Biglietteria compagnia di navigazione

IPP - Incoronato, Perrella, Pitacoro

3 giugno 2022

Indice

1 Biglietteria compagnia di navigazione **2**

2 Stima dei costi **3**

3 Analisi e specifica dei requisiti **6**
3.1 Revisione dei requisiti 6
3.2 Glossario dei termini 7
3.3 Classificazione dei requisiti 8
3.3.1 Requisiti funzionali 8
3.3.2 Requisiti sui dati 9
3.3.3 Vincoli/Altri requisiti 9

4 Casi d'uso **10**
4.1 Specifica degli scenari 10

5 Diagramma dei casi d'uso **13**

6 Diagramma delle classi di analisi **14**

7 BCE **15**

8 Diagramma delle classi di progettazione **16**

9 BCED **17**

Capitolo 1

Biglietteria compagnia di navigazione

Si vuole realizzare un'applicazione per compagnie di navigazione per l'emissione dei biglietti per viaggi marittimi.
Una compagnia di navigazione (caratterizzata dal proprio nome) fornisce servizi di trasporto passeggeri (operati con aliscafi e traghetti) e di trasporto passeggeri e autoveicoli (operati con traghetti). Ogni nave ha un nome e una capienza in termini di numero di posti disponibili per i passeggeri e, per i traghetti, per gli autoveicoli.
La compagnia di navigazione opera con le proprie navi un insieme di corse giornaliere tra N porti. Esistono due tipi di biglietti, per passeggero e per autoveicolo. Ogni corsa è identificata da un codice e ha orario e porto di partenza e di arrivo, e un prezzo dei biglietti prefissato. Una corsa è effettuata sempre dalla stessa nave.
La biglietteria di una compagnia può emettere biglietti viaggio, validi per una specifica corsa in un determinato giorno. Prima di emettere un biglietto, il sistema deve valutare l'effettiva disponibilità del posto. All'atto dell'emissione dovrà essere aggiornata la disponibilità residua.
L'applicazione supporta l'emissione di biglietti da parte degli impiegati di una compagnia. Ogni impiegato deve accedere al sistema solo previa autenticazione. Per ogni biglietto il sistema deve stampare la data e l'ora di emissione. Per ciascun biglietto il sistema deve registrare internamente, al momento dell'emissione, data e ora di emissione e codice dell'impiegato che lo ha emesso. Per i biglietti per i veicoli occorre registrare e stampare la targa del veicolo.
È altresì previsto l'uso dell'applicazione da parte di utenti clienti di una compagnia, che possono prenotare via web un biglietto passeggero e/o per autoveicolo. L'emissione del biglietto prenotato avverrà tuttavia a cura di un impiegato della compagnia, cui il cliente dovrà esibire una stampa della prenotazione che riporti gli estremi della corsa e, eventualmente, la targa del veicolo.

Capitolo 2

Stima dei costi

In questa sezione viene riportata la stima dei costi effettuata secondo IFPUG.

ILF (Internal Logical File): 1

EI (External Input):

- Autenticazione $\Longrightarrow$ BASSO x3
- Registrazione_biglietto $\Longrightarrow$ MEDIO x4
- Registrazione_cliente $\Longrightarrow$ BASSO x3
- Prenotazione_cliente $\Longrightarrow$ MEDIO x4
- Gestione_posti $\Longrightarrow$ BASSO x3

EQ (External inQuiry):

- Emissione_biglietto_biglietteria $\Longrightarrow$ BASSO x3
- Emissione_biglietto_impiegato $\Longrightarrow$ BASSO x3
- Ricerca_rotta $\Longrightarrow$ MEDIO x4

CORSA ENTITY TYPE	RET1
Codice	DET1
Orario_partenza	DET2
Porto_partenza	DET3
Orario_Arrivo	DET4
Porto_Arrivo	DET5
Prezzo	DET6

BIGLIETTO ENTITY TYPE	RET2
Nome	DET1
Tipo	DET2
Giorno	DET3
Impiegato	DET4
Targa	DET5
Ora	DET6

IMPIEGATO ENTITY TYPE	RET4
Codice	DET1
Nome	DET2
Password	DET3

CLIENTE ENTITY TYPE	RET3
Codice_Fiscale	DET1

NAVE ENTITY TYPE	RET5
Nome	DET1
Tipo	DET2
Capienza_Passeggeri	DET3
Capienza_Autoveicoli	DET4

	DataElementTypes		
RET	$1-19$	$20-50$	≥ 51
1	Basso	Basso	Medio
$2-5$	Basso	Medio	Alto
≥ 5	Medio	Alto	Alto

Tabella 2.1: Valutazione complessità ILF

In definitiva, con riferimento alla tabella2.1

- ILF: 1
- RET: 5 $\Longrightarrow$ ILF ha complessità: MEDIO
- DET: 21

Componente	Basso	Medio	Alto
EI	$\times 3$	$\times 4$	$\times 6$
EO	$\times 4$	$\times 5$	$\times 7$
EQ	$\times 3$	$\times 4$	$\times 6$
ILF	$\times 7$	$\times 10$	$\times 15$
EIF	$\times 5$	$\times 7$	$\times 10$

Tabella 2.2: Tabella del conteggio UFP

$$UFP = 10 + (3 + 4 + 3 + 4 + 3 + 3 + 3 + 4) = 10 + 27 = 37$$

Adesso valutiamo i fattori correttivi per completare il calcolo dei Function Point.

