambini che vivono stabilmente nell'abitazione (Req. 2.4.)
indisp	Periodo	(0,N)	I periodi di indisponibilità del proprietario di questa abitazione (Req. 5.)

Entità Stanza

Ogni istanza di questa entità rappresenta una stanza di una abitazione censita dal sistema (Req. 3.)

attributo	dominio	molteplicità	descrizione
tipo	{privata, comune}		Il tipo della stanza (Req. 3.2.)

Entità Sistemazione

Ogni istanza di questa entità rappresenta una sistemazione censita dal sistema (Req. 4.)

attributo	dominio	molteplicità	descrizione
tipo	{divano, letto}		La tipologia della sistemazione (Req. 4.2.)
posti	intero > 0		Il numero di posti della sistemazione (Req. 4.3.)

Entità Città

Ogni istanza di questa entità rappresenta una città

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della città

Entità Nazione

Ogni istanza di questa entità rappresenta una nazione

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della nazione

Entità Prenotazione

Ogni istanza di questa entità rappresenta una prenotazione (Req. 6.)

attributo	dominio	molteplicità	descrizione
periodo	Periodo		Il periodo della prenotazione (Req. 6.1.1.)

Entità PrenAcc

Ogni istanza di questa entità rappresenta una prenotazione accettata

Attributi: Nessuno

Entità PrenRif

Ogni istanza di questa entità rappresenta una prenotazione rifiutata

Attributi: Nessuno

Relationship inNazione

Ogni istanza di questa relationship lega una Città alla Nazione a cui appartiene

Attributi: Nessuno

Relationship inCitta

Ogni istanza di questa relationship lega una Abitazione alla Città in cui è sita (Req. 2.1.)

Attributi: Nessuno

Relationship risiede

Ogni istanza di questa relationship lega un Utente alla Città in cui risiede (Req. 1.5.)

Attributi: Nessuno

Relationship utAbit

Ogni istanza di questa relationship lega un Utente Attivo ad una Abitazione che mette a disposizione (Req. 1.6.)

Attributi: Nessuno

Relationship inStanza

Ogni istanza di questa relationship lega una Sistemazione alla Stanza a cui appartiene (Req. 4.1.)

Attributi: Nessuno

Relationship inAbit

Ogni istanza di questa relationship lega una Stanza all'Abitazione a cui appartiene (Req. 2.5.)

Attributi: Nessuno

Relationship prSist

Ogni istanza di questa relationship lega una Prenotazione ad una delle Sistemazioni alle quali è relativa (Req. 6.1.2.)

Attributi: Nessuno

Relationship *utPren*

Ogni istanza di questa relationship lega una *Prenotazione* ad uno degli *Utenti Attivi* partecipanti, ovvero l'*Utente Attivo* che l'ha effettuata oppure un *Utente Attivo* accompagnatore (Req. 6.1.4.)

Attributi: Nessuno

Relationship *feedback*

Ogni istanza di questa relationship lega una *Prenotazione* consumata ad un *Utente* per il quale esiste un feedback. L'*UtenteAttivo* può essere sia l'*UtenteAttivo* ospitante che uno degli *Utenti Attivi* ospitati

attributo	dominio	molteplicità	descrizione
suOspite	[0, 5]	(0,1)	Il feedback sull'utente ospitato da parte dell'utente ospitante (Req. 7.2.)
suOspitante	[0, 5]	(0,1)	Il feedback sull'utente ospitante da parte dell'utente ospitato (Req. 7.1.)

Relationship *utResp*

Ogni istanza di questa relationship lega una *Prenotazione* all'*Utente Attivo* che l'ha effettuata (Req. 6.1.3.)

Attributi: Nessuno

Dominio Famiglia

Il dominio è un record composto dai seguenti campi:

- adulti: intero > 0
- bambini: intero ≥ 0

Dominio Periodo

Il dominio è di tipo record, le cui istanze sono composte dai seguenti campi:

- da: data
- a: data

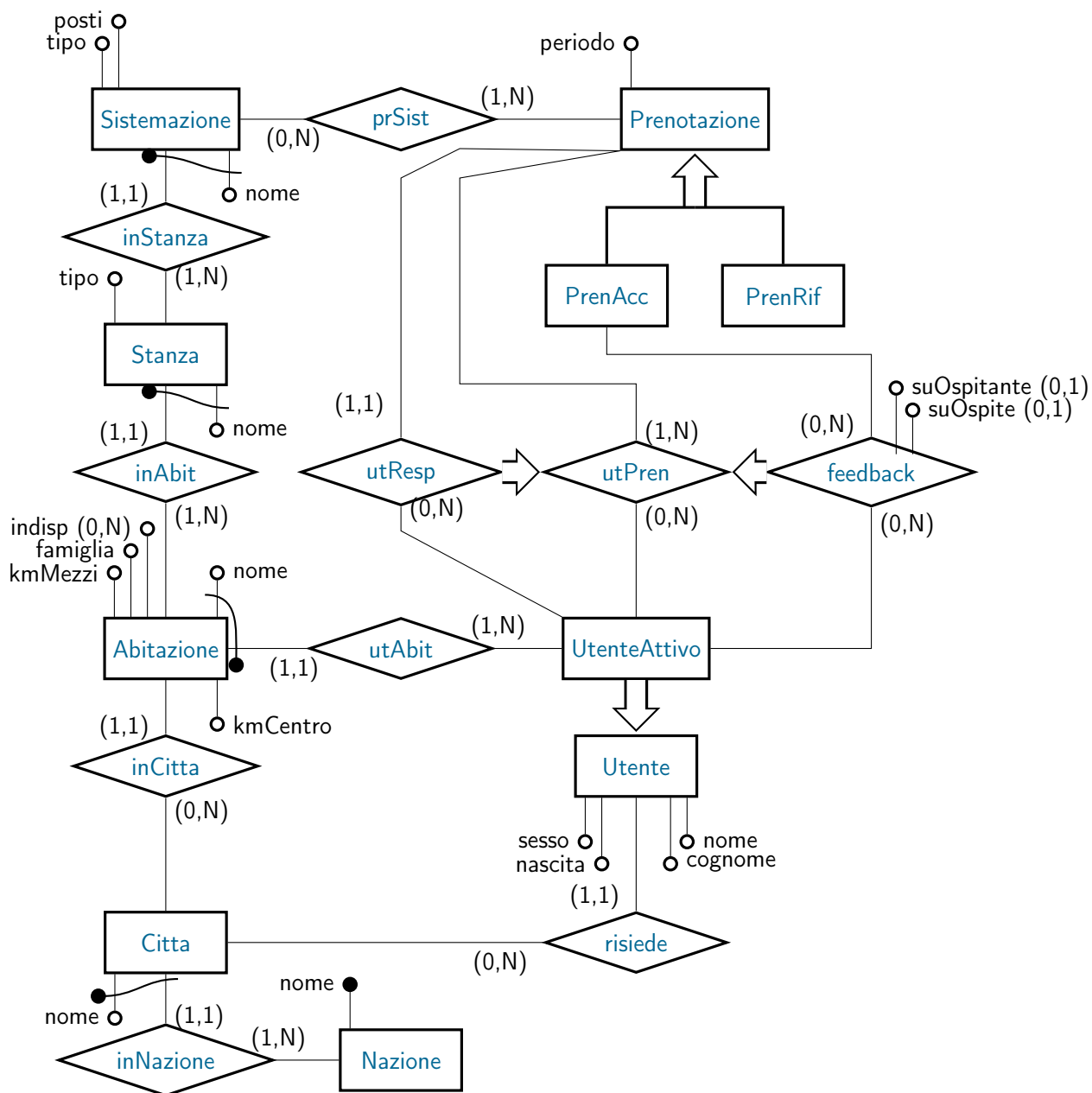
A.4

Identificazione delle Entità e Definizione Informale di Vincoli Esterni

A.4.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti estendendo il diagramma ER concettuale per l'applicazione ed il dizionario dei dati con eventuali vincoli di identificazione delle entità che provengono dai requisiti e la definizione informale di eventuali ulteriori vincoli esterni.

A.4.2 Soluzione



Specifiche dei Dati

Entità **Utente**

Ogni istanza di questa entità rappresenta un utente registrato ([Req. 1.](#)). Le istanze di questa entità potrebbero non essere coinvolte in istanze della relationship **utAbit**

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome dell'utente (Req. 1.1.)
cognome	stringa		Il cognome dell'utente (Req. 1.2.)
sex	{M,F}		Il sesso dell'utente (Req. 1.3.)
nascita	intero > 0		L'anno di nascita dell'utente (Req. 1.4.)

Entità **UtenteAttivo**

Ogni istanza di questa entità rappresenta un utente attivo, che ha reso disponibile almeno una sua [Abitazione](#), e che quindi può effettuare e ricevere prenotazioni ([Req. 1.](#))

Attributi: Nessuno

Entità **Abitazione**

Ogni istanza di questa entità rappresenta una abitazione censita dal sistema ([Req. 2.](#))

attributo	dominio	molteplicità	descrizione
kmCentro	reale ≥ 0		La distanza dal centro città (in Km) (Req. 2.2.)
kmMezzi	reale ≥ 0		La distanza dalla stazione/fermata più vicina del sistema di trasporto pubblico (in Km) (Req. 2.3.)
famiglia	Famiglia		Il numero di adulti e bambini che vivono stabilmente nell'abitazione (Req. 2.4.)
indisp	Periodo	(0,N)	I periodi di indisponibilità del proprietario di questa abitazione (Req. 5.)
nome	stringa		Il nome dell'abitazione

Entità Stanza

Ogni istanza di questa entità rappresenta una stanza di una abitazione censita dal sistema (Req. 3.)

attributo	dominio	molteplicità	descrizione
tipo	{privata, comune}		Il tipo della stanza (Req. 3.2.)
nome	stringa		Il nome della stanza

Entità Sistemazione

Ogni istanza di questa entità rappresenta una sistemazione censita dal sistema (Req. 4.)

attributo	dominio	molteplicità	descrizione
tipo	{divano, letto}		La tipologia della sistemazione (Req. 4.2.)
posti	intero > 0		Il numero di posti della sistemazione (Req. 4.3.)
nome	stringa		Il nome della sistemazione

Entità Città

Ogni istanza di questa entità rappresenta una città

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della città

Entità Nazione

Ogni istanza di questa entità rappresenta una nazione

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della nazione

Entità Prenotazione

Ogni istanza di questa entità rappresenta una prenotazione (Req. 6.)

attributo	dominio	molteplicità	descrizione
periodo	Periodo		Il periodo della prenotazione (Req. 6.1.1.)

Vincoli:

[V.Prenotazione.ospitante] Le **Sistemazioni** relative ad ogni singola **Prenotazione** sono tutte della stessa **Abitazione** (e quindi offerte dallo stesso **Utente** ospitante):

Per ogni istanza p dell'entità **Prenotazione** e per ogni coppia $sist'$ e $sist''$ di istanze dell'entità **Sistemazione** tali che sia $(p, sist')$ che $(p, sist'')$ sono istanze della relationship **prSist**:

- sia st' l'istanza di entità **Stanza** tale che $(sist', st')$ è istanza della relationship **inStanza**
- sia st'' l'istanza di entità **Stanza** tale che $(sist'', st'')$ è istanza della relationship **inStanza**
- sia a' l'istanza di entità **Abitazione** tale che (st', a') è istanza della relationship **inAbit**.

