

Travel to the moon

1 Requisiti

I dati di interesse per il sistema sono

1. Requisiti sulle **crociere**:

- 1.1. codice
- 1.2. data di inizio
- 1.3. data di fine
- 1.4. nave utilizzata (v. req. 2)
- 1.5. itinerario (v. req. 4)
- 1.6. il tipo, uno tra:
 - 1.6.1. luna di miele, di cui interessa:
 - 1.6.1.1. sottotipo, uno tra:
 - 1.6.1.1.1. tradizionali
 - 1.6.1.1.2. alternative
 - 1.6.2. per famiglie, di cui interessa:
 - 1.6.3. se adatte ai bambini (booleano)

2. Requisiti sulle **navi**:

- 2.1. nome
- 2.2. comfort (3..5)
- 2.3. capienza

3. Requisiti sulle **destinazioni**:

- 3.1. nome
- 3.2. continente
- 3.3. posti da vedere (v. req. 5)
- 3.4. tipo, almeno uno tra:
 - 3.4.1. romantico
 - 3.4.2. divertente

4. Requisiti sugli **itinerari**:

- 4.1. sequenza ordinata di elementi, di cui interessa:
 - 4.1.1. porto (v. req. 3)
 - 4.1.2. arrivo:
 - 4.1.2.1. il numero d'ordine del giorno (rispetto alla data di inizio della crociera)
 - 4.1.2.2. ora
 - 4.1.3. ripartenza
 - 4.1.3.1. il numero d'ordine del giorno (rispetto alla data di inizio della crociera)

4.1.3.2. ora

5. Requisiti sui **posti da vedere**:

- 5.1. nome
- 5.2. descrizione
- 5.3. orari di apertura, nella forma di una mappa che associa ad ogni giorno della settimana (lunedì, ..., domenica) un insieme di fasce orarie, dove ogni fascia oraria è definita in termini di una coppia di orari

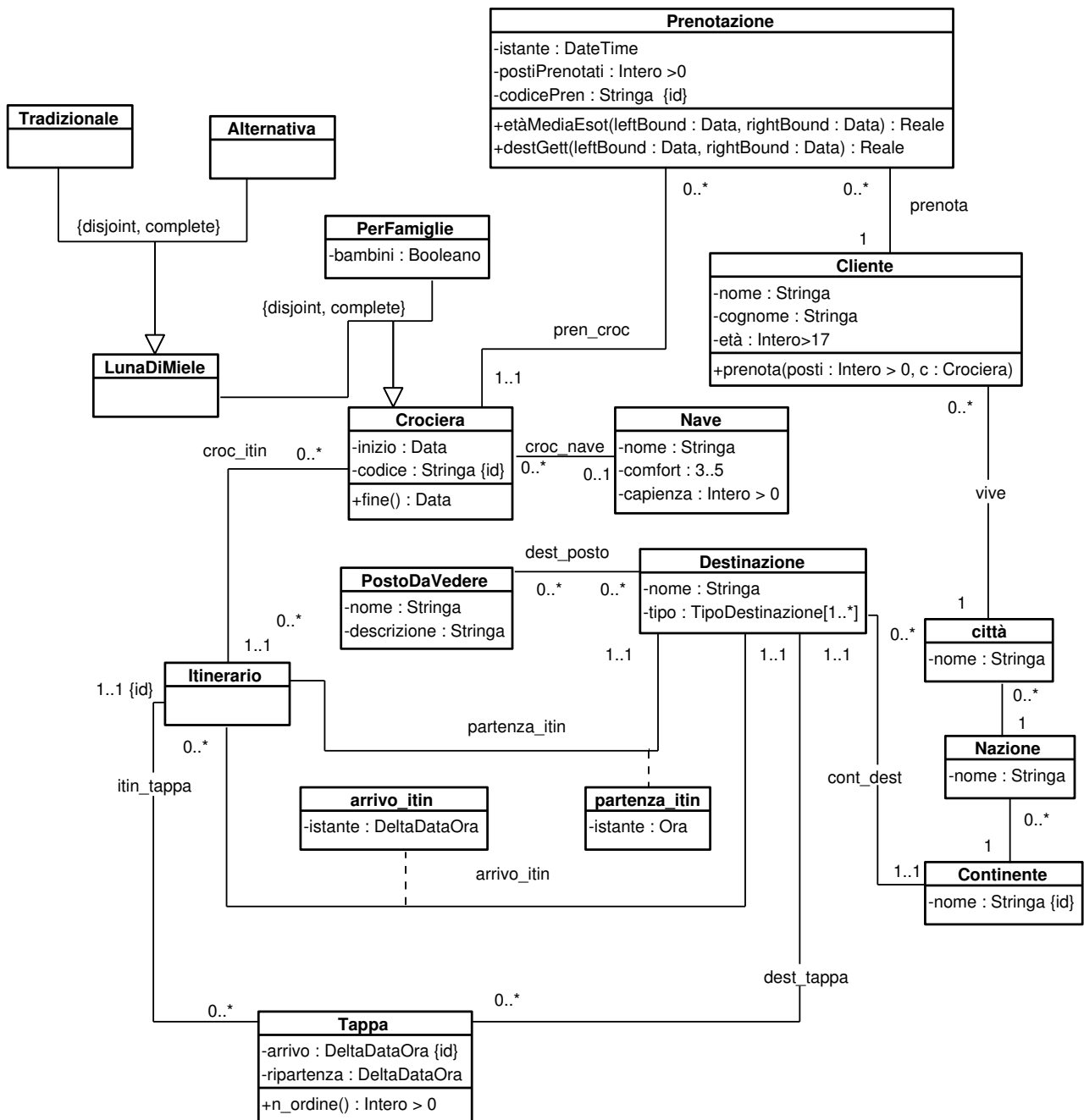
6. Requisiti sulle **prenotazioni**:

