

Raffinamento dei Requisiti:

0. Recapito:

- 0.1. indirizzo :
 - 0.1.1. Via : stringa
 - 0.1.2. Civico : stringa
 - 0.1.3. Cap : intero di 5 numeri
- 0.2. telefono : stringa
- 0.3. email : stringa

1. Anagrafica:

- 1.1. nome : stringa
- 1.2. cognome : stringa
- 1.3. data di nascita : data

2. Pazienti interni:

- 2.1. anagrafica (v. req. 0)
- 2.2. recapito : (v. req. 0)
- 2.3. data ricovero : se e solo se disponibilità di posti letto ha esito positivo
- 2.4. stanza del ricovero (v. req. 4) : non potrà cambiare
- 2.5. inizio ricovero : intero < fine ricovero
- 2.6. fine ricovero : intero > inizio ricovero

3. Medici:

- 3.1. anagrafica (v. req. 0)
- 3.2. Pazienti in cura (v. req. 2)
- 3.3. Pazienti esterni per prestazione (. req. 7)
- 3.4. specializzazioni:
 - 3.4.1. primaria : stringa 1..1
 - 3.4.2. secondarie : stringa 0..*

4. Stanze:

- 4.1. numero posti letto : intero compreso tra 1 e 8 compreso
- 4.2. numero piano : intero ≥ 0
- 4.3. numero settore : intero ≥ 0

5. storico dei pazienti:

- 5.1. pazienti (v. req. 2)

6. Giro delle visite:

- 6.1. medici (v. req. 3)
- 6.2. ordine del giro : un insieme del numero stanza e numero settore in ordine crescente [(1, 1), (1, 3), (3, 4), (7, 1), (7, 7)]
 - 6.2.1. numero stanza
 - 6.2.2. numero settore

7. Pazienti Esterne:

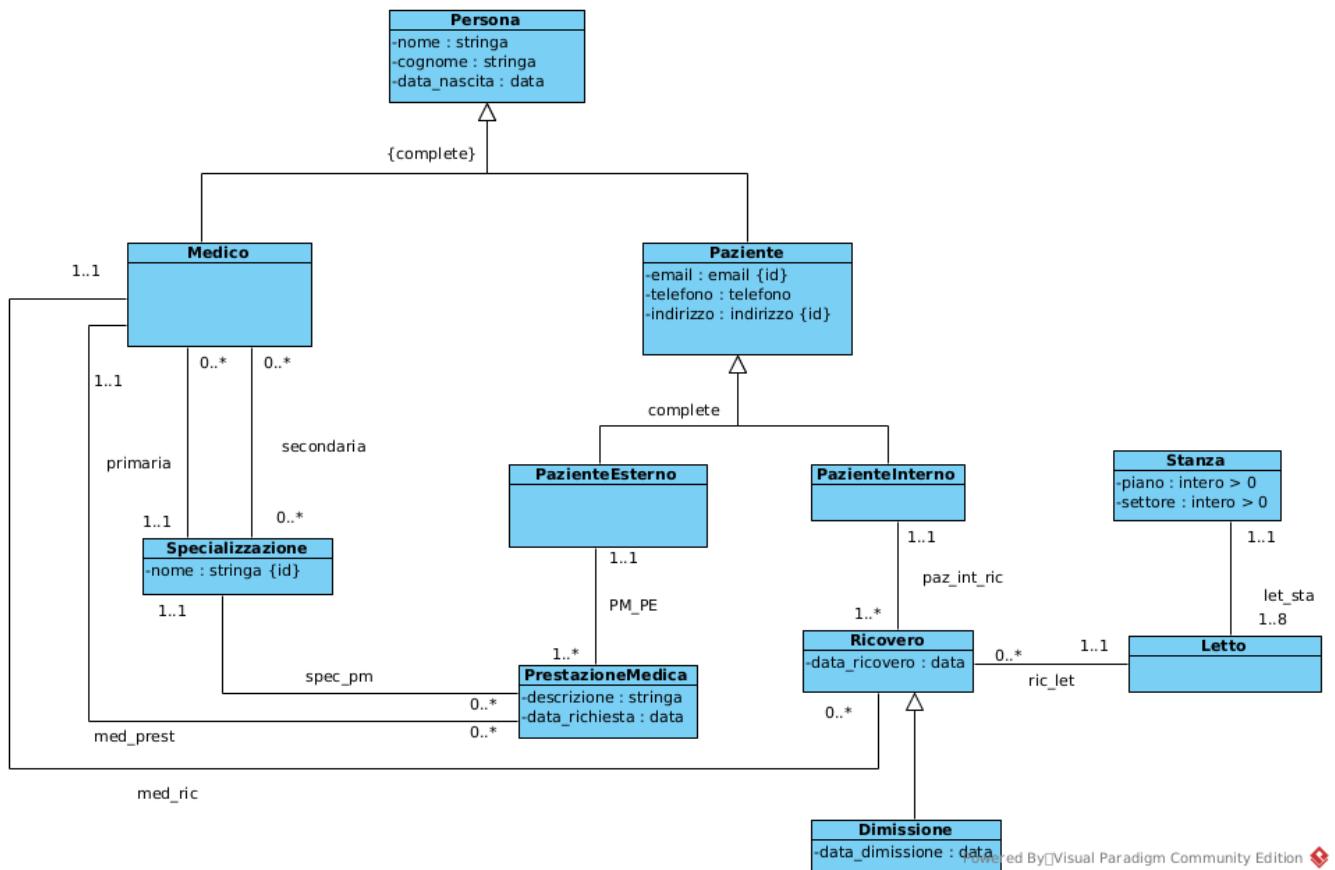
- 7.1. anagrafica (v. req. 1)
- 7.2. prestazione medica : ortopedica, dermatologica ect...