Deve essere che (st'', a') è istanza della relationship **inAbit**.

[V.Prenotazione.posti] Il numero totale di posti delle **Sistemazioni** associate ad ogni singola **Prenotazione** è pari o superiore al numero di **Utenti** ospitati.

Per ogni istanza p dell'entità **Prenotazione** siano:

- S_p l'insieme delle sistemazioni associate a p , ovvero l'insieme delle istanze $sist$ dell'entità **Sistemazione** tali che $(p, sist)$ è istanza della relationship **prSist**
- N_p la somma dei valori dell'attributo posti su tutti gli elementi di S_p
- U_p l'insieme degli utenti associati a p , ovvero l'insieme delle istanze u dell'entità **Utente** tali che (u, p) è istanza della relationship **utPrenot**.

Deve essere che N_p è maggiore o uguale alla cardinalità di U_p .

Entità **PrenAcc**

Ogni istanza di questa entità rappresenta una prenotazione accettata

Attributi: Nessuno

Entità **PrenRif**

Ogni istanza di questa entità rappresenta una prenotazione rifiutata

Attributi: Nessuno

Relationship **inNazione**

Ogni istanza di questa relationship lega una **Città** alla **Nazione** a cui appartiene

Attributi: Nessuno

Relationship inCittà

Ogni istanza di questa relationship lega una **Abitazione** alla **Città** in cui è sita (Req. 2.1.)

Attributi: Nessuno

Relationship risiede

Ogni istanza di questa relationship lega un **Utente** alla **Città** in cui risiede (Req. 1.5.)

Attributi: Nessuno

Relationship utAbit

Ogni istanza di questa relationship lega un **Utente Attivo** ad una **Abitazione** che mette a disposizione (Req. 1.6.)

Attributi: Nessuno

Relationship inStanza

Ogni istanza di questa relationship lega una **Sistemazione** alla **Stanza** a cui appartiene (Req. 4.1.)

Attributi: Nessuno

Relationship inAbit

Ogni istanza di questa relationship lega una **Stanza** all'**Abitazione** a cui appartiene (Req. 2.5.)

Attributi: Nessuno

Relationship prSist

Ogni istanza di questa relationship lega una **Prenotazione** ad una delle **Sistemazioni** alle quali è relativa (Req. 6.1.2.)

Attributi: Nessuno

Relationship utPren

Ogni istanza di questa relationship lega una **Prenotazione** ad uno degli **Utenti Attivi** partecipanti, ovvero l'**Utente Attivo** che l'ha effettuata oppure un **Utente Attivo** accompagnatore (Req. 6.1.4.)

Attributi: Nessuno

Relationship `feedback`

Ogni istanza di questa relationship lega una `Prenotazione` consumata ad un `Utente` per il quale esiste un feedback. L'`UtenteAttivo` può essere sia l'`UtenteAttivo` ospitante che uno degli `Utenti Attivi` ospitati

attributo	dominio	molteplicità	descrizione
suOspite	[0, 5]	(0,1)	Il feedback sull'utente ospitato da parte dell'utente ospitante (Req. 7.2.)
suOspitante	[0, 5]	(0,1)	Il feedback sull'utente ospitante da parte dell'utente ospitato (Req. 7.1.)

Relationship `utResp`

Ogni istanza di questa relationship lega una `Prenotazione` all'`Utente Attivo` che l'ha effettuata ([Req. 6.1.3.](#))

Attributi: Nessuno

Dominio Famiglia

Il dominio è un record composto dai seguenti campi:

- adulti: intero > 0
- bambini: intero ≥ 0

Dominio Periodo

Il dominio è di tipo record, le cui istanze sono composte dai seguenti campi:

- da: data
- a: data

[V.Periodo.legale] Per ogni istanza del dominio, il valore del campo a non deve precedere il valore del campo da.



