

Travel To The Moon

Indice

1	Specifiche del progetto	2
2	Raffinamento dei requisiti	3
3	Diagramma UML delle classi	6
4	Diagramma degli Use Case	7
5	Specifiche dei tipi di dati	8
6	Specifiche delle classi e delle associazioni	9
7	Specifiche dei vincoli esterni	28
8	Specifiche delle operazioni	30

1 Specifiche del progetto

I dati di interesse per il sistema sono le crociere offerte dall'agenzia con le relative prenotazioni e le destinazioni in catalogo. Il sistema deve essere in grado di rappresentare le crociere offerte dall'agenzia, con codice, date di inizio e fine, e la nave utilizzata. Delle navi, che hanno un nome (ad es. LoveBoat), interessa il grado di comfort, espresso in un numero di stelle che può variare da 3 a 5, e il numero massimo di passeggeri che possono ospitare.

Ciascuna crociera consta di un itinerario caratterizzato da un nome (ad es. Panorami d'Oriente) il quale prevede una sequenza ordinata di destinazioni. Di queste interessa il nome e il continente in cui si trovano. Gli itinerari fissano, oltre che l'ordine delle destinazioni da visitare, anche la relativa data ed ora di arrivo e di partenza. Dato che, in generale, un itinerario può essere previsto da più di una crociera, le date di arrivo e partenza relative ad una destinazione vengono espresse come differenze rispetto la data di inizio della crociera stessa (ad es., l'itinerario Panorami d'Oriente prevede di raggiungere la destinazione x alle 16:00 del quinto giorno di crociera, e di ripartire alle 12:00 del giorno successivo, il sesto).

Inoltre, le destinazioni sono caratterizzate da un insieme di posti da vedere durante eventuali escursioni organizzate. Questi ultimi sono caratterizzati dal nome, dalla descrizione, e dalla fascia oraria consigliata per le visite. Il sistema deve permettere di risalire ai posti da vedere in ogni singola destinazione.

L'agenzia classifica le crociere in crociere di luna di miele e crociere per famiglia (di queste ultime interessa conoscere se sono adatte o meno ai bambini), e le destinazioni in romantiche e divertenti. Si noti che possono esistere destinazioni che sono sia romantiche che divertenti. Per venire incontro alle nuove tendenze delle giovani coppie, le crociere di luna di miele vengono ulteriormente classificate in tradizionali e alternative: sono definite tradizionali quelle che prevedono un numero di destinazioni romantiche maggiore o uguale al numero di destinazioni divertenti, alternative le altre.

Infine, il sistema deve anche permettere di gestire le prenotazioni di crociere effettuate dai clienti. In particolare, dei clienti interessa nome, cognome, età ed indirizzo, mentre delle prenotazioni interessa l'istante di prenotazione, la crociera ed il numero di posti prenotati.

Le funzionalità richieste al sistema sono le seguenti:

1. Dato un cliente che desidera prenotare un certo numero di posti per una crociera c , il personale dell'Ufficio Prenotazioni deve poter effettuare la relativa prenotazione. La richiesta di prenotazione deve essere rifiutata nel caso il numero di posti disponibili, all'istante corrente, per la crociera c non sia sufficiente.
2. L'Ufficio Marketing deve poter calcolare l'età media dei clienti che hanno prenotato, in un dato periodo, almeno una crociera che prevede una destinazione esotica (ovvero che si trova in un continente diverso dall'Europa).
3. L'Ufficio Marketing deve poter calcolare la percentuale delle destinazioni da considerarsi gettonate in un periodo dato. Una destinazione va considerata gettonata in un certo periodo se è stata raggiunta, in quel periodo, da almeno dieci crociere di luna di miele, oppure da almeno quindici crociere per famiglie.

2 Raffinamento dei requisiti

In questa sezione verranno descritte le fasi di raffinamento dei requisiti.

1. Requisiti sulle [crociere](#)
 - 1.1. codice: [StringaS](#)
 - 1.2. data di inizio: [Data](#)
 - 1.3. data di fine: [Data](#)
 - 1.4. nave utilizzata ([v.req.2](#))
 - 1.5. itinerario ([v.req.4](#))
 - 1.6. Le prenotazioni ([v.req.7](#))
 - 1.7. il tipo, uno tra:
 - 1.7.1. [crociera di luna di miele](#), di cui interessa:
 - 1.7.1.1. [tradizionale](#)
 - 1.7.1.2. [alternativa](#)
 - 1.7.2. [crociera per famiglie](#)
 - 1.7.2.1. [adatta ai bambini](#): [booleano](#)
2. Requisiti sulle [navi](#)
 - 2.1. nome: [StringaS](#)
 - 2.2. grado di comfort (da 3 a 5 stelle): [ValutazioneNave](#)
 - 2.3. numero massimo di passeggeri: [InteroGZ](#)
 - 2.4. le crociere che fanno uso della nave ([v.req.1](#))
3. Requisiti sulle [destinazioni](#)
 - 3.1. nome: [StringaS](#)
 - 3.2. la città in cui si trova ([v.req.8](#))
 - 3.3. posti da vedere ([v.req.5](#))
 - 3.4. i porti dai quali può essere raggiunta ([v.req.11](#))
 - 3.5. Il tipo, uno o entrambe tra:
 - 3.5.1. [romantica](#)
 - 3.5.2. [divertente](#)
4. Requisiti sugli [itinerari](#)
 - 4.1. nome: [StringaS](#)
 - 4.2. le tappe ([v.req.13](#))
 - 4.3. destinazione di partenza ([v.req.3](#))
 - 4.3.1. ora di partenza: [DeltaOra](#)