Fattori correttivi:

1.	Comunicazione Dati	4
2.	Distribuzione Elaborazione	3
3.	Prestazioni	0
4.	Utilizzo intensivo configurazione	2
5.	Frequenza delle transazioni	1
6.	Inserimento dati interattivo	5
7.	Efficienza per l'utente finale	3
8.	Aggiornamento interattivo	4
9.	Complessità Elaborativa	3
10.	Riusabilità	4
11.	Facilità Installazione	0
12.	Facilità Gestione Operativa	1
13.	Molteplicità di Siti	1
14.	Facilità di Modifica	3
	TOTALE	34

$$FP = UFP \times \left(0.65 + 0.01 \times \sum_{i=1}^{14} F_i\right) = 37 \times (0.65 + 0.01 \times 34) = 37 \times 0.99 = 36.63$$

Considerazioni effettuate:

- I biglietti generano un EI ed un EO, perchè bisogna "registrare internamente" (EI) e bisogna "stampare data e ora di emissione" (EO)

Capitolo 3

Analisi e specifica dei requisiti

3.1 Revisione dei requisiti

1. Una compagnia di navigazione che utilizza il sistema è identificata dal proprio nome.
2. Ogni nave è caratterizzata dal proprio nome.
3. Ogni aliscafo ha una capienza in termini di numero di posti disponibili per i passeggeri.
4. Ogni traghetto ha una capienza in termini di numero di posti disponibili per i passeggeri e per autoveicoli.
5. Il sistema memorizza le corse effettuate da ogni nave della compagnia.
6. I biglietti possono essere di due tipi, per passeggero e per autoveicolo.
7. Ogni corsa è identificata da un codice.
8. Ogni corsa è caratterizzata da orario e porto di partenza e di arrivo.
9. Il prezzo dei biglietti per una determinata corsa è prefissato.
10. Una corsa è effettuata sempre dalla stessa nave.
11. Il sistema emette, tramite una biglietteria, biglietti che sono validi per una specifica corsa in un determinato giorno.
12. Il sistema deve valutare la disponibilità del posto.
13. Il sistema deve aggiornare la disponibilità residua, all'atto dell'emissione.
14. Il sistema emette, tramite un impiegato, biglietti che sono validi per una specifica corsa in un determinato giorno.
15. L'impiegato deve autenticarsi per accedere al sistema.
16. Il sistema deve stampare data e ora di emissione, per ogni biglietto, al momento dell'emissione dello stesso.

17. Il sistema deve memorizzare data e ora di emissione e codice dell'impiegato per ogni biglietto, al momento dell'emissione dello stesso.

18. Il sistema deve stampare data e ora di emissione e targa, per ogni biglietto autoveicolo, al momento dell'emissione dello stesso.

19. Il sistema deve memorizzare data e ora di emissione, codice dell'impiegato e targa, per ogni biglietto autoveicolo, al momento dell'emissione dello stesso.

20. Il sistema deve fornire al cliente la funzionalità per prenotare via web un biglietto passeggero.

21. Il sistema deve fornire al cliente la funzionalità per prenotare via web un biglietto autoveicolo.

22. Il sistema deve offrire all'utente una funzionalità per ricercare le corse per data e porto di partenza e per data e porto di arrivo.

3.2 Glossario dei termini

[OPZIONALE] Riportare un glossario dei termini in una tabella termine-descrizione (significato)-eventuali sinonimi, come nel seguente esempio.

Termine	Descrizione	Sinonimi
Cliente	Individuo che usufruisce del servizio di trasporto marittimo.	Utente
Impiegato	Individuo che lavora presso una compagnia, che si occupa dell'emissione di biglietti e alla gestione dei biglietti prenotati.	Dipendente
Biglietteria	Sistema esterno automatizzato che emette biglietti.	
Corsa	Percorso di una nave da un porto di partenza a un porto destinazione ad uno specificato orario.	
Compagnia di navigazione	Società che esegue il trasporto marittimo di persone e/o autoveicoli utilizzando navi di proprietà.	
Biglietto	Titolo di viaggio valido per una specifica corsa in un determinato giorno.	

3.3 Classificazione dei requisiti

Classificare i requisiti (funzionali, sui dati, vincoli/altri requisiti), riportando le frasi dei requisiti (revisionati) che li esprimono (ciascuna frase deve esprimere un singolo requisito), numerandoli secondo una codifica (es.: RF01, RF02, RFn per i requisiti funzionali; RD01, RD02, RDm per i requisiti sui dati). Verificare che i requisiti siano chiari, completi e verificabili.

3.3.1 Requisiti funzionali

ID	Requisito	Origine(n. frase dei requisiti revisionati)
RF01	Il sistema memorizza le corse effettuate da ogni nave della compagnia.	5
RF02	Il sistema emette, tramite una biglietteria, biglietti che sono validi per una specifica corsa in un determinato giorno.	11
RF03	Il sistema deve valutare la disponibilità del posto.	12
RF04	Il sistema deve aggiornare la disponibilità residua, all'atto dell'emissione.	13
RF05	Il sistema emette, tramite un impiegato, biglietti che sono validi per una specifica corsa in un determinato giorno.	14
RF06	L'impiegato deve autenticarsi per accedere al sistema.	15
RF07	Il sistema deve stampare data e ora di emissione, per ogni biglietto, al momento dell'emissione dello stesso.	16
RF08	Il sistema deve memorizzare data e ora di emissione e codice dell'impiegato per ogni biglietto, al momento dell'emissione dello stesso.	17
RF09	Il sistema deve stampare data e ora di emissione e targa, per ogni biglietto autoveicolo, al momento dell'emissione dello stesso.	18
RF10	Il sistema deve memorizzare data e ora di emissione, codice dell'impiegato e targa per ogni biglietto autoveicolo, al momento dell'emissione dello stesso.	19
RF11	Il sistema deve fornire al cliente la funzionalità per prenotare via web un biglietto passeggero.	20
RF12	Il sistema deve fornire al cliente la funzionalità per prenotare via web un biglietto autoveicolo.	21
RF13	Il sistema deve offrire al cliente una funzionalità per ricercare le corse per data e porto di partenza e per data e porto di arrivo.	22

ID	Requisito	Origine(n. frase dei requisiti revisionati)
RD01	Una compagnia di navigazione che utilizza il sistema è identificata dal proprio nome.	1
RD02	Ogni nave è caratterizzata dal proprio nome.	2
RD03	Ogni aliscafo ha una capienza in termini di numero di posti disponibili per i passeggeri.	3
RD04	Ogni traghetto ha una capienza in termini di numero di posti disponibili per i passeggeri e per autoveicoli.	4
RD05	I biglietti possono essere di due tipi, per passeggero e per autoveicolo.	6
RD06	Ogni corsa è identificata da un codice.	7
RD07	Ogni corsa è caratterizzata da orario e porto di partenza e di arrivo.	8
RD08	Il prezzo dei biglietti per una determinata corsa è prefissato.	9

3.3.2 Requisiti sui dati

3.3.3 Vincoli/Altri requisiti

Descrivere eventuali vincoli o requisiti non funzionali. Per es., stimare il volume dei dati e delle transazioni (Quanti utenti si stimano? Quante operazioni si prevedono al giorno per ogni utente? Qual è la frequenza relativa dei vari tipi di operazioni (es: interrogazioni 100/h, modifiche 5/g, ...))

ID	Requisito	Origine(n. frase dei requisiti revisionati)
RNF01	Una corsa è effettuata sempre dalla stessa nave.	10

Capitolo 4

Casi d'uso

4.1 Specifica degli scenari

CASO D'USO: EmissioneBiglietto
ID: 1
Breve descrizione: il Sistema emette un biglietto valido per una specifica corsa in un determinato giorno, previo controllo della disponibilità del posto.
Attori primari: Impiegato
Attori secondari: None
Precondizioni: l'Impiegato deve essere autenticato dal sistema.
Sequenza degli eventi: 1. L'impiegato avvia il caso d'uso EmissioneBiglietto. 2. L'impiegato seleziona la corsa e il giorno. 3. Il Sistema mostra le corse. 4. L'impiegato sceglie il tipo di biglietto da emettere. punto di estensione: ContaPosti 5. Se il numero di posti disponibili è 0 5.1. Il Sistema segnala un errore e interrompe la procedura di emissione del biglietto. 6. Altrimenti 6.1. Il Sistema emette il biglietto, aggiornando il numero di posti disponibili. 6.2. Se il biglietto è per persona 6.2.1. Il Sistema registra data e ora di emissione del biglietto 6.3. Se il biglietto è per autoveicolo 6.3.1. Il Sistema registra data e ora di emissione e targa del biglietto
Postcondizioni: None
Sequenze degli eventi alternative: None

CASO D'USO DI ESTENSIONE: ContaPosti
ID: 2
Breve descrizione: il Sistema deve valutare l'effettiva disponibilità del posto.
Attori primari: None
Attori secondari: None
Precondizioni: 1. L'impiegato seleziona una corsa programmata per un determinato giorno. 2. L'impiegato deve essere autenticato dal sistema.
Sequenza degli eventi: 1. Il Sistema conta i posti disponibili e li mostra all'impiegato.
Postcondizioni: None
Sequenze degli eventi alternative: None

CASO D'USO: PrenotaBiglietto
ID: 3
Breve descrizione: un Cliente, tramite l'uso dell'applicazione, effettua la prenotazione di un biglietto passeggero e/o per autoveicolo.
Attori primari: Cliente
Attori secondari: None
Precondizioni: None
Sequenza degli eventi: 1. Il Cliente avvia il caso d'uso PrenotaBiglietto. 2. Il Cliente seleziona la corsa e il giorno desiderati. 3. Il Sistema mostra le corse. 4. Il Cliente sceglie il tipo di biglietto da prenotare. punto di estensione: ContaPosti 5. Se il numero di posti disponibili è 0 5.1. Il Sistema segnala un errore e interrompe la procedura di prenotazione del biglietto. 6. Altrimenti 6.1. Il Sistema effettua la prenotazione, aggiornando il numero di posti disponibili.
Postcondizioni: La prenotazione è stata aggiunta al sistema
Sequenze degli eventi alternative: None

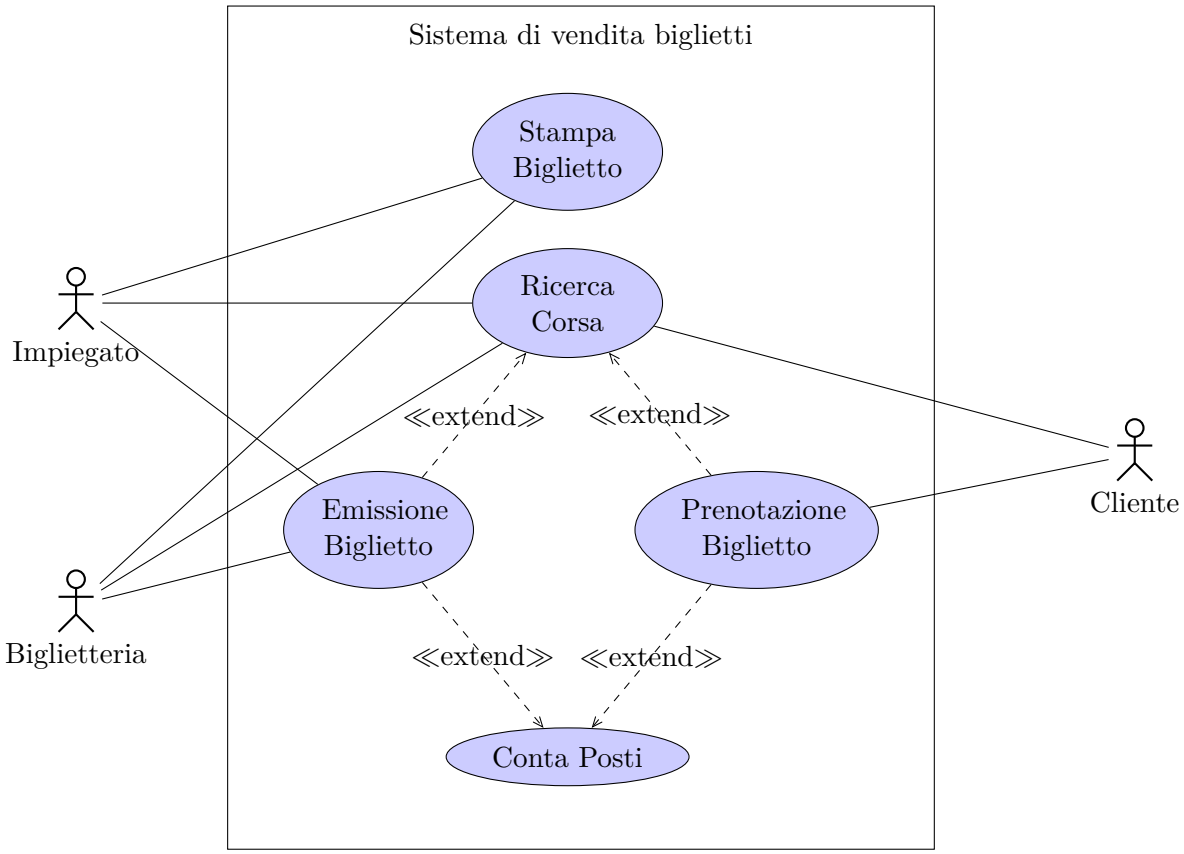
CASO D'USO: StampaBiglietto
ID: 4
Breve descrizione: il Sistema stampa le informazioni relative al biglietto al momento della sua emissione.
Attori primari: Impiegato
Attori secondari: None
Precondizioni: l'Impiegato deve essere autenticato dal sistema.
Sequenza degli eventi: 1. L'impiegato avvia il caso d'uso StampaBiglietto per un biglietto. 2. Se il biglietto è per un autoveicolo 2.1. Il Sistema raccoglie data e ora di emissione del biglietto e la targa del veicolo. 2.2. Il Sistema stampa il biglietto. 3. Altrimenti 3.1. Il Sistema raccoglie data e ora di emissione del biglietto. 3.2. Il Sistema stampa il biglietto.
Postcondizioni: None
Sequenze degli eventi alternative: al punto 1, se il Cliente ha prenotato una corsa allora l'Impiegato avvia il caso d'uso StampaBiglietto utilizzando gli estremi della corsa e, eventualmente la targa del veicolo, forniti dal Cliente.

CASO D'USO: RicercaCorsa
ID: 5
Breve descrizione: Il sistema deve offrire al cliente una funzionalità per ricercare le corse per data e porto di partenza e per data e porto di arrivo.
Attori primari: Impiegato, Biglietteria, Cliente
Attori secondari: None.
Precondizioni: None.
Sequenza degli eventi: 1. L'Utente avvia il caso d'uso "RicercaCorsa". 2. L'Utente seleziona i criteri di ricerca e inserisce il valore. 3. Il Sistema cerca le corse che soddisfano i criteri dell'utente. 4. Se il sistema trova delle corse che soddisfano i criteri 4.1. Per ogni corsa 4.1.1. Il Sistema mostra un riepilogo dei dettagli della corsa. 5. Altrimenti 5.1. Il Sistema informa il Cliente che i criteri immessi non corrispondono a nessuna corsa.
Postcondizioni: None
Sequenze degli eventi alternative:None.

Capitolo 5

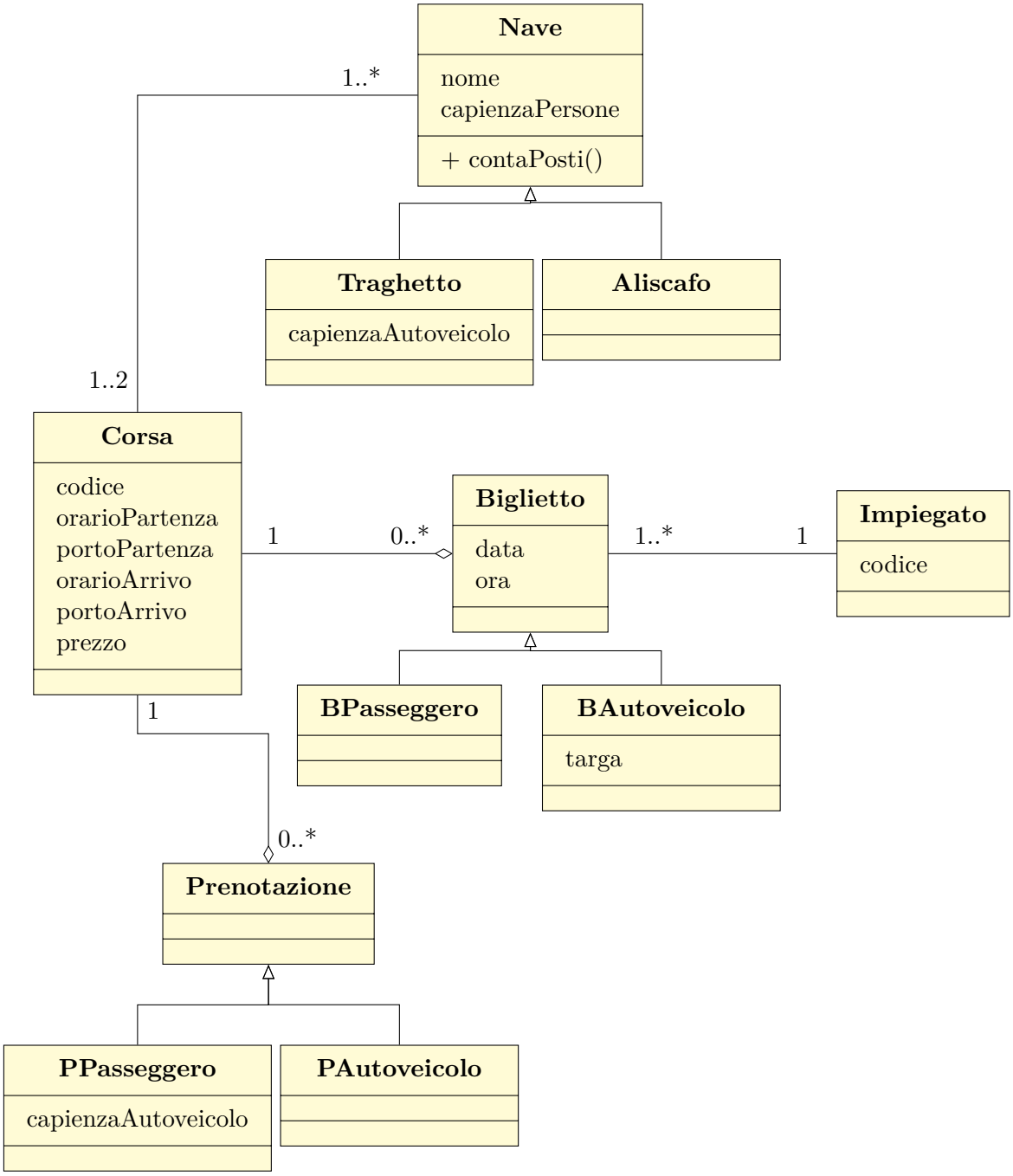
Diagramma dei casi d'uso

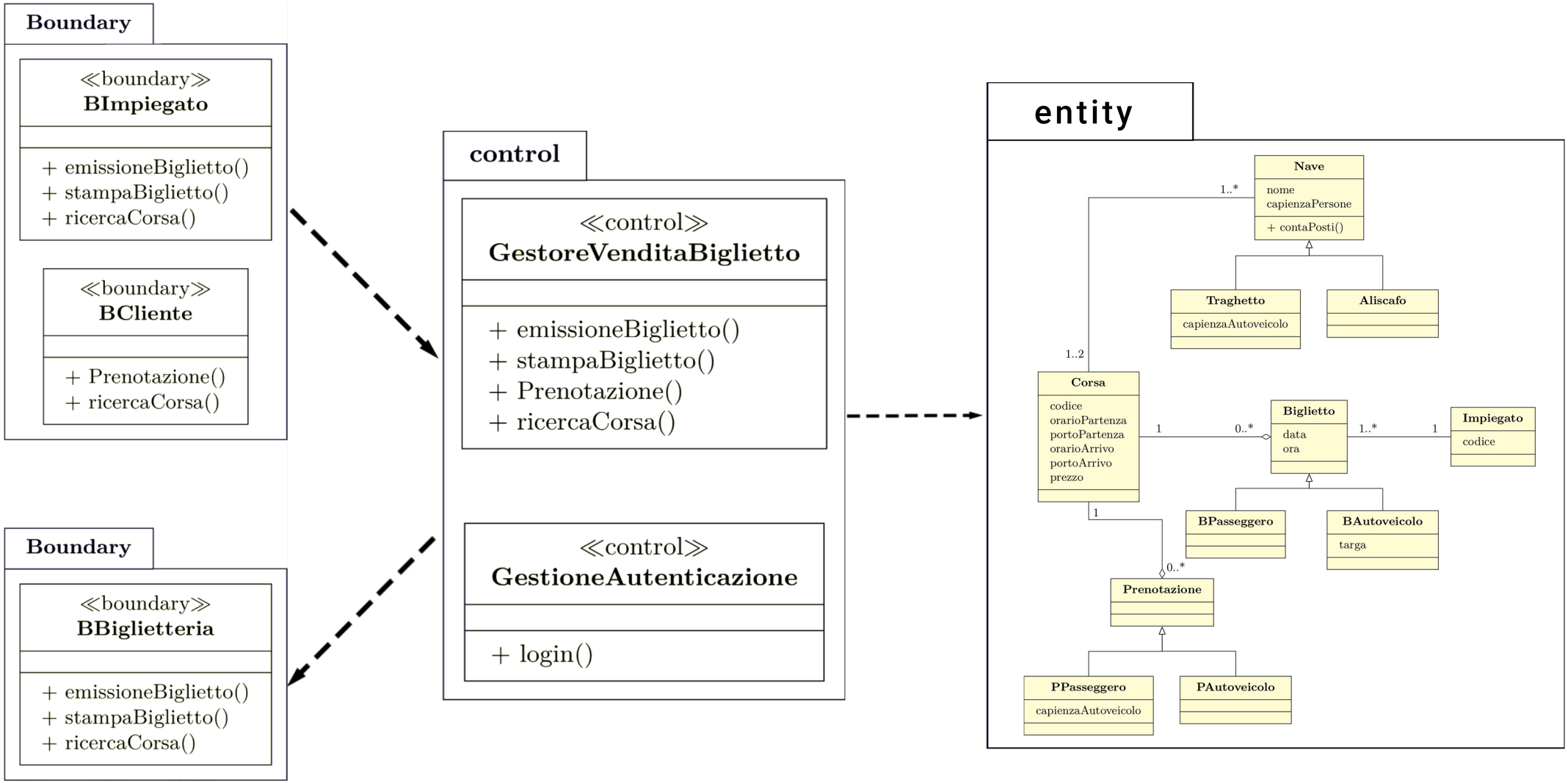
Il diagramma dei casi d'uso è mostrato nella figura che segue



Capitolo 6

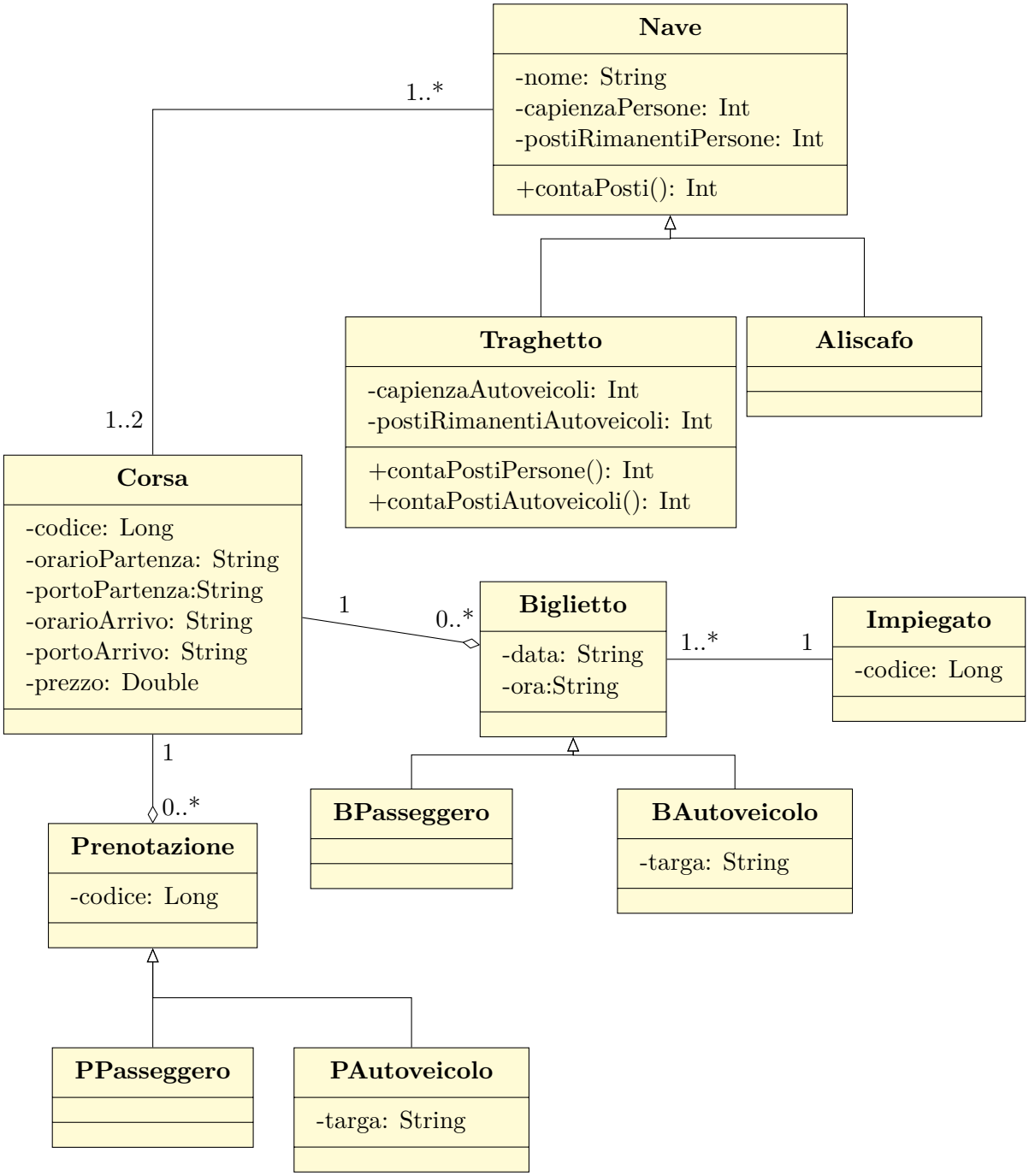
Diagramma delle classi di analisi

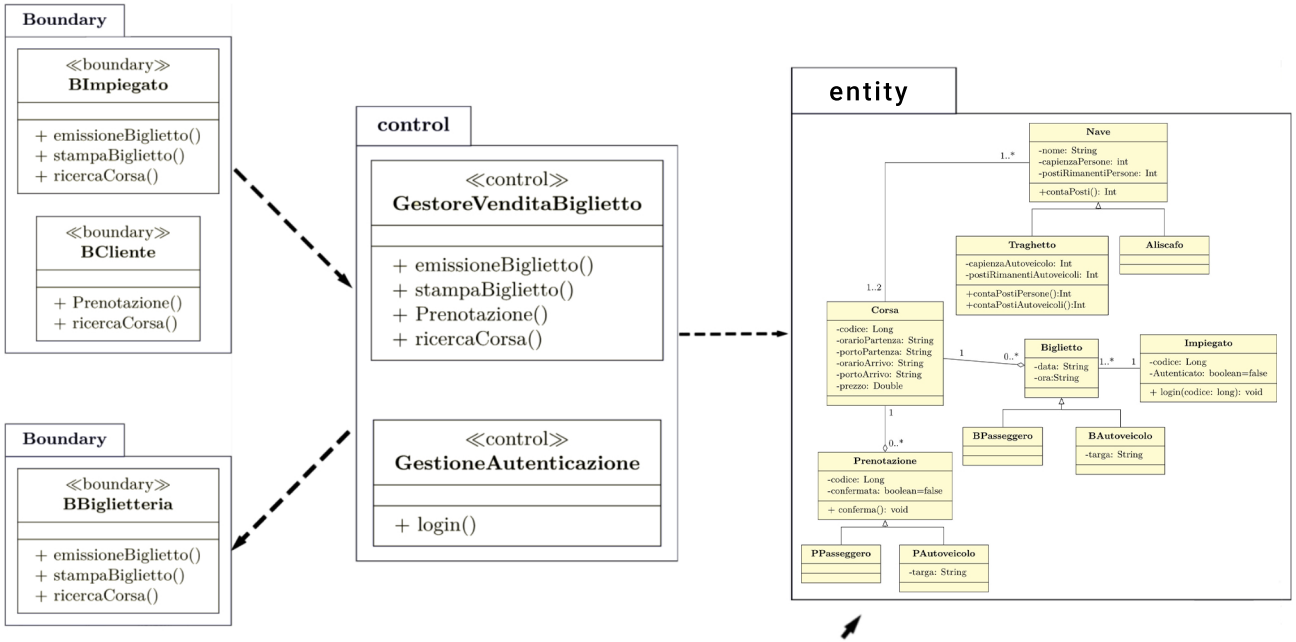


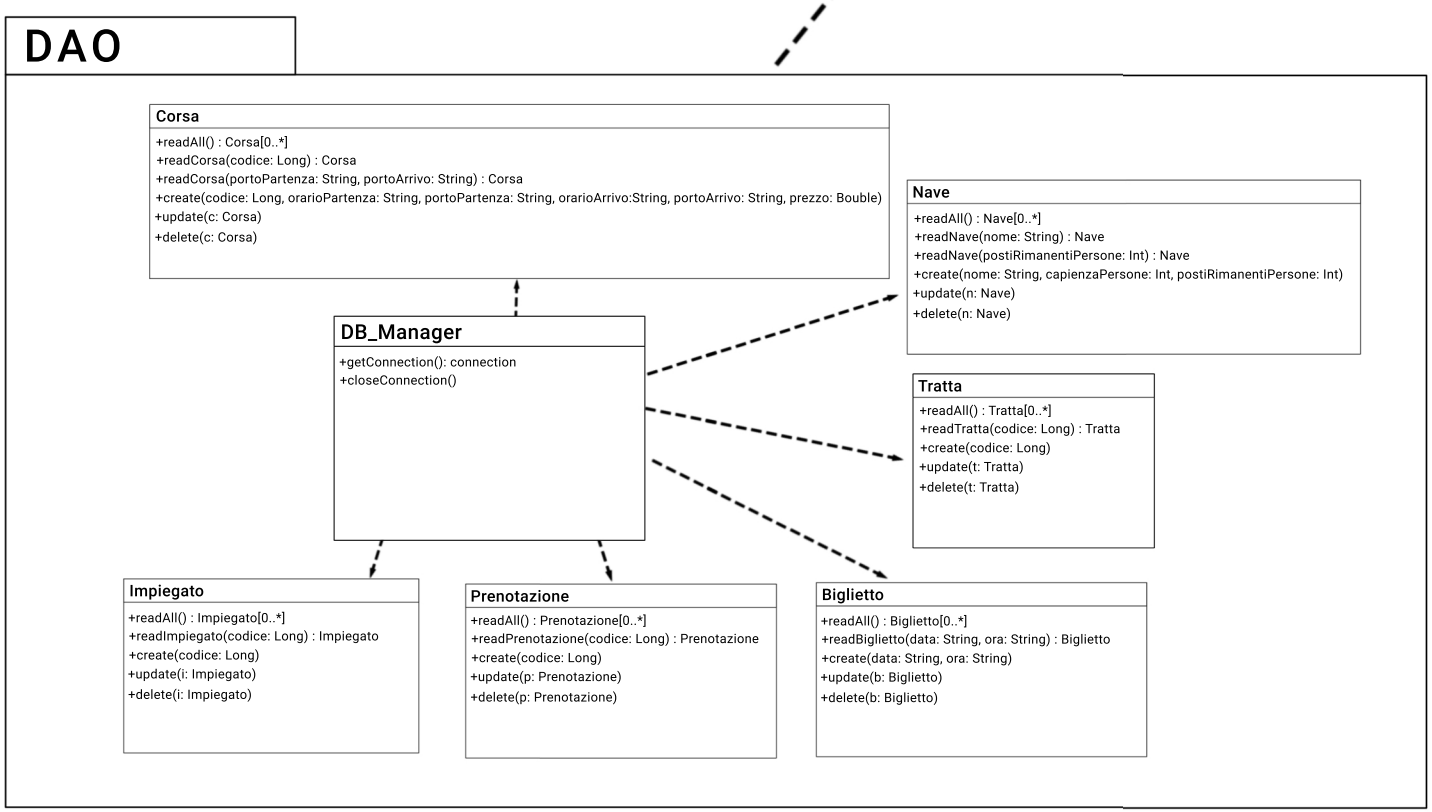

Boundary
≪boundary≫
BImpiegato
+ emissioneBiglietto()
+ stampaBiglietto()
+ ricercaCorsa()
≪boundary≫
BCliente
+ Prenotazione()
+ ricercaCorsa()
Boundary
≪boundary≫
BBiglietteria
+ emissioneBiglietto()
+ stampaBiglietto()
+ ricercaCorsa()
control
≪control≫
GestoreVenditaBiglietto
+ emissioneBiglietto()
+ stampaBiglietto()
+ Prenotazione()
+ ricercaCorsa()
≪control≫
GestioneAutenticazione
+ login()
entity
Nave
nome
capienzaPersone
+ contaPosti()
1..*
Traghetto
capienzaAutoveicolo
Aliscafo
1..2
Corsa
codice
orarioPartenza
portoPartenza
orarioArrivo
portoArrivo
prezzo
1
0..*
Biglietto
data
ora
1..*
1
Impiegato
codice
BPasseggero
BAutoveicolo
targa
1
0..*
Prenotazione
PPasseggero
capienzaAutoveicolo
PAutoveicolo

Capitolo 7

Diagramma delle classi di progettazione




Boundary
≪boundary≫
BImpiegato
+ emissioneBiglietto()
+ stampaBiglietto()
+ ricercaCorsa()
≪boundary≫
BCliente
+ Prenotazione()
+ ricercaCorsa()
Boundary
≪boundary≫
BBiglietteria
+ emissioneBiglietto()
+ stampaBiglietto()
+ ricercaCorsa()
control
≪control≫
GestoreVenditaBiglietto
+ emissioneBiglietto()
+ stampaBiglietto()
+ Prenotazione()
+ ricercaCorsa()
≪control≫
GestioneAutenticazione
+ login()
entity
Nave
-nome: String
-capienzaPersone: int
-postiRimanentiPersone: Int
+contaPosti(): Int
Traghetto
-capienzaAutoveicolo: Int
-postiRimanentiAutoveicoli: Int
+contaPostiPersone():Int
+contaPostiAutoveicoli():Int
Aliscafo
Corsa
-codice: Long
-orarioPartenza: String
-portoPartenza: String
-orarioArrivo: String
-portoArrivo: String
-prezzo: Double
Biglietto
-data: String
-ora:String
Impiegato
-codice: Long
-Autenticato: boolean=false
+ login(codice: long): void
BPasseggero
BAutoveicolo
-targa: String
Prenotazione
-codice: Long
-confermata: boolean=false
+ conferma(): void
PPasseggero
PAutoveicolo
-targa: String
1..*
1..2
1
0..*
1..*
1


DAO
Corsa
+readAll() : Corsa[0..*]
+readCorsa(codice: Long) : Corsa
+readCorsa(portoPartenza: String, portoArrivo: String) : Corsa
+create(codice: Long, orarioPartenza: String, portoPartenza: String, orarioArrivo:String, portoArrivo: String, prezzo: Bouble)
+update(c: Corsa)
+delete(c: Corsa)
Nave
+readAll() : Nave[0..*]
+readNave(nome: String) : Nave
+readNave(postiRimanentiPersone: Int) : Nave
+create(nome: String, capienzaPersone: Int, postiRimanentiPersone: Int)
+update(n: Nave)
+delete(n: Nave)
DB_Manager
+getConnection(): connection
+closeConnection()
Tratta
+readAll() : Tratta[0..*]
+readTratta(codice: Long) : Tratta
+create(codice: Long)
+update(t: Tratta)
+delete(t: Tratta)
Impiegato
+readAll() : Impiegato[0..*]
+readImpiegato(codice: Long) : Impiegato
+create(codice: Long)
+update(i: Impiegato)
+delete(i: Impiegato)
Prenotazione
+readAll() : Prenotazione[0..*]
+readPrenotazione(codice: Long) : Prenotazione
+create(codice: Long)
+update(p: Prenotazione)
+delete(p: Prenotazione)
Biglietto
+readAll() : Biglietto[0..*]
+readBiglietto(data: String, ora: String) : Biglietto
+create(data: String, ora: String)
+update(b: Biglietto)
+delete(b: Biglietto)