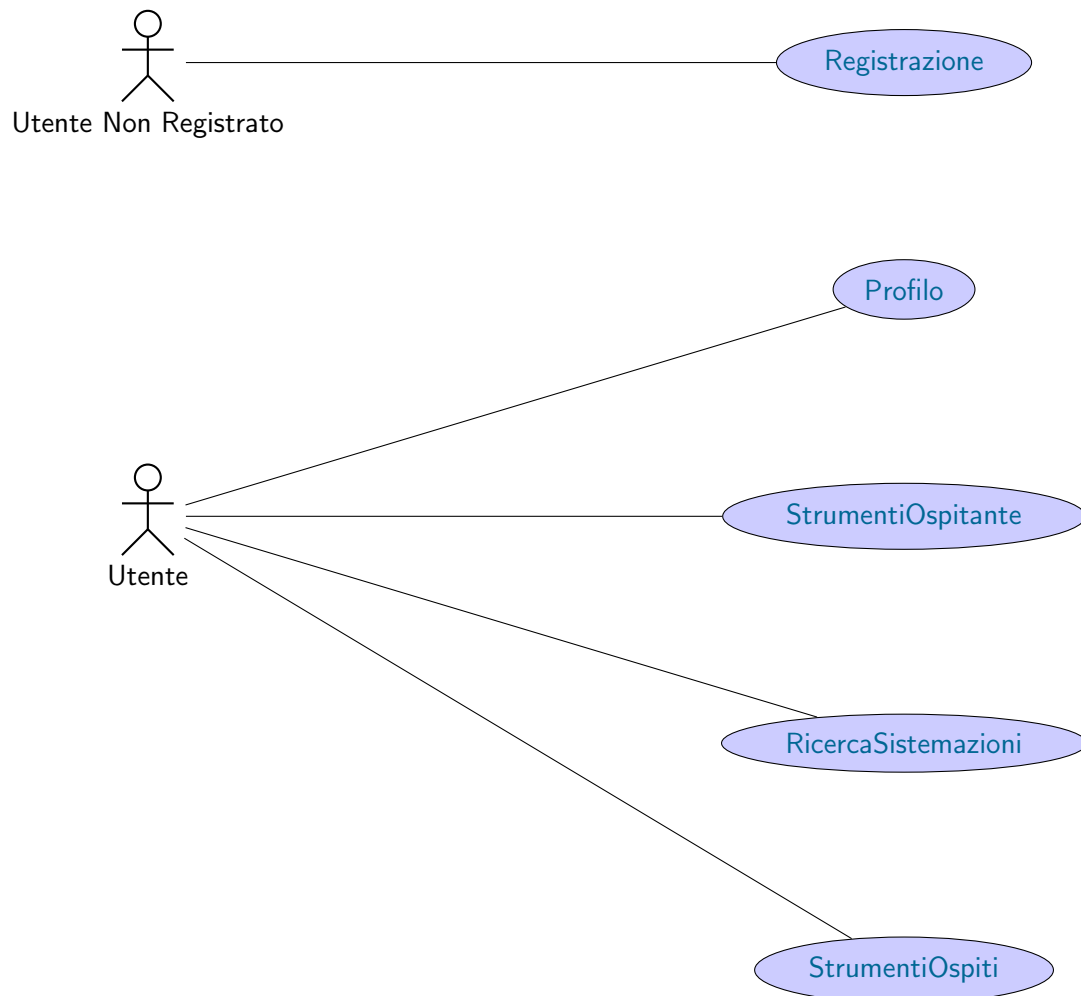
A.5

Diagramma UML degli Use-Case

A.5.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti producendo un diagramma UML degli use-case.

A.5.2 Soluzione



A.6

Specifiche Concettuali Informali degli Use-Case

A.6.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti producendo le specifiche concettuali informali degli use-case.

A.6.2 Soluzione

Specifica Use-Case **Registrazione**

- **registraUtente**(n: stringa, c: stringa, s: {M,F}, nasc : data, cit : Città)
: Utente (**Req. 8.1.**)

precondizioni: nessuna

postcondizioni: Il livello estensionale dei dati viene modificato creando le seguenti nuove istanze:

- α dell'entità **Utente**, con valori n, c, s, nasc per, rispettivamente, gli attributi nome, cognome, sesso, nascita
- (α , cit) della relationship risiede.

Viene restituita l'istanza α .

Specifica Use-Case **Profilo**

- **nuovaAbitazione**($u : \text{Utente}$, $n : \text{stringa}$, $\text{kmM} : \text{reale} \geq 0$, $\text{kmC} : \text{reale} \geq 0$, $\text{fam} : \text{Famiglia}$, $\text{cit} : \text{Citta}$) : **Abitazione** (**Req. 8.1.**)

precondizioni: nessuna

postcondizioni: Il livello estensionale dei dati viene modificato creando le seguenti nuove istanze:

- α dell'entità **Abitazione**, con valori n , kmM , kmC , fam per, rispettivamente, gli attributi **nome**, **kmMezzi**, **kmCentro**, **famiglia**
- (α, u) della relationship **utAbit**
- (α, cit) della relationship **inCitta**.

Viene restituita l'istanza α .

- **nuovaStanza**($a : \text{Abitazione}$, ...) : **Stanza** (**Req. 8.1.**)

precondizioni: ... (per esercizio)

postcondizioni: ... (per esercizio)

- **nuovaSistemazione**($st : \text{Stanza}$, ...) : **Sistemazione** (**Req. 8.1.**)

precondizioni: ... (per esercizio)

postcondizioni: ... (per esercizio)

Nota: i vincoli di molteplicità (1,N) nelle relationship **utAbit**, **inAbit**, **inStanza** impongono che, contestualmente all'invocazione della funzione **nuovaAbitazione()**, vengano invocate anche le funzioni **nuovaStanza()** e **nuovaSistemazione()**. Questo vuol dire che il sistema realizzato dovrà consentire l'inserimento di una sistemazione e di una stanza all'atto dell'inserimento di una abitazione.

Specifica Use-Case **StrumentiOspitante**

- **nuovaIndisp(a : Abitazione, p: Periodo) (Req. 8.1.)**

precondizioni: nessuna

postcondizioni: Il livello estensionale dei dati viene modificato aggiungendo p tra i valori dell'attributo indis disp dell'istanza a.
Non viene restituito alcun valore.
- **cancellaIndisp(a : Abitazione, p: Periodo) (Req. 8.1.)**

precondizioni: L'istanza a ha p tra i valori per l'attributo indis disp.

postcondizioni: Il livello estensionale dei dati viene modificato rimuovendo p tra i valori dell'attributo indis disp dell'istanza a.
Non viene restituito alcun valore.
- **approvaRichiestaPren(p : Prenotazione) (Req. 8.4.)**

precondizioni: p non è istanza dell'entità **PrenAcc** e non è istanza dell'entità **PrenRif**.

postcondizioni: Il livello estensionale dei dati viene modificato come segue:

 - p è anche istanza dell'entità **PrenAcc**.
 Non viene restituito alcun valore.
- **rifiutaRichiestaPren(p : Prenotazione, m: stringa) (Req. 8.4.)**

precondizioni: p non è istanza dell'entità **PrenAcc** e non è istanza dell'entità **PrenRif**.

postcondizioni: Il livello estensionale dei dati viene modificato come segue:

 - p è anche istanza dell'entità **PrenRif**, con valore per l'attributo motivo pari ad m.
 Non viene restituito alcun valore.
- **feedbackSuOspite(p : Prenotazione, o: UtenteAttivo, f: [0,5]) (Req. 8.6.)**

precondizioni: La prenotazione p deve essere accettata e già consumata, e l'utente o deve essere tra gli utenti ospitati relativamente alla prenotazione p:
p è istanza dell'entità **PrenAcc** e il valore del campo fine del suo attributo periodo è antecedente al giorno corrente. Inoltre, (o, p) è istanza della relationship **utPren**.

postcondizioni: Il livello estensionale dei dati viene modificato come segue:

- se non esiste, viene creata una istanza (o, p) della relationship **feedback**
- il valore dell'attributo suOspite dell'istanza (o, p) della relationship **feedback** viene assegnato a f.