- 4.4. destinazione di arrivo ([v.req.3](#))
 - 4.4.1. ora di arrivo: [DeltaOra](#)
 - 4.4.2. giorno di arrivo (rappresentato come differenza rispetto la data di inizio della crociera): [InteroGEZ](#)
- 5. Requisiti sui [posti da vedere](#)
 - 5.1. nome: [StringaS](#)
 - 5.2. descrizione: [StringaL](#)
 - 5.3. fascia oraria consigliata ([v.req.11](#))
 - 5.4. la città in cui si trovano ([v.req.8](#))
- 6. Requisiti sugli [utenti](#)
 - 6.1. nome: [StringaS](#)
 - 6.2. cognome: [StringaS](#)
 - 6.3. indirizzo: [StringaM](#)
 - 6.4. data di nascita: Data
 - 6.5. città di residenza ([v.req.8](#))
- 7. Requisiti sulle [prenotazioni](#)
 - 7.1. istante di prenotazione: Dataora
 - 7.2. crociera prenotata ([v.req.1](#))
 - 7.3. numero di posti prenotati: [InteroGZ](#)
 - 7.4. l'utente che ha effettuato la prenotazione ([v.req.6](#))
- 8. Requisiti sulle [città](#)
 - 8.1. nome: [StringaS](#)
 - 8.2. la nazione in cui si trova ([v.req.9](#))
- 9. Requisiti sulle [nazioni](#)
 - 9.1. nome: [StringaS](#)
 - 9.2. il continente in cui si trova ([v.req.10](#))
- 10. Requisiti sui [continenti](#)
 - 10.1. nome: [StringaS](#)
- 11. Requisiti sui [porti](#)
 - 11.1. nome: [StringaS](#)
 - 11.2. la destinazione (0..1) ([v.req.3](#))
 - 11.3. la città in cui si trova ([v.req.8](#))
- 12. Requisiti sugli [orari dei posti da vedere](#)

- 12.1. giorno: [Giorno](#)
- 12.2. ora di inizio: [DeltaOra](#)
- 12.3. ora di fine: [DeltaOra](#)
- 13. Requisiti delle [tappe degli itinerari](#)
 - 13.1. l'itinerario ([v.req.4](#))
 - 13.2. la destinazione ([v.req.3](#))
 - 13.3. il giorno di arrivo (rappresentato come differenza rispetto la data di inizio della crociera): [InteroGEZ](#)
 - 13.4. l'ora di arrivo: [DeltaOra](#)
 - 13.5. il giorno di partenza (rappresentato come differenza rispetto la data di inizio della crociera): [InteroGEZ](#)
 - 13.6. l'ora di partenza: [DeltaOra](#)

3 Diagramma UML delle classi

In questa sezione verrà mostrato il diagramma UML delle classi.