- 7.2.1 data prestazione
- 7.2.1 descrizione : stringa

999. Il sistema gestisce anche:

- 999.1. prestazioni mediche fatte da medici dell'ospedale a pazienti esterni.
- 999.2. il sistema deve restituire l'insieme dei medici maggiormente idonei a soddisfare la scelta del paziente della specializzazione del medico. Il criterio di idoneità è il seguente:
 - - se esistono medici con specializzazione primaria pari ad s, il risultato è l'insieme di tali medici.
 - - Altrimenti, il risultato è l'insieme dei medici che hanno s tra le loro specializzazioni secondarie.
- 999.3. Il sistema QuickHospital è accessibile ai medici, al personale amministrativo e a quello dell'ufficio prenotazioni

Diagramma UML concettuale:



Specifica dei tipi di dati:

- indirizzo : tipo record con campi:
 - via : stringa
 - civico : stringa
 - cap : stringa di 5 caratteri alfanumerici
- telefono : stringa $\wedge (\dagger \dagger ? \dagger \dagger \{1,4\})? (\dagger \dagger ? [(\dagger \dagger)-\dagger \dagger \dagger] \{3,15\})$$
- email : stringa $\wedge [a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]\{2,\}$$

Specifica delle classi e vincoli:

Operazioni ausiliari

- sovrapposti(i1 : Data, f1 : Data, i2 : Data, f2 : Data) : Booleano:
- precondizioni:
 - deve essere vero che $i1 \leq f1$ e che $i2 \leq f2$
- postcondizioni:
 - L'operazione non modifica il livello estensionale
 - result = true se $i1 \leq f2$ oppure $i2 \leq f1$ altrimenti false

Specifica della classe Persona:

[V.Paziente.non_ricovero_se_stesso] Un paziente non può occuparsi del ricovero di se stesso:

- Per ogni pers:Persona, non deve esistere un ric:Ricovero tale che:
 - sia med:Medico, tale che (ric, med):ric_med e che med = pers
 - sia paz:Interno, tale che (paz, ric):paz_ric e che paz = pers

[V.Paziente.non_prestazione_a_se_stesso] Un paziente non può eseguire una prestazione a se stesso:

- Per ogni pers:Persona, non deve esistere una prest:Prestazione tale che:
 - sia med:Medico, tale che (prest, med):prest_med e che med = pers
 - e che sia paz:Esterno, tale che (paz, ric):paz_ric e che paz = pers

Specifica della classe Paziente:

[V.Paziente.non_ricoverato_più_volte_stessa_data] Un paziente non può essere ricoverato un'altra volta mentre è già ricoverato

- Per ogni p:Interno, sia R l'insieme di tutti i r:Ricovero, tali che (p, r):paz_ric.
- Non devono esistere r' e r'' tali che $r' \neq r''$ e che sovrapposti(r'.data_ricovero, r'.data_dimiss, r''.data_ricovero, r''.data_dimiss)

Specifica della classe Ricovero:

[V.Ricovero.data_valida] La data di ricovero deve venire prima della data di dimissione

- Per ogni r:Ricovero, deve essere vero che $r.data_ricovero \leq r.data_dimiss$

Specifica della classe PostoLetto:

[V.PostoLetto.posto_leotto_non_occupato] due ricoveri non possono esser fatti sullo stesso posto letto alla stessa data

- Per ogni pl:PostoLetto, sia R l'insieme di tutti i r:Ricovero, tali che (r, pl):ric_pl.
- Non devono esistere r' e r" tali che r' != r" e che sovrapposti(r'.data_ricovero, r'.data_dimiss, r".data_ricovero, r".data_dimiss)
- disponibili(d : Data) : PostoLetto[0..*]:
- precondizioni:
 - Nessuna
- postcondizioni:
 - L'operazione non modifica il livello estensionale
 - Sia PL l'insieme di tutti i pl:PostoLetto, tali che:
 - Sia R l'insieme di tutti i r:Ricovero tali che (r, pl):ric_pl
 - Non esista un r in R tale che sovrapposti(r.data_ricovero, r.data_dimiss, d, d)
 - result = PL

Specifica della classe Prestazione:

[V.Prestazione.medico_ha_specializzazione_richiesta] Il medico deve avere la specializzazione richiesta dalla prestazione