Non viene restituito alcun valore.

Specifica Use-Case **RicercaSistemazioni**

- **postiDisp(a: Abitazione, p: Periodo) : intero ≥ 0**

precondizioni: nessuna

postcondizioni: Il livello estensionale dei dati non viene modificato.

Se l'abitazione a è indisponibile nel periodo p , ovvero se esiste un valore i per l'attributo **indisp** dell'istanza a che si sovrappone a p , allora viene restituito il valore 0.

Altrimenti, sia S l'insieme delle istanze **sist** dell'entità **Sistemazione** tali che:

- esiste una istanza st dell'entità **Stanza** tale che:
 - * (st, a) è istanza della relationship **inAbit**
 - * $(sist, st)$ è istanza della relationship **inStanza**
- non esiste un'istanza **pren** dell'entità **Prenotazione** tale che:
 - * $(pren, sist)$ è istanza della relationship **prSist**
 - * il valore dell'attributo **periodo** di **pren** si sovrappone a p .

Viene restituita la somma dei valori dell'attributo **posti** su tutti gli elementi dell'insieme S .

- **perCitta(c: Citta, p: Periodo, n: intero > 0) : Abitazione (0,N)**
(Req. 8.2.)

precondizioni: nessuna

postcondizioni: Il livello estensionale dei dati non viene modificato.

Viene restituito l'insieme delle istanze **a** dell'entità **Abitazione** tali che:

- (a, c) è istanza della relationship **inCitta**
- **RicercaSistemazioni.postiDisp** (a, p) $\geq n$.

Specifica Use-Case **StrumentiOspiti**

- prenotazione(...) (**Req. 8.3.**)
 - precondizioni: ... (per esercizio)
 - postcondizioni: ... (per esercizio)
- feedbackSuOspitante(...) (**Req. 8.5.**)
 - precondizioni: ... (per esercizio)
 - postcondizioni: ... (per esercizio)