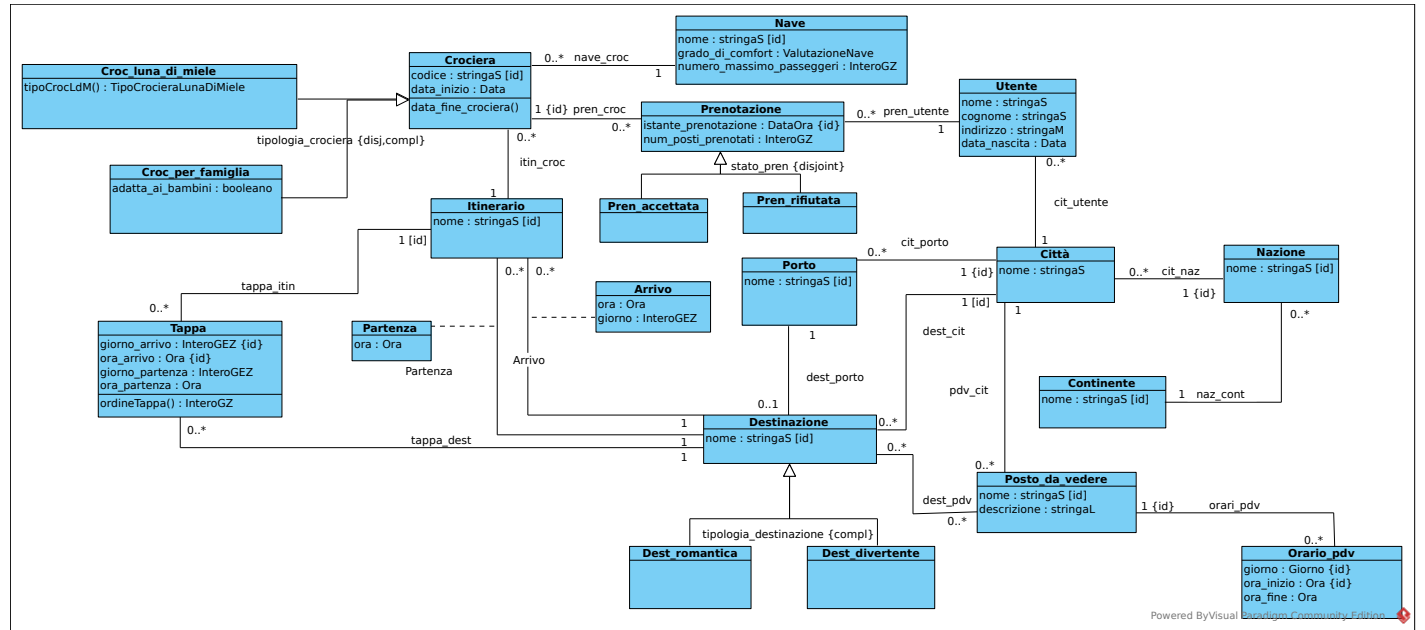


Figure 1: Diagramma delle classi

4 Diagramma degli Use Case

In questa sezione verrà mostrato il diagramma degli Use Case.