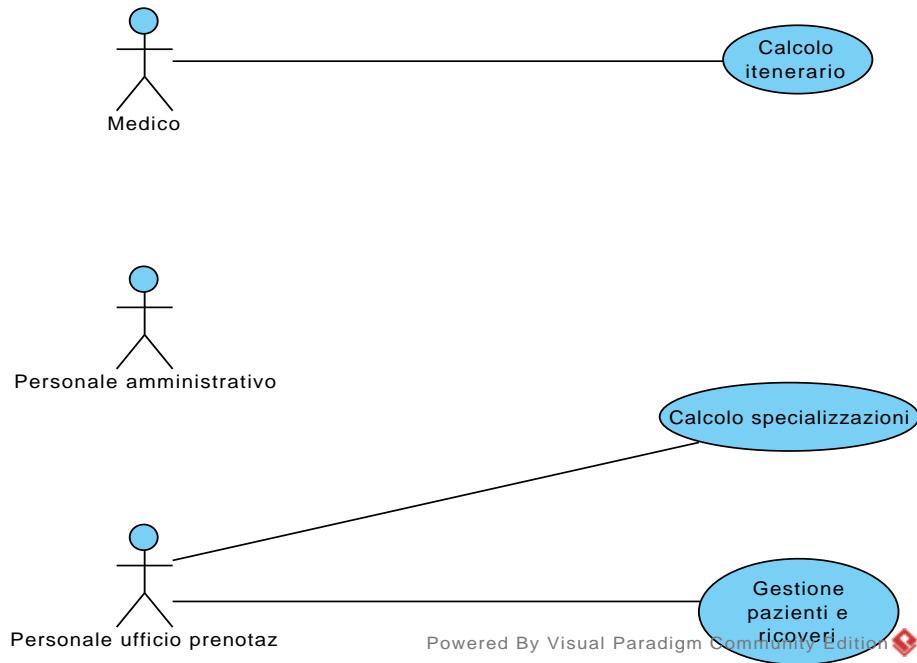
- Per ogni prest:Prestazione, sia med:Medico tale che (prest, med):prest_med e sia spec:Specializzazione tale che (prest, spec):prest_spec,
- deve esistere il link (spec, med):spec_primaria o il link (spec, med):spec_secondaria

Specifica della classe Medico:

[V.Medico.specializzazione_medico_coerenti] Il medico non può avere una specializzazione sia come secondaria che come primaria

- Per ogni med:Medico, non deve esistere una spec:Specializzazione tale che esistono entrambi i link (med, spec):spec_primaria e (med, spec):spec_secondaria

Diagramma Use-Case:



Specifiche di Use-Case:

[Calcolo itinerario]

calcolo_itinerario(med: Medico) : {Intero >= 0, Intero >= 0} :

- precondizioni:

 - Nessuna

- postcondizioni:

- Sia R l'insieme di tutti i ric:Ricovero, tali che (ric, med):med_ric
- Per ogni ric in R, sia PL l'insieme di tutti i pl:PostoLetto, tali che (ric, pl):ric_pl
- Sia ST l'insieme di tutte le st:Stanza, tali che st partecipa al link dell'associazione con un pl in PL

- Sia Res l'insieme delle coppie (p, s), dove per ogni st:Stanza in ST:

- p è l'attributo st.piano
- s è l'attributo st.settore

- result = Res

Powered By Visual Paradigm Community Edition

[Calcolo specializzazioni:]

calcolo_specializzazioni(s: Specializzazione) : Medico[0..*]:

- precondizioni:

- Nessuna

- postcondizioni

- Sia M_primaria l'insieme di tutti i med:Medico, tale che (s, med):spec_primaria
- Sia M_secondaria l'insieme di tutti i med:Medico, tale che (s, med):spec_secondaria
- Se M_primaria è un insieme vuoto allora result = M_secondaria altrimenti result = M_primaria

[Gestione pazienti e ricoveri:]

registrazione_ricovero(data_ricovero : Data, int : Interno, med : Medico, pl : PostoLetto) :

- precondizioni:

- pl deve essere presente nell'insieme pl.disponibili(data_ricovero)

- postcondizioni:

- viene creati e restituiti i seguenti link/oggetti:
 - ric:Ricovero, tale che:
 - ric.data_ricovero = data_ricovero
 - (ric, pl):ric_pl
 - (int, ric):paz_ric
 - (ric, med):ric_med

registra_dimissione(ric : Ricovero, data_dimissione : Data) : Data:

- precondizioni:

- Nessuna

- postcondizioni:

- viene creato e restituito l'attributo ric.data_dimissione = data_dimissione

registra_paziente(nome : Stringa, cognome : Stringa, data_nasc : Data, email : Email [0..*], tel : NumTel[0..*], recapito : Indirizzo, prest : Prestazione[0..1]) :

- precondizioni:

- Vengono creati e restituiti i seguenti link/oggetti:

- paz:Paziente, tale che:
 - paz.nome = nome
 - paz.cognome = cognome
 - paz.data_nasc = data_nasc
 - paz.email = email
 - paz.tel = tel
 - paz.recapito = recapito
- (paz, prest):paz_prest