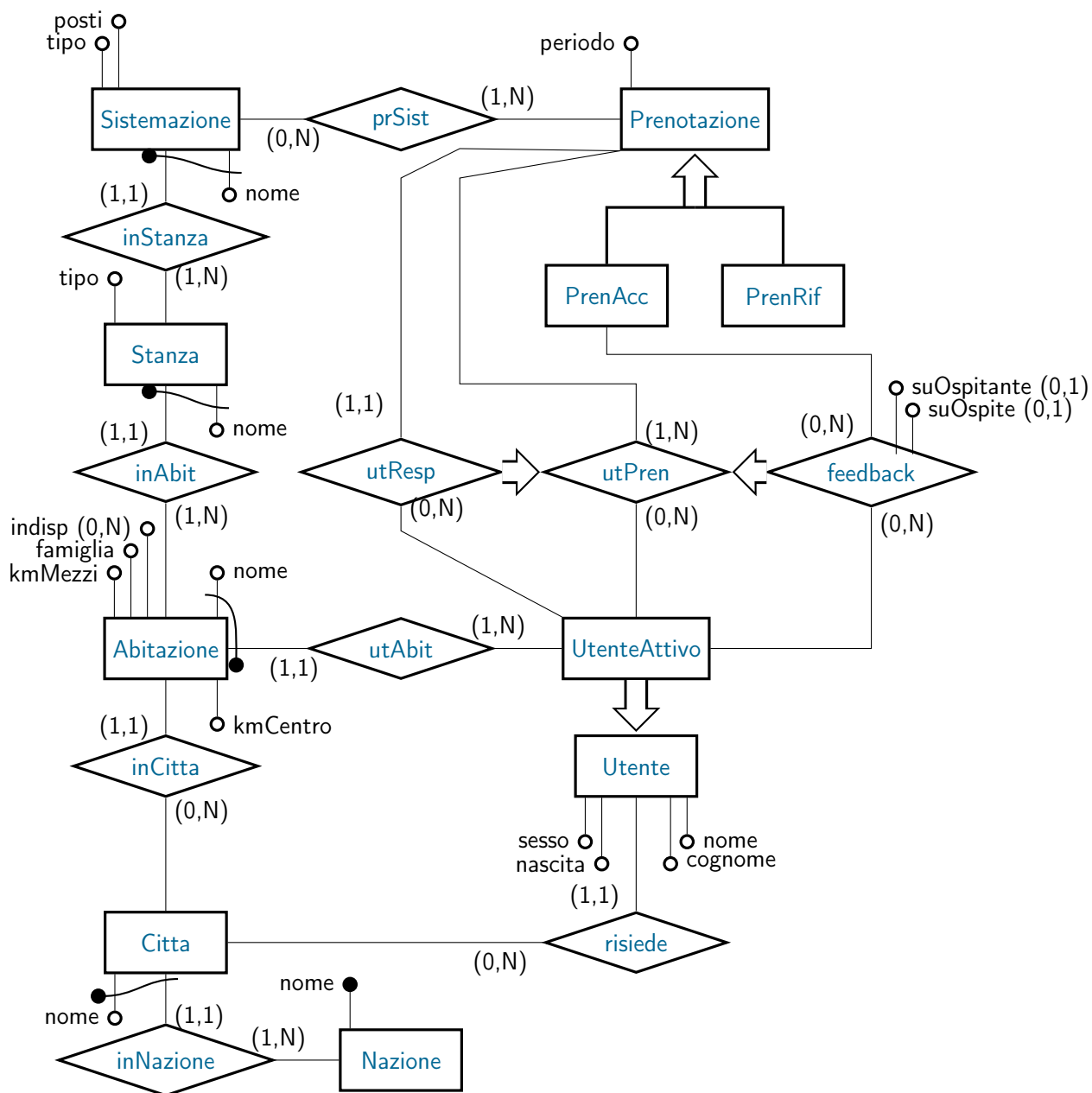
A.7

Vincoli Esterni

A.7.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti formalizzando, in logica del primo ordine, i vincoli esterni già definiti informalmente.

A.7.2 Soluzione



Specifiche dei Dati

Entità **Utente**

Ogni istanza di questa entità rappresenta un utente registrato ([Req. 1.](#)). Le istanze di questa entità potrebbero non essere coinvolte in istanze della relationship **utAbit**

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome dell'utente (Req. 1.1.)
cognome	stringa		Il cognome dell'utente (Req. 1.2.)
sex	{M,F}		Il sesso dell'utente (Req. 1.3.)
nascita	intero > 0		L'anno di nascita dell'utente (Req. 1.4.)

Entità **UtenteAttivo**

Ogni istanza di questa entità rappresenta un utente attivo, che ha reso disponibile almeno una sua **Abitazione**, e che quindi può effettuare e ricevere prenotazioni ([Req. 1.](#))

Attributi: Nessuno

Entità **Abitazione**

Ogni istanza di questa entità rappresenta una abitazione censita dal sistema ([Req. 2.](#))

attributo	dominio	molteplicità	descrizione
kmCentro	reale ≥ 0		La distanza dal centro città (in Km) (Req. 2.2.)
kmMezzi	reale ≥ 0		La distanza dalla stazione/fermata più vicina del sistema di trasporto pubblico (in Km) (Req. 2.3.)
famiglia	Famiglia		Il numero di adulti e bambini che vivono stabilmente nell'abitazione (Req. 2.4.)
indisp	Periodo	(0,N)	I periodi di indisponibilità del proprietario di questa abitazione (Req. 5.)
nome	stringa		Il nome dell'abitazione

Entità Stanza

Ogni istanza di questa entità rappresenta una stanza di una abitazione censita dal sistema (Req. 3.)

attributo	dominio	molteplicità	descrizione
tipo	{privata, comune}		Il tipo della stanza (Req. 3.2.)
nome	stringa		Il nome della stanza

Entità Sistemazione

Ogni istanza di questa entità rappresenta una sistemazione censita dal sistema (Req. 4.)

attributo	dominio	molteplicità	descrizione
tipo	{divano, letto}		La tipologia della sistemazione (Req. 4.2.)
posti	intero > 0		Il numero di posti della sistemazione (Req. 4.3.)
nome	stringa		Il nome della sistemazione

Entità Città

Ogni istanza di questa entità rappresenta una città

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della città

Entità Nazione

Ogni istanza di questa entità rappresenta una nazione

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della nazione

Entità Prenotazione

Ogni istanza di questa entità rappresenta una prenotazione (Req. 6.)

attributo	dominio	molteplicità	descrizione
periodo	Periodo		Il periodo della prenotazione (Req. 6.1.1.)

Vincoli:

[V.Prenotazione.ospitante] Le **Sistemazioni** relative ad ogni singola **Prenotazione** sono tutte della stessa **Abitazione** (e quindi offerte dallo stesso **Utente** ospitante).

Formalmente:

$$\begin{aligned} \forall p, \text{sist}', \text{sist}'', \text{st}', \text{st}'', \text{ab}' \\ \text{Prenotazione}(p) \wedge \text{Sistemazione}(\text{sist}') \wedge \text{Sistemazione}(\text{sist}'') \wedge \\ \text{Stanza}(\text{st}') \wedge \text{Stanza}(\text{st}'') \wedge \text{Abitazione}(\text{ab}') \wedge \\ \text{prSist}(p, \text{sist}') \wedge \text{prSist}(p, \text{sist}'') \wedge \text{inStanza}(\text{sist}', \text{st}') \wedge \\ \text{inStanza}(\text{sist}'', \text{st}'') \wedge \text{inAbit}(\text{st}', \text{ab}') \\ \rightarrow \text{inAbit}(\text{st}'', \text{ab}') \end{aligned}$$

[V.Prenotazione.posti] Il numero totale di posti delle **Sistemazioni** associate ad ogni singola **Prenotazione** è pari o superiore al numero di **Utenti** ospitati.

Formalmente, si definiscano i seguenti insiemi parametrici rispetto ad una istanza p dell'entità **Prenotazione**:

$$\begin{aligned} S_p &= \{(\text{sist}, \text{pos}) \mid \text{Sistemazione}(\text{sist}) \wedge \text{prSist}(p, \text{sist}) \wedge \text{posti}(\text{sist}, \text{pos})\} \\ U_p &= \{u \mid \text{Utente}(u) \wedge \text{utPren}(u, p)\} \end{aligned}$$

La seguente formula deve essere vera:

$$\forall p \text{ Prenotazione}(p) \rightarrow \sum_{(\text{sist}, \text{pos}) \in S_p} \text{pos} \geq |U_p|$$