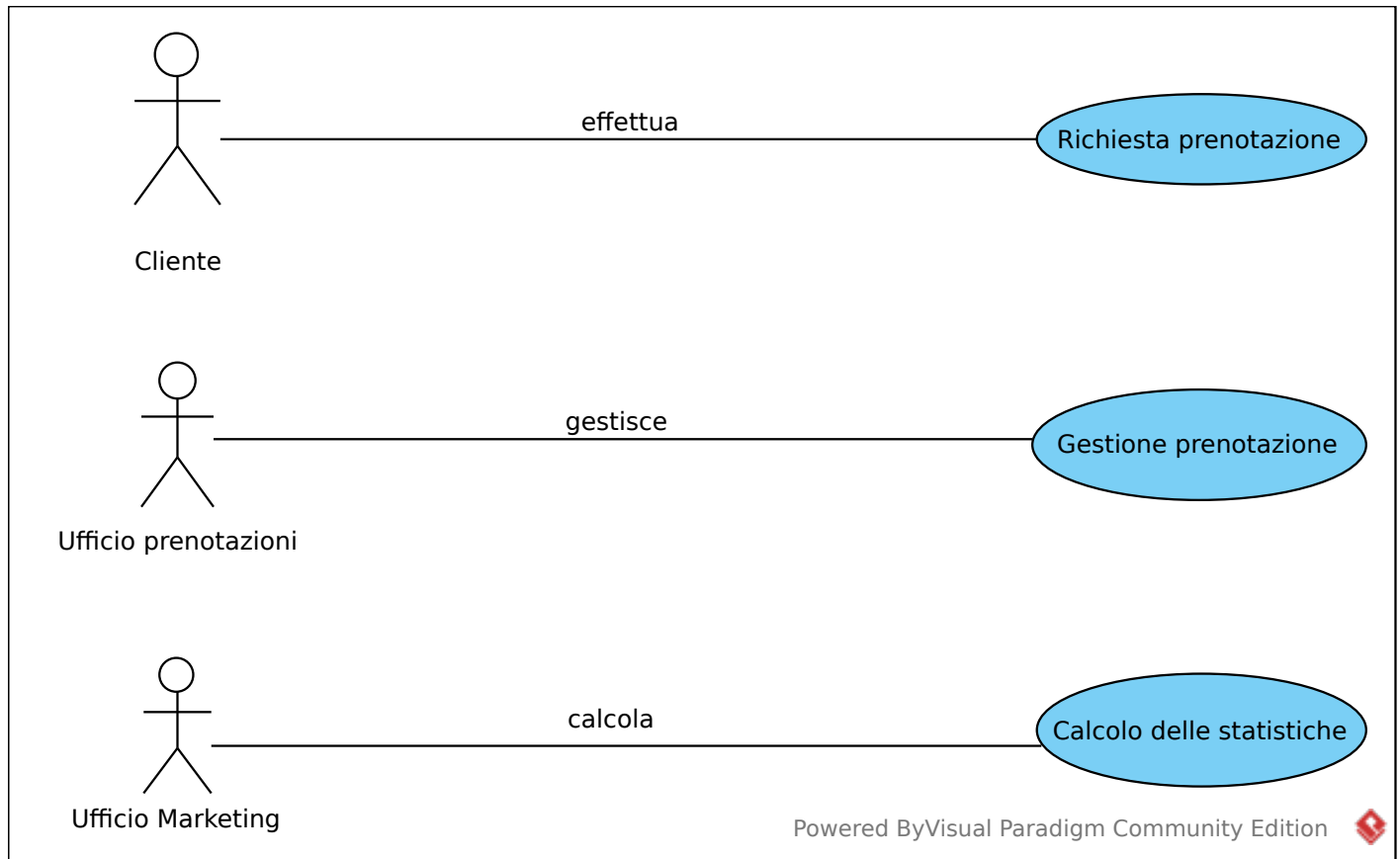


Figure 2: Diagramma degli Use Case

5 Specifiche dei tipi di dati

1. ValutazioneNave Intero in $[3,5]$
2. InteroGEZ Intero in $[0, +\infty]$
3. InteroGZ Intero in $(0, +\infty)$
4. Giorno {Lun, Mar, Mer, Gio, Ven, Sab, Dom}
5. DeltaOra {ora: intero in $[0,23]$, minuti: intero in $[0,59]$ }
6. StringaS varchar di lunghezza 100
7. StringaM varchar di lunghezza 500
8. StringaL varchar di lunghezza 1000
9. TipoCrocieraLunaDiMiele {tradizionale, alternativa}

Operazioni del tipo [DeltaOra](#):

1. $< (d1: \text{DeltaOra}, d2: \text{DeltaOra})$: booleano
 - 1.1. Pre: nessuna
 - 1.2. Post: no side effect
 - 1.3. Return:
 - 1.3.1. se $d1.ora < d2.ora$, restituisce vero
 - 1.3.2. se $d1.ora = d2.ora$ e $d1.minuti < d2.minuti$, restituisce vero
 - 1.3.3. altrimenti, restituisce falso

6 Specifiche delle classi e delle associazioni

Classe Crociera, [v.req.1](#)

Tabella delle specifiche degli attributi:

Attributo	Tipo	Cardinalità	Descrizione
codice	Stringa	1	Codice della crociera
data_inizio	Data	1	Data di inizio della crociera

Table 1: Attributi della classe Crociera

Tabella delle specifiche delle associazioni:

Associazione	Cardinalità	Descrizione
nave_croc	1	La nave utilizzata per la crociera
itin_croc	1	L'itinerario della crociera
pren_croc	0..*	Le prenotazioni della crociera

Table 2: Associazioni della classe Crociera

Specifiche delle operazioni della classe Crociera:

`data_fine_crociera()`: Data

1. Pre: nessuna
2. Post: no side effect
3. Siano arrivo, i, d, a tali che $\text{itin_croc}(i, \text{this}) \wedge \text{arrivo}(d, i)$
4. $\text{durata} = \text{giorno}(a, d, i)$
5. Return:
 - 5.1. restituisce la data di fine della crociera, calcolata come $\text{data_inizio}(\text{this}) + \text{durata}$

Specifiche delle chiavi primarie della classe Crociera:

1. codice

[Vincoli della classe Crociera](#)

Classe `Croc_luna_di_miele`, v.req.1.7.1.

Tabella delle specifiche degli attributi:

Attributo	Tipo	Cardinalità	Descrizione
tipo	TipoCrocieraLunaDiMiele	1	Tipo della crociera di luna di miele

Table 3: Attributi della classe `Classe Croc_luna_di_miele`

Specifica delle operazioni della classe `Croc_luna_di_miele`:

`tipCrocLdM()`: [TipoCrocieraLunaDiMiele](#)

1. Pre: nessuna
2. Post: no side effect
3. Return:
 - 3.1. se il numero di destinazioni romantiche \geq il numero di destinazioni divertenti, restituisce tradizionale
 - 3.2. altrimenti, restituisce alternativa

Classe `Croc_per_famiglie`, [v.req.1.7.2](#).

Tabella delle specifiche degli attributi:

Attributo	Tipo	Cardinalità	Descrizione
<code>adatta_ai_bambini</code>	booleano	1	Indica se la crociera è adatta ai bambini

Table 4: Attributi della classe `Classe Croc_per_famiglie`

Classe Nave, [v.req.2](#)

Tabella delle specifiche degli attributi:

Attributo	Tipo	Cardinalità	Descrizione
nome	StringaS	1	Nome della nave
grado_di_comfort	ValutazioneNave	1	Grado di comfort della nave
numero_massimo_di_passeggeri	InteroGZ	1	Numero massimo di passeggeri

Table 5: Attributi della classe Nave

Tabella delle specifiche delle associazioni:

Associazione	Cardinalità	Descrizione
nave_croc	0..*	Le crociere che fanno uso della nave

Table 6: Associazioni della classe Nave

Specifica delle chiavi primarie della classe Nave:

1. nome

[Vincoli della classe Nave](#)

Classe Destinazione, [v.req.3](#)

Tabella delle specifiche degli attributi:

Attributo	Tipo	Cardinalità	Descrizione
nome	StringaS	1	Nome della destinazione

Table 7: Attributi della classe Destinazione

Tabella delle specifiche delle associazioni:

Associazione	Cardinalità	Descrizione
dest_cit	1	La città in cui si trova la destinazione
dest_pdv	0..*	I posti da vedere della destinazione
dest_porto	1	Il porto dal quale può essere raggiunta la destinazione

Table 8: Associazioni della classe Destinazione

Specifica delle chiavi primarie della classe Destinazione:

1. (nome, dest_cit)

[Vincoli della classe Destinazione](#)