- 6.1 data ed ora della prenotazione.
- 6.2 cliente coinvolto.
- 6.3 crociera coinvolta
- 6.4 numero posti prenotati

7. Requisiti sui **clienti**:

- 7.1 nome.
- 7.2 cognome.
- 7.3 età.
- 7.4 indirizzo.

2 UML



3 Specifiche

3.1 Specifica dei Tipi di Dato

TipoDestinazione = romantica, divertente

DeltaDataOra = (
giorno: Intero ≥ 0 ,
orario: Ora
)

Operazioni del tipo di dato DeltaDataOra:

$< (x:\text{DeltaDataOra}, y:\text{DeltaDataOra}) : \text{Boolano}$
pre: nessuna
postcondizioni:
result = true se e solo se:
(x.giorno \leq y.giorno)
oppure
(x.giorno = y.giorno e x.orario \leq y.orario)

3.2 Specifica delle Classi

3.2.1 Crociera

Operazione `fine()` : Data

- *pre-condizioni* : nessuna
- *post-condizioni* : Sia i:Itinerario tale che (this, i):croc_itin.

Sia x:DeltaDataOra il valore dell'attributo "istante" dell'unico link di assoc. arrivo_itin in cui "i" è coinvolto.

result = this.inizio + x.giorno "giorni".

Vincoli esterni :

[V.Crociera.date]

Per ogni oggetto c:Crociera deve essere: c.inizio \leq c.fine

[V.Crociera.posti]

Per ogni oggetto c:Crociera deve essere che

- sia R l'insieme delle relazioni (c,p) dove p :Prenotazione coinvolta con c:Crociera. Considero $n = \sum_{(c,p_i) \in R} p_i.\text{postiPrenotati}$
- sia $(c,nave)$ il link di tipo croc_nave e sia nave:Nave.

Deve essere che $n \leq \text{nave.capienza}$.

3.2.2 Itinerario

Vincoli esterni :

[V.Itinerario.arrivo_dopo_ultima_tappa]

Per ogni i :Itinerario siano:

- T l'insieme delle istanze 't' di classe Tappa tali che $(i,t):itin_tappa$
- x il valore dell'attributo 'istante' dell'unico link di associazione $arrivo_itin$ in cui 'i' è coinvolto

Per ogni $t:T$, deve essere: $t.ripartenza < x$.

[V.Itinerario.prima_tappa_dopo_partenza]

Per ogni i :Itinerario siano:

- T l'insieme delle istanze 't' di classe Tappa tali che $(i,t):itin_tappa$ e $t.arrivo.giorno = 1$
- x il valore dell'attributo 'istante' dell'unico link di associazione $partenza_itin$ in cui 'i' è coinvolto

Per ogni $t:T$, deve essere $t.arrivo.orario > x$.