Entità **PrenAcc**

Ogni istanza di questa entità rappresenta una prenotazione accettata

Attributi: Nessuno

Entità **PrenRif**

Ogni istanza di questa entità rappresenta una prenotazione rifiutata

Attributi: Nessuno

Relationship **inNazione**

Ogni istanza di questa relationship lega una **Città** alla **Nazione** a cui appartiene

Attributi: Nessuno

Relationship inCitta

Ogni istanza di questa relationship lega una **Abitazione** alla **Città** in cui è sita (**Req. 2.1.**)

Attributi: Nessuno

Relationship risiede

Ogni istanza di questa relationship lega un **Utente** alla **Città** in cui risiede (**Req. 1.5.**)

Attributi: Nessuno

Relationship utAbit

Ogni istanza di questa relationship lega un **Utente Attivo** ad una **Abitazione** che mette a disposizione (**Req. 1.6.**)

Attributi: Nessuno

Relationship inStanza

Ogni istanza di questa relationship lega una **Sistemazione** alla **Stanza** a cui appartiene (**Req. 4.1.**)

Attributi: Nessuno

Relationship inAbit

Ogni istanza di questa relationship lega una **Stanza** all'**Abitazione** a cui appartiene (**Req. 2.5.**)

Attributi: Nessuno

Relationship prSist

Ogni istanza di questa relationship lega una **Prenotazione** ad una delle **Sistemazioni** alle quali è relativa (**Req. 6.1.2.**)

Attributi: Nessuno

Relationship utPren

Ogni istanza di questa relationship lega una **Prenotazione** ad uno degli **Utenti Attivi** partecipanti, ovvero l'**Utente Attivo** che l'ha effettuata oppure un **Utente Attivo** accompagnatore (**Req. 6.1.4.**)

Attributi: Nessuno

Relationship feedback

Ogni istanza di questa relationship lega una **Prenotazione** consumata ad un **Utente** per il quale esiste un feedback. L'**UtenteAttivo** può essere sia l'**UtenteAttivo** ospitante che uno degli **Utenti Attivi** ospitati

attributo	dominio	molteplicità	descrizione
suOspite	[0, 5]	(0,1)	Il feedback sull'utente ospitato da parte dell'utente ospitante (Req. 7.2.)
suOspitante	[0, 5]	(0,1)	Il feedback sull'utente ospitante da parte dell'utente ospitato (Req. 7.1.)

Relationship utResp

Ogni istanza di questa relationship lega una **Prenotazione** all'**Utente Attivo** che l'ha effettuata (Req. 6.1.3.)

Attributi: Nessuno

Dominio Famiglia

Il dominio è un record composto dai seguenti campi:

- adulti: intero > 0
- bambini: intero ≥ 0

Dominio Periodo

Il dominio è di tipo record, le cui istanze sono composte dai seguenti campi:

- da: data
- a: data

[V.Periodo.legale] Per ogni istanza del dominio, il valore del campo a non deve precedere il valore del campo da.

Formalmente:

$$\forall p, x, y \text{ Periodo}(p) \wedge da(p, x) \wedge a(p, y) \rightarrow y \geq x$$

A fine di decidere se due istanze del dominio Periodo rappresentano periodi sovrapposti nel tempo, verrà utilizzato il simbolo di predicato sovrapposto/2, definito per coppie di istanze del dominio come segue:

$$\begin{aligned} \forall p, x, y, p', x', y' \text{ Periodo}(p) \wedge da(p, x) \wedge a(p, y) \wedge \\ \text{Periodo}(p') \wedge da(p', x') \wedge a(p', y') \rightarrow \\ \neg \text{sovrapposti}(p, p') \leftrightarrow (y < x' \vee y' < x) \end{aligned}$$

A.8

Specifiche Concettuali degli Use-Case

A.8.1 Testo

Proseguire la fase di Analisi Concettuale dei requisiti formalizzando, in logica del primo ordine, le specifiche concettuali degli use-case già definite informalmente.

A.8.2 Soluzione

Specifica Use-Case **Registrazione**

- **registraUtente**(n: stringa, c: stringa, s: {M,F}, nasc : data, cit : Citta)
: Utente (**Req. 8.1.**)

precondizioni: nessuna

postcondizioni:

Modifica del livello estensionale dei dati: Il livello estensionale dei dati al termine dell'esecuzione della funzione differisce da quello di partenza come segue:

Variazioni nel dominio di interpretazione: un nuovo elemento α

Variazioni nelle ennuple di predicati: $Utente(\alpha)$, $nome(\alpha, n)$, $cognome(\alpha, c)$, $sexso(\alpha, s)$, $nascita(\alpha, nasc)$, $risiede(\alpha, cit)$.

Valore di ritorno: $result = \alpha$

Specifica Use-Case **Profilo**

- **nuovaAbitazione**(u : Utente, n : stringa, kmM : reale ≥ 0 , kmC : reale ≥ 0 , fam : Famiglia, cit : Città) : **Abitazione** (**Req. 8.1.**)

precondizioni: nessuna

postcondizioni:

Modifica del livello estensionale dei dati: Il livello estensionale dei dati al termine dell'esecuzione della funzione differisce da quello di partenza come segue:

Variazioni nel dominio di interpretazione: un nuovo elemento α

Variazioni nelle ennuple di predicati: $Abitazione(\alpha)$, $nome(\alpha, n)$, $kmMezzi(\alpha, kmM)$, $kmCentro(\alpha, kmC)$, $famiglia(\alpha, fam)$, $utAbit(\alpha, u)$, $inCitta(\alpha, cit)$.