Classe Itinerario, v.req.4

Tabella delle specifiche degli attributi:

Attributo	Tipo	Cardinalità	Descrizione
nome	StringaS	1	Nome dell'itinerario

Table 9: Attributi della classe Itinerario

Tabella delle specifiche delle associazioni:

Associazione	Cardinalità	Descrizione
tappa_itin	1..*	Le tappe dell'itinerario
itin_croc	0..*	Le crociere che prevedono l'itinerario
partenza	1	La destinazione di partenza dell'itinerario
arrivo	1	La destinazione di arrivo dell'itinerario

Table 10: Associazioni della classe Itinerario

Specifica delle chiavi primarie della classe Itinerario:

1. nome

[Vincoli della classe Itinerario](#)

Association Class Partenza, [v.req.4.3](#).

Attributo	Tipo	Cardinalità	Descrizione
ora	DeltaOra	1	Ora di partenza

Table 11: Attributi dell'association class Partenza'

Association Class Arrivo, [v.req.4.4](#).

Attributo	Tipo	Cardinalità	Descrizione
ora	DeltaOra	1	Ora di arrivo
giorno	InteroGEZ	1	Giorno di arrivo

Table 12: Attributi dell'association class Arrivo'

Classe PostoDaVedere, v.req.5

Tabella delle specifiche degli attributi:

Attributo	Tipo	Cardinalità	Descrizione
nome	StringaS	1	Nome del posto da vedere
descrizione	StringaL	1	Descrizione del posto da vedere

Table 13: Attributi della classe Posto Da Vedere

Tabella delle specifiche delle associazioni:

Associazione	Cardinalità	Descrizione
pdv_cit	1	La città in cui si trova il posto da vedere
dest_pdv	0..*	Le destinazioni in cui si trova il posto da vedere
orari_pdv	0..*	Gli orari del posto da vedere

Table 14: Associazioni della classe Posto Da Vedere

Specifica delle chiavi primarie della classe Posto Da Vedere:

1. nome

[Vincoli della classe Posto Da Vedere](#)

Classe Utente, [v.req.6](#)

Tabella delle specifiche degli attributi:

Attributo	Tipo	Cardinalità	Descrizione
nome	StringaS	1	Nome dell'utente
cognome	StringaS	1	Cognome dell'utente
indirizzo	StringaM	1	Indirizzo dell'utente
data_di_nascita	Data	1	Data di nascita dell'utente

Table 15: Attributi della classe Utente

Tabella delle specifiche delle associazioni:

Associazione	Cardinalità	Descrizione
pren_utente	0..*	Le prenotazioni effettuate dall'utente
cit_utente	1	La città di residenza dell'utente

Table 16: Associazioni della classe Utente

Specifica delle chiavi primarie della classe Utente:

1. (nome, cognome, data_nascita)

[Vincoli della classe Utente](#)

Classe Prenotazione, [v.req.7](#)

Tabella delle specifiche degli attributi:

Attributo	Tipo	Cardinalità	Descrizione
istante_prenotazione	Dataora	1	Istante di prenotazione
num_posti_prenotati	InteroGZ	1	Numero di posti prenotati

Table 17: Attributi della classe Prenotazione

Tabella delle specifiche delle associazioni:

Associazione	Cardinalità	Descrizione
pren_croc	1	La crociera prenotata
pren_utente	1	L'utente che ha effettuato la prenotazione

Table 18: Associazioni della classe Prenotazione

Specifica delle chiavi primarie della classe Prenotazione:

1. (pren_croc, istante_prenotazione)

[Vincoli della classe Prenotazione](#)

Classe Pren_accettata

Classe Pren_rifiutata

Classe Città, v.req.8

Tabella delle specifiche degli attributi:

Attributo	Tipo	Cardinalità	Descrizione
nome	StringaS	1	Nome della città

Table 19: Attributi della classe Città

Tabella delle specifiche delle associazioni:

Associazione	Cardinalità	Descrizione
cit_naz	1	La nazione in cui si trova la città
dest_cit	0..*	Le destinazioni che si trovano nella città
pdv_cit	0..*	I posti da vedere che si trovano nella città
cit_porto	0..*	I porti che si trovano nella città
cit_utente	0..*	Gli utenti che risiedono nella città

Table 20: Associazioni della classe Città

Specifica delle chiavi primarie della classe Città:

1. (nome, cit_naz)

[Vincoli della classe Città](#)

Classe Nazione, v.req.9

Tabella delle specifiche degli attributi:

Attributo	Tipo	Cardinalità	Descrizione
nome	StringaS	1	Nome della nazione

Table 21: Attributi della classe Nazione

Tabella delle specifiche delle associazioni:

Associazione	Cardinalità	Descrizione
naz_cont	1	Il continente in cui si trova la nazione
cit_naz	0..*	Le città che si trovano nella nazione

Table 22: Associazioni della classe Nazione

Specifica delle chiavi primarie della classe Nazione:

1. nome

[Vincoli della classe Nazione](#)

Classe Continente, [v.req.10](#)

Tabella delle specifiche degli attributi:

Attributo	Tipo	Cardinalità	Descrizione
nome	StringaS	1	Nome del continente

Table 23: Attributi della classe Continente

Tabella delle specifiche delle associazioni:

Associazione	Cardinalità	Descrizione
naz_cont	0..*	Le nazioni che si trovano nel continente

Table 24: Associazioni della classe Continente

Specifica delle chiavi primarie della classe Continente:

1. nome

[Vincoli della classe Continente](#)

Classe Porto, [v.req.11](#)

Tabella delle specifiche degli attributi:

Attributo	Tipo	Cardinalità	Descrizione
nome	StringaS	1	Nome del porto

Table 25: Attributi della classe Porto

Tabella delle specifiche delle associazioni:

Associazione	Cardinalità	Descrizione
cit_porto	1	La città in cui si trova il porto
dest_porto	0..1	La destinazione che può essere raggiunta dal porto

Table 26: Associazioni della classe Porto

Specifica delle chiavi primarie della classe Porto:

1. (nome, cit_porto)

[Vincoli della classe Porto](#)

Classe OrarioPostoDaVedere, [v.req.12](#)

Tabella delle specifiche degli attributi:

Attributo	Tipo	Cardinalità	Descrizione
giorno	Giorno	1	Giorno in cui è possibile visitare il posto
ora_inizio	DeltaOra	1	Ora di inizio della visita
ora_fine	DeltaOra	1	Ora di fine della visita

Table 27: Attributi della classe Orario Posto Da Vedere

Tabella delle specifiche delle associazioni:

Associazione	Cardinalità	Descrizione
orari_pdv	1	Il posto da vedere per il quale è definito l'orario

Table 28: Associazioni della classe Orario Posto Da Vedere

Specifica delle chiavi primarie della classe Orario Posto Da Vedere:

1. (giorno, ora_inizio, orari_pdv)

[Vincoli della classe Orario Posto Da Vedere](#)

Classe Tappa, [v.req.13](#)

Tabella delle specifiche degli attributi:

Attributo	Tipo	Cardinalità	Descrizione
giorno_arrivo	InteroGEZ	1	Giorno di arrivo della tappa
ora_arrivo	DeltaOra	1	Ora di arrivo della tappa
giorno_partenza	InteroGEZ	1	Giorno di partenza della tappa
ora_partenza	DeltaOra	1	Ora di partenza della tappa

Table 29: Attributi della classe Tappa

Specifica delle operazioni della classe Tappa:

ordineTappa(): [InteroGZ](#)

1. Pre: nessuna
2. Post: no side effect
3. Return:
 - 3.1. sia I l'itinerario tale che $\text{tappa_itin}(\text{this}, I) = \text{True}$
 - 3.2. sia T l'insieme delle tappe di I
 - 3.3. sia $T' = \{t' \in T \mid (\text{giorno_arrivo}(t') \leq \text{giorno_arrivo}(t) \wedge \text{ora_partenza}(t') \leq \text{ora_arrivo}(t))\}$
 - 3.4. restituisce la $|T'|+1$

Tabella delle specifiche delle associazioni:

Associazione	Cardinalità	Descrizione
tappa_itin	1	L'itinerario della tappa
tappa_dest	1	La destinazione della tappa

Table 30: Associazioni della classe Tappa

Specifica delle chiavi primarie della classe Tappa:

1. (tappa_itin, giorno_arrivo, ora_arrivo)

[Vincoli della classe Tappa](#)

7 Specifiche dei vincoli esterni

[V.Crociera.]

[V.Crociera.data_inizio_leq_data_fine.]:

1. data di inizio \leq data di fine

[V.Crociera.prenotazioni_leq_numero_massimo_passeggeri.]:

1. La somma dei posti prenotati per la crociera \leq numero massimo di passeggeri

[V.Nave.]

[V.Destinazione.]

[V.Itinerario.]

[V.PostoDaVedere.]

[V.Utente.]

[V.Prenotazione.]

[V.Prenotazione.istante_di_prenotazione_less_data_inizio_crociera.]:

1. istante di prenotazione $<$ data di inizio crociera

[V.Città.]

[V.Nazione.]

[V.Continente.]

[V.Porto.]

[V.OrarioPostoDaVedere.]

[V.OrarioPostoDaVedere.ora_inizio_less_ora_fine.]:

1. ora di inizio $<$ ora di fine

[V.**Tappa.**]

[V.Tappa.giorno_arrivo_leq_giorno_partenza.]

1. giorno di arrivo \leq giorno di partenza

8 Specifiche delle operazioni

In questa sezione verranno descritte le operazioni che il sistema deve essere in grado di eseguire.

[accettaPrenotazione(Crociera *c*, Utente *u*, InteroGZ *num_posti*)]

1. Permette all'Ufficio Prenotazioni di effettuare una prenotazione. Restituisce un oggetto di tipo Prenotazione se $\text{getPostiDisponibili}(c) \geq \text{num_posti}$, altrimenti rifiuta.
2. precondizioni: $\text{getPostiDisponibili}(c) \geq \text{num_posti} \wedge \text{istante_corrente} \leq \text{data.inizio}(c)$
3. Se la precondizione non è soddisfatta, restituisce un oggetto di tipo *pren_rifiutata*.
4. postcondizioni:
5. Viene creato un nuovo oggetto *p* della classe *pren_accettata*:
 - 5.1. $\text{istante_prenotazione}(p) = \text{istante_corrente}$
 - 5.2. $\text{num_posti_prenotati}(p) = \text{num_posti}$
6. E vengono create le seguenti nuove associazioni:
 - 6.1. $\text{pren_croc}(p, c)$
 - 6.2. $\text{pren_utente}(p, u)$
7. return *p*

[getEtàMediaClientiCheHannoPrenotatoInUnPeriodoUnaCrocieraConAlmenoUnaDestinazioneEsotica(Data inizio, Data fine)]

1. Restituisce l'età media dei clienti che hanno prenotato in un periodo una crociera con almeno una destinazione esotica.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. $P = \text{getPrenotazioniInUnDatoPeriodo}(\text{inizio}, \text{fine})$
5. $C = \text{getCrociereDaPrenotazioni}(P)$
6. $D = \text{getDestinazioniDaCrociera}(C)$
7. $D_{\text{esotiche}} = \{d \in D : \text{isDestinazioneEsotica}(d)\}$
8. $U = \{u \in Utente : \exists p, c, I, t, d \wedge p \in P \wedge c \in C \wedge d \in D_{\text{esotiche}} \wedge I \in Itinerario \wedge t \in Tappa \wedge \text{pren_utente}(p, u) \wedge \text{pren_croc}(p, c) \wedge \text{itin_croc}(I, c) \wedge \text{tappa_itin}(t, I) \wedge \text{tappa_dest}(t, d)\}$
9. return $\text{getEtàMediaUtenti}(U)$

[getPercentualeDestinazioniGettonateInUnDatoPeriodo(Data inizio, Data fine)]

1. Restituisce la percentuale di destinazioni gettonate in un dato periodo.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. $D =$ l'insieme di tutte le destinazioni
5. $R = \text{getDestinazioniGettonateInUnDatoPeriodo}(\text{inizio}, \text{fine})$
6. return $|R|*100/|D|$

[getPostiDisponibili(Crociera c)]

1. Restituisce il numero di posti disponibili data una determinata Crociera.
2. precondizioni: $\exists n \mid \text{nave_croc}(n, c)$
3. postcondizioni: no side effect
4. $P = \text{getPrenotazioniAccettate}(\text{Crociera } c)$
5. $\text{num_posti} = \text{num_posti_disponibili}(n) - \sum_{p \in P} \text{num_posti_prenotati}(p)$
6. return num_posti

[getPrenotazioniAccettate(Crociera c)]

1. Restituisce l'insieme delle prenotazioni accettate di una determinata Crociera.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. return $\{p \in \text{Prenotazione} : \text{pren_croc}(p, c) \wedge \text{Pren_accettata}(p)\}$

[effettuaRichiestaPrenotazione(Crociera c, Utente u, InteroGZ num_posti)]

1. Permette all'Utente di effettuare una richiesta di prenotazione. Restituisce un oggetto di tipo Prenotazione.
2. precondizioni: nessuna
3. postcondizioni: viene modificato l'aspetto estensionale dei dati
4. Viene creato una nuova istanza p della classe Prenotazione
5. $\text{istante_prenotazione}(p) = \text{istante_corrente}$
6. $\text{num_posti_prenotati}(p) = \text{num_posti}$
7. Viene creata l'associazione $\text{pren_croc}(p, c)$
8. Viene creata l'associazione $\text{pren_utente}(p, u)$
9. return p

[isDestinazioneEsotica(Destinazione d)]

1. Restituisce vero se la destinazione è esotica (se il suo continente è diverso da Europa), altrimenti falso.
2. precondizioni: $\exists \text{ cit, naz, cont} \mid \text{dest_cit}(\text{d}, \text{cit}) \wedge \text{cit_naz}(\text{cit}, \text{naz}) \wedge \text{naz_cont}(\text{naz}, \text{cont})$
3. postcondizioni: no side effect
4. return: $\text{nome}(\text{cont}) \neq \text{Europa}$

[isPrenotazioneInPeriodo(Prenotazione p, Data inizio, Data fine)]

1. Restituisce vero se la prenotazione è in un periodo compreso tra inizio e fine, altrimenti falso.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. return $\text{istante_prenotazione}(p) \geq \text{inizio} \wedge \text{istante_prenotazione}(p) \leq \text{fine}$

[getEtàMediaUtenti(Utente[] U)]

1. Restituisce l'età media degli utenti.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. return $\frac{\sum_{i=0}^{|U|} \text{getEta}(U[i])}{|U|}$
5. (operazione [getEta](#))

[getEta(Utente u)]

1. Restituisce l'età dell'utente.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. return $\text{istante_corrente} - \text{data_di_nascita}(u)$

[getPrenotazioniInUnDatoPeriodo(Data inizio, Data fine)]