[V.Itinerario.arrivo_dopo_partenza_se_senza_tappe]

Per ogni i :Itinerario tale siano:

- T l'insieme delle istanze 't' di classe Tappa tali che $(i,t):itin_tappa$
- x il valore dell'attributo 'istante' dell'unico link di associazione $partenza_itin$ in cui 'i' è coinvolto
- y il valore dell'attributo 'istante' dell'unico link di associazione $arrivo_itin$ in cui 'i' è coinvolto

se $T = \text{vuoto}$ e $y.giorno = 1$, allora deve essere $x < y.orario$.

Per ogni $t:T$, deve essere $t.arrivo.orario > x$.

3.2.3 Tappa

Operazione `n_ordine() : Int > 0`

- *pre-condizioni* : nessuna
- *post-condizioni* : No side-effect. Sia i l'itinerario della tappa `this`; Sia `TappePrecedenti` l'insieme degli oggetti $t:Tappa$ tali che:

– $(i,t): itin_tappa$

– $t.arrivo < this.arrivo$;

$result = |TappePrecedenti| + 1$.

Vincoli esterni :

[V.Tappa.date]

Per ogni istanza $t:Tappa$, deve essere: $t.arrivo < t.ripartenza$

3.2.4 Prenotazioni

Vincoli esterni :

Operazione : `etàMediaEsot(leftBound : Data, rightBound : Data) : Reale`

- *pre-condizioni* : $\text{leftBound} < \text{rightBound}$.
- *post-condizioni* : Si consideri P_1 l'insieme delle prenotazioni $p:\text{Prenotazione}$ tale che $\text{leftBound} \leq p.\text{istante}$ *le* rightBound .

Si consideri P_2 l'insieme delle prenotazioni $p:\text{Prenotazione}$, per cui, vi è almeno un link (p,c) con $c:\text{Crociera}$, per cui tale crociera è coinvolta in almeno un link (c,i) con $i:\text{Itinerario}$, per cui tale itinerario è coinvolto in almeno un link (i,d) con $d:\text{Destinazione}$, per cui tale destinazione è coinvolta in almeno un link (d,c) con $c:\text{Continente}$, per cui $c.\text{Stringa} \neq \text{"Europa"}$.

Si consideri $P_1 \cap P_2$, si consideri C l'insieme dei clienti $c:\text{Cliente}$ per cui esiste un link (c,p) dove $p \in P$. Si consideri $n = \sum_{c \in C} c.\text{età}$. $\text{result} = \frac{n}{|C|}$.

Operazione : `destGett(leftBound : Data, rightBound : Data) : Reale`

- *pre-condizioni* : $\text{leftBound} < \text{rightBound}$. Esiste almeno un oggetto di tipo $p:\text{Prenotazione}$ per cui $\text{leftBound} \leq p.\text{istante}$ *le* rightBound .
- *post-condizioni* : Sia P l'insieme delle prenotazioni $p:\text{Prenotazione}$ tale che $\text{leftBound} \leq p.\text{istante}$ *le* rightBound . **DA CONTINUARE**

[V.Prenotazioni.date]

Per ogni istanza $p:\text{Prenotazione}$, ed ogni link r fra $p:\text{Prenotazione}$ e $c:\text{Crociera}$, deve essere: $p.\text{data_prenotazione} < c.\text{inizio}$

3.2.5 Cliente

Operazione : `prenota(posti:Interio > 0, c:Crociera)`

- *pre-condizioni* : sia R l'insieme delle relazioni (c,p) dove $p:\text{Prenotazione}$ coinvolta con $c:\text{Crociera}$. Considero $n = \sum_{(c,p_i) \in R} p_i.\text{postiPrenotati}$.

sia $(c,nave)$ il link di tipo `croc_nave` e sia $nave:Nave$, considero $free = nave.\text{capienza} - n$ (i posti liberi).

Deve essere che $\text{posti} \leq free$.

- *post-condizioni* : Viene creato un oggetto $p:\text{Prenotazione}$ in relazione con `this` ed in relazione con $c:\text{Crociera}$ (parametro in input), sarà che : $p.\text{istante} = \text{now}$, $p.\text{postiPrenotati} = \text{posti}$, $p.\text{codicePren} = \text{Stringa univoca randomica}$.