Valore di ritorno: $result = \alpha$

- **nuovaStanza**(a : **Abitazione**, ...) : **Stanza** (**Req. 8.1.**)

precondizioni: ... (per esercizio)

postcondizioni: ... (per esercizio)

- **nuovaSistemazione**(st : **Stanza**, ...) : **Sistemazione** (**Req. 8.1.**)

precondizioni: ... (per esercizio)

postcondizioni: ... (per esercizio)

Nota: i vincoli di molteplicità (1,N) nelle relationship **utAbit**, **inAbit**, **inStanza** impongono che, contestualmente all'invocazione della funzione **nuovaAbitazione()**, vengano invocate anche le funzioni **nuovaStanza()** e **nuovaSistemazione()**. Questo vuol dire che il sistema realizzato dovrà consentire l'inserimento di una sistemazione e di una stanza all'atto dell'inserimento di una abitazione.

Specifica Use-Case **StrumentiOspitante**

- **nuovaIndisp(a : Abitazione, p: Periodo) (Req. 8.1.)**

precondizioni: nessuna

postcondizioni:

Modifica del livello estensionale dei dati: Il livello estensionale dei dati al termine dell'esecuzione della funzione differisce da quello di partenza come segue:

Variazioni nel dominio di interpretazione: nessuna

Variazioni nelle ennuple di predicati: $\text{indisp}(a, p)$.

Valore di ritorno: nessuno

- **cancellaIndisp(a : Abitazione, p: Periodo) (Req. 8.1.)**

precondizioni: $\text{indisp}(a, p)$.

postcondizioni:

Modifica del livello estensionale dei dati: Il livello estensionale dei dati al termine dell'esecuzione della funzione differisce da quello di partenza come segue:

Variazioni nel dominio di interpretazione: nessuna

Variazioni nelle ennuple di predicati: $\neg \text{indisp}(a, p)$.

Valore di ritorno: nessuno

- **approvaRichiestaPren(p : Prenotazione) (Req. 8.4.)**

precondizioni: $\neg \text{PrenAcc}(p) \wedge \neg \text{PrenRif}(p)$.

postcondizioni:

Modifica del livello estensionale dei dati: Il livello estensionale dei dati al termine dell'esecuzione della funzione differisce da quello di partenza come segue:

Variazioni nel dominio di interpretazione: nessuna

Variazioni nelle ennuple di predicati: $\text{PrenAcc}(p)$.

Valore di ritorno: nessuno

- **rifiutaRichiestaPren(p : Prenotazione, m: stringa) (Req. 8.4.)**

precondizioni: $\neg \text{PrenAcc}(p) \wedge \neg \text{PrenRif}(p)$.

postcondizioni:

Modifica del livello estensionale dei dati: Il livello estensionale dei dati al termine dell'esecuzione della funzione differisce da quello di partenza come segue:

Variazioni nel dominio di interpretazione: nessuna

Variazioni nelle ennuple di predicati: PrenRif(p), motivo(p, m).

Valore di ritorno: nessuno

- feedbackSuOspite(p : Prenotazione, o: UtenteAttivo, f: [0,5]) (**Req. 8.6.**)

precondizioni: La prenotazione p deve essere accettata e già consumata, e l'utente o deve essere tra gli utenti ospitati relativamente alla prenotazione p.

Formalmente:

$$\begin{aligned} & \text{PrenAcc}(p) \wedge \text{utPren}(o, p) \wedge \\ & (\forall \text{per}, f, \text{oggi} \text{ periodo}(p, \text{per}) \wedge \text{fine}(\text{per}, f) \wedge \text{giorno}(\text{adesso}, \text{oggi}) \rightarrow \\ & f < \text{oggi}) \end{aligned}$$

postcondizioni:

Modifica del livello estensionale dei dati: Il livello estensionale dei dati al termine dell'esecuzione della funzione differisce da quello di partenza come segue:

Variazioni nel dominio di interpretazione: nessuna

Variazioni nelle ennuple di predicati: feedback(o, p), suOspite(o, p, f).

Valore di ritorno: nessuno

Specifica Use-Case **RicercaSistemazioni**

- **postiDisp(a: Abitazione, p: Periodo) : intero ≥ 0**

precondizioni: nessuna

postcondizioni:

Modifica del livello estensionale dei dati: nessuna

Valore di ritorno: Sia S l'insieme delle coppie (sist, n) dove sist è una sistemazione dell'abitazione a non prenotata e non indisponibile per tutto il periodo p ed n è il numero di posti di sist:

$$S = \left\{ (sist, n) \mid \begin{array}{l} \exists st \text{ inStanza}(sist, st) \wedge \text{inAbit}(st, a) \wedge \text{posti}(sist, n) \wedge \\ \neg [\exists \text{pren}, p' \text{ Prenotazione}(\text{pren}) \wedge \text{prSist}(\text{pren}, sist) \wedge \\ \text{periodo}(\text{pren}, p') \wedge \text{sovrapposti}(p, p')] \wedge \\ \neg [\exists ind \text{ indisp}(a, ind) \wedge \text{sovrapposti}(p, ind)] \end{array} \right\}.$$

$$\text{result} = \sum_{(sist, n) \in S} n.$$

- **perCitta(c: Citta, p: Periodo, n: intero > 0) : Abitazione (0,N)**
(Req. 8.2.)

precondizioni: nessuna

postcondizioni:

Modifica del livello estensionale dei dati: nessuna

Valore di ritorno:

$$\text{result} = \left\{ a \mid \begin{array}{l} \text{Abitazione}(a) \wedge \text{inCitta}(a, c) \wedge \\ \text{RicercaSistemazioni.postiDisp}(a, p) \geq n. \end{array} \right\}$$

Specifica Use-Case **StrumentiOspiti**

- prenotazione(...) (**Req. 8.3.**)
 - precondizioni: ... (per esercizio)
 - postcondizioni: ... (per esercizio)
- feedbackSuOspitante(...) (**Req. 8.5.**)
 - precondizioni: ... (per esercizio)
 - postcondizioni: ... (per esercizio)