1. Restituisce l'insieme delle prenotazioni accettate in un dato periodo.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. return $\{p \in \text{Pren_accettata} : \text{isPrenotazioneInPeriodo}(p, \text{inizio}, \text{fine})\}$

[getCrociereDaPrenotazioni(Prenotazione[] P)]

1. Restituisce l'insieme delle crociere a partire da un insieme di prenotazioni.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. return $\{c \in Crociera : p \in P \wedge pren_croc(p, c)\}$

[getDestinazioniDaCrociere(Crociere[] C)]

1. Restituisce l'insieme delle destinazioni a partire da un insieme di crociere.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. return $\{d \in Destinazione : \exists c \in C : tappa_dest(t, d) \wedge t \in Tappa \wedge tappa_itin(t, I) \wedge I \in Itinerario \wedge itin_croc(I, c)\}$

[getDestinazioniGettonateInUnDatoPeriodo(Data inizio, Data fine)]

1. Restituisce il numero di destinazioni da considerarsi gettonate in un dato periodo. Una destinazione è gettonata se è raggiunta da almeno 10 crociere luna di miele in quel periodo oppure da almeno 15 crociere per famiglia.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. D = l'insieme di tutte le destinazioni
5. $R = \{d \in D : isDestinazioneGettonataInUnDatoPeriodo(d, inizio, fine)\}$
6. return R

[isDestinazioneGettonataInUnDatoPeriodo(Destinazione d, Data inizio, Data fine)]

1. Restituisce vero se la destinazione è gettonata in un dato periodo, altrimenti falso.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. $CpF = getCrocierePerFamiglieCheRaggiungeUnaDataDestinazioneInUnDatoPeriodo(d, inizio, fine)$
5. $CLdM = getCrociereLunaDiMieleCheRaggiungeUnaDataDestinazioneInUnDatoPeriodo(d, inizio, fine)$
6. return True se $|CpF| \geq 15 \vee |CLdM| \geq 10$, altrimenti False

[getCrocieraPerFamiglieCheRaggiungeUnaDataDestinazioneInUnDatoPeriodo(Destinazione d, Data inizio, Data fine)]

1. Restituisce le crociere per famiglie che raggiungono una data destinazione in un dato periodo.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. $P = \text{getCrocieraPerFamiglieInUnDatoPeriodo}(\text{inizio}, \text{fine})$
5. $D = \text{getCrocieraPerFamiglieInDestinazione}(d)$
6. return $P \cap D$

[getCrocieraLunaDiMieleCheRaggiungeUnaDataDestinazioneInUnDatoPeriodo(Destinazione d, Data inizio, Data fine)]

1. Restituisce le crociere luna di miele che raggiungono una data destinazione in un dato periodo.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. $P = \text{getCrocieraLunaDiMieleInUnDatoPeriodo}(\text{inizio}, \text{fine})$
5. $D = \text{getCrocieraLunaDiMieleInDestinazione}(d)$
6. return $P \cap D$

[isCrocieraInPeriodo(Crociera c, Data inizio, Data fine)]

1. Restituisce vero se la crociera è in un periodo compreso tra inizio e fine, altrimenti falso.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. return true se $\text{data_inizio}(c) \geq \text{inizio} \wedge \text{data_fine}(c) \leq \text{fine}$, altrimenti false

[getCrocieraPerFamiglieInUnDatoPeriodo(Data inizio, Data fine)]

1. Restituisce le crociere per famiglie in un dato periodo.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. return $\{c \in \text{Crociera} : \text{isCrocieraInPeriodo}(c, \text{inizio}, \text{fine}) \wedge c \in \text{CrocieraPerFamiglie}\}$

[getCrociereLunaDiMieleInUnDatoPeriodo(Data inizio, Data fine)]

1. Restituisce le crociere luna di miele in un dato periodo.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. return $\{c \in Crociere : \text{isCrocieraInPeriodo}(c, inizio, fine) \wedge c \in CrociereLunaDiMiele\}$

[getCrociereLunaDiMieleInDestinazione(Destinazione d)]

1. Restituisce le crociere luna di miele che raggiungono una data destinazione.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. return $\{c \in CrociereLunaDiMiele : \text{isCrocieraInDestinazione}(c, d)\}$

[getCrocierePerFamiglieInDestinazione(Destinazione d)]

1. Restituisce le crociere per famiglie che raggiungono una data destinazione.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. return $\{c \in CrocierePerFamiglie : \text{isCrocieraInDestinazione}(c, d)\}$

[isCrocieraInDestinazione(Crociera c, Destinazione d)]

1. Restituisce vero se la crociera raggiunge la destinazione, altrimenti falso.
2. precondizioni: nessuna
3. postcondizioni: no side effect
4. return
 - 4.1. true se esistono i, t tale che $tappa_itin(t, i) \wedge tappa_dest(t, d) \wedge itin_croc(i, c)$
 - 4.2. \vee se esiste i tale che $arrivo(i, d) \wedge itin_croc(i, c)$
 - 4.3. \vee se esiste i tale che $partenza(i, d) \wedge itin_croc(i, c)$
 - 4.4. altrimenti false