

Documentazione progetto

Giulia Sonzogni e Nunzio Marco Bisceglia

Matricole n. 1045979 e 1046319



Università degli Studi di Bergamo

Laurea Magistrale in Ingegneria Informatica

Corso di Informatica III

Modulo di Progettazione e Algoritmi (6 CFU)

Settembre 2021

Indice

1	Iterazione 0	4
1.1	Introduzione e panoramica del sistema	4
1.2	Requisiti funzionali e analisi dei casi d'uso	5
1.2.1	Coda ad alta priorità	6
1.2.2	Coda a media priorità	6
1.2.3	Coda a bassa priorità	6
1.3	Requisiti non funzionali	7
1.3.1	Manutenibilità	7
1.3.2	Efficienza	7
1.3.3	Usabilità	7
1.4	Topologia del sistema	8
1.5	Design pattern MVP	8
2	Iterazione 1	10
2.1	Introduzione	10
2.2	UC1: Gestione barche	10
2.2.1	UC1.1 Visualizzazione barche	11
2.2.2	UC1.2 Inserimento barca	12
2.2.3	UC1.3 Modifica barca	12
2.2.4	UC1.4 Eliminazione barca	13
2.3	UC2: Gestione escursioni	13
2.3.1	UC2.1 Visualizzazione escursioni	14
2.3.2	UC2.2 Inserimento escursione	14
2.3.3	UC2.3 Modifica escursione	15
2.3.4	UC2.4 Eliminazione escursione	16
2.4	UC3: Registrazione dell'utente	16
2.5	UC4: Login dell'utente	17
2.6	UML Component diagram	17
2.7	UML Class diagram per Interfacce	18
2.8	UML Class diagram per tipi di dato	20
2.9	UML Deployment diagram	20
3	Iterazione 2	22
3.1	Introduzione	22
3.2	UC1: Prenotazioni	22
3.2.1	UC1.1 Visualizzazione prenotazioni	23
3.2.2	UC1.2 Nuova prenotazione	23
3.2.3	UC1.3 Modifica prenotazione	24

3.2.4	UC1.4 Elimina prenotazione	24
3.3	UC2: Algoritmo barche ed escursioni	25
3.4	UML Component Diagram	26
3.5	UML Class Diagram per interfacce	26
3.6	UML Class Diagram per tipi di dato	26
3.7	UML Deployment diagram	26

Elenco delle figure

1	Diagramma UML dei casi d'uso	5
2	Topologia del sistema.	9
3	Diagramma dei componenti UML.	18
4	Diagramma delle classi per interfacce UML.	19
5	Diagramma delle classi per tipi di dato UML.	20
6	Deployment diagram UML.	21
7	Diagramma dei componenti UML.	26
8	Diagramma delle classi per interfacce UML.	27
9	Tipo di dato Booking	28
10	Deployment diagram UML.	28

Elenco delle tabelle

1	Casi d'uso con priorità alta	6
2	Casi d'uso con priorità media	6
3	Casi d'uso con priorità bassa	6

1 Iterazione 0

1.1 Introduzione e panoramica del sistema

Il sistema che verrà implementato in questo progetto di studio si occuperà della gestione di un centro di immersioni, noto anche come *diving center* o *dive center*. Un centro di immersione è una struttura che fornisce supporto, attrezzatura e corsi per la pratica delle attività subacquee. Un centro di immersione fornisce principalmente tre tipi di servizi:

- Immersioni guidate.
- Noleggio attrezzatura.
- Scuola di immersione.

Per immersione guidata si intende un'immersione effettuata in gruppo o singolarmente insieme a guide subacquee addestrate, esperte e a conoscenza dei punti di immersione più importanti ed interessanti. Questa soluzione è ottimale quando ci si vuole immergere in luoghi non conosciuti o in parchi marini protetti per cui è necessaria un'autorizzazione e una guida. Inoltre, il centro diving offre i servizi di accompagnamento e assistenza, di trasposto con imbarcazioni o gommoni, che permettono l'entrata e l'uscita dall'acqua con praticità e comfort.

Il focus del nostro progetto sarà la gestione e l'organizzazione efficiente delle immersioni guidate e la gestione del noleggio attrezzatura; non verrà incluso l'insegnamento della pratica subacquea.

Un centro di immersione ha una disponibilità limitata di barche e guide per cui ha la possibilità di effettuare un numero limitato di escursioni durante una giornata. Il sistema che verrà sviluppato si occuperà di organizzare al meglio l'allocazione delle risorse disponibili affinché un numero maggiore di persone possa parteciparvi. Spesso le persone che vogliono partecipare a questo tipo di escursioni si trovano in gruppo, per cui è necessario ottimizzare l'allocazione delle risorse disponibili, che in questo caso costituiscono i posti disponibili in barca, con il vincolo di mantenere il più possibile intatti i gruppi.

Il sistema implementato rappresenterà un centro di immersione: si occuperà della gestione delle risorse (barche, escursioni, attrezzatura disponibile) e della gestione delle prenotazioni delle escursioni da parte degli utenti.

1.2 Requisiti funzionali e analisi dei casi d'uso

In questa sezione verranno introdotti i requisiti funzionali del sistema attraverso l'utilizzo dei casi d'uso.

Lo schema UML dei casi d'uso viene riportato in Figura 1.

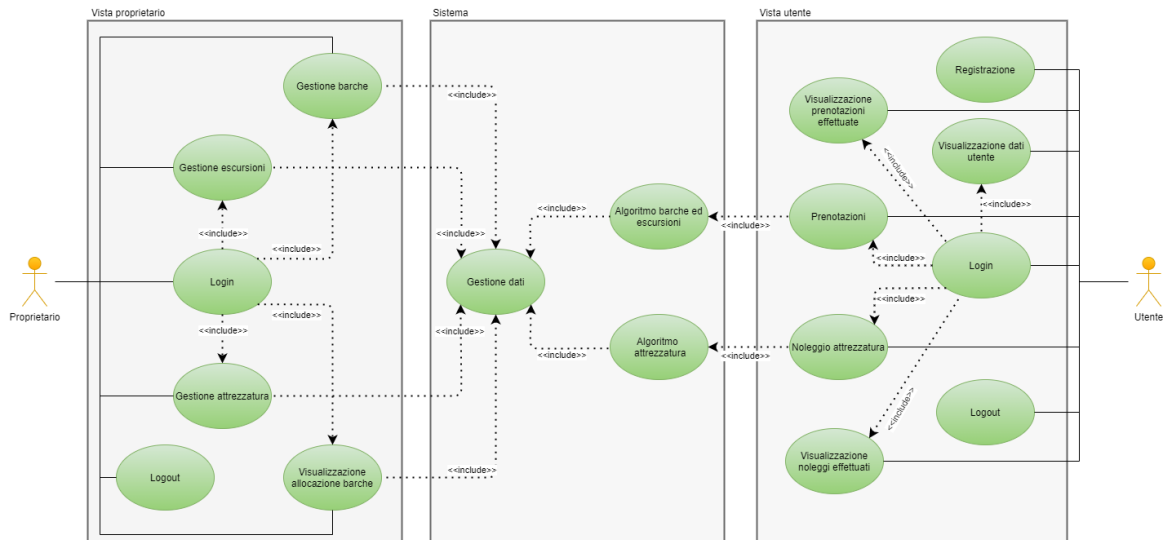


Figura 1: Diagramma UML dei casi d'uso

Al fine di procedere ad uno sviluppo efficiente, è stato deciso di dividere le specifiche funzionali in tre code di priorità: alta, media e bassa. Nella coda ad alta priorità si troveranno i casi d'uso indispensabili al corretto funzionamento dell'applicazione, nella coda a media priorità saranno inseriti i casi d'uso riguardanti le funzionalità aggiuntive e nella coda a bassa priorità le funzionalità non strettamente necessarie e che verranno implementate in versioni future.

1.2.1 Coda ad alta priorità

Codice	Titolo
UC1	Gestione barche (visualizzazione, inserimento, modifica, elimina)
UC2	Gestione escursioni (visualizzazione, inserimento, modifica, elimina)
UC3	Registrazione utente
UC4	Login utente
UC5	Logout utente
UC6	Gestione prenotazioni (visualizzazione, inserimento)
UC7	Algoritmo gestione barche ed escursioni

Tabella 1: Casi d'uso con priorità alta

1.2.2 Coda a media priorità

Codice	Titolo
UC8	Visualizzazione allocazione barche
UC9	Gestione attrezzatura (visualizzazione, inserimento, modifica, elimina)
UC10	Noleggio attrezzatura
UC11	Algoritmo attrezzatura

Tabella 2: Casi d'uso con priorità media

1.2.3 Coda a bassa priorità

Codice	Titolo
UC12	Visualizzazione dati utente
UC13	Modifica dati utente
UC14	Visualizzazione prenotazioni effettuate
UC15	Visualizzazione noleggi effettuati

Tabella 3: Casi d'uso con priorità bassa

1.3 Requisiti non funzionali

Il progetto verrà sviluppato tenendo in considerazione anche alcuni requisiti non funzionali, quali la manutenibilità, l'efficienza e l'usabilità.

1.3.1 Manutenibilità

Il requisito di manutenibilità verrà rispettato mediante l'implementazione di tutte le risorse persistenti, ovvero barche, orari e date delle escursioni, attrezzatura. In questo modo, anche se in futuro dovessero cambiare alcune condizioni, sarebbe facile per il proprietario rifletterle all'interno del software.

1.3.2 Efficienza

Il requisito dell'efficienza è anche l'obiettivo primario del progetto, ovvero l'allocazione ottimale delle risorse disponibili: i posti presenti nelle barche.

1.3.3 Usabilità

L'usabilità è garantita dalla decisione di sviluppare il programma attraverso un'applicazione Android, di facile utilizzo sia per gli utenti che per il proprietario. In base alla topologia (Figura 2) è possibile notare che il database verrà ospitato da un hosting online e quindi sarà sempre possibile, mediante cellulare, accedere a tutte le funzionalità dell'applicazione.

1.4 Topologia del sistema

La topologia del sistema mostrata in Figura 2, evidenzia il requisito non funzionale dell'usabilità: sia gli utenti che il proprietario possono accedere al servizio tramite l'applicazione. Il WebServer espone un set di API mediante HTTP/REST con cui è possibile accedere alle funzionalità dell'applicazione. Infine, il WebServer è collegato ad un database relazionale.

Questo progetto è stato sviluppato utilizzando un'architettura Three-tier, ovvero un'architettura hardware di tipo multi-tier. Essa prevede la suddivisione dell'applicazione in tre diversi moduli, i quali risiedono su macchine separate, dedicati rispettivamente a:

- Presentation, interfaccia utente
- Application, logica funzionale
- Data, gestione dei dati persistenti.

È possibile visualizzare questa divisione in Figura 2.

1.5 Design pattern MVP

Il sistema e i suoi componenti verranno sviluppati utilizzando il design pattern architetturale Model View Presenter (MVP), una derivazione del Model View Controller (MVC). MVP è composto da tre elementi:

- Model, che definisce i dati da visualizzare e modificare, può essere visto come un'interfaccia responsabile di accedere alle API connesse con il database.
- View, che visualizza i dati e notifica le azioni dell'utente. Solitamente non presenta nessuna logica applicativa, ma ha una funzione solo visuale.
- Presenter, che funziona come man-in-the-middle tra la View e il Model, prendendo i dati dal Model e formattandoli per la View.

A differenza del MVC, la View non è direttamente collegata al Model. L'interazione può avvenire solo tramite il Presenter. Questo permette uno sviluppo indipendente dei tre livelli. Tale separazione è visibile anche nelle componenti software, come si vede in Figura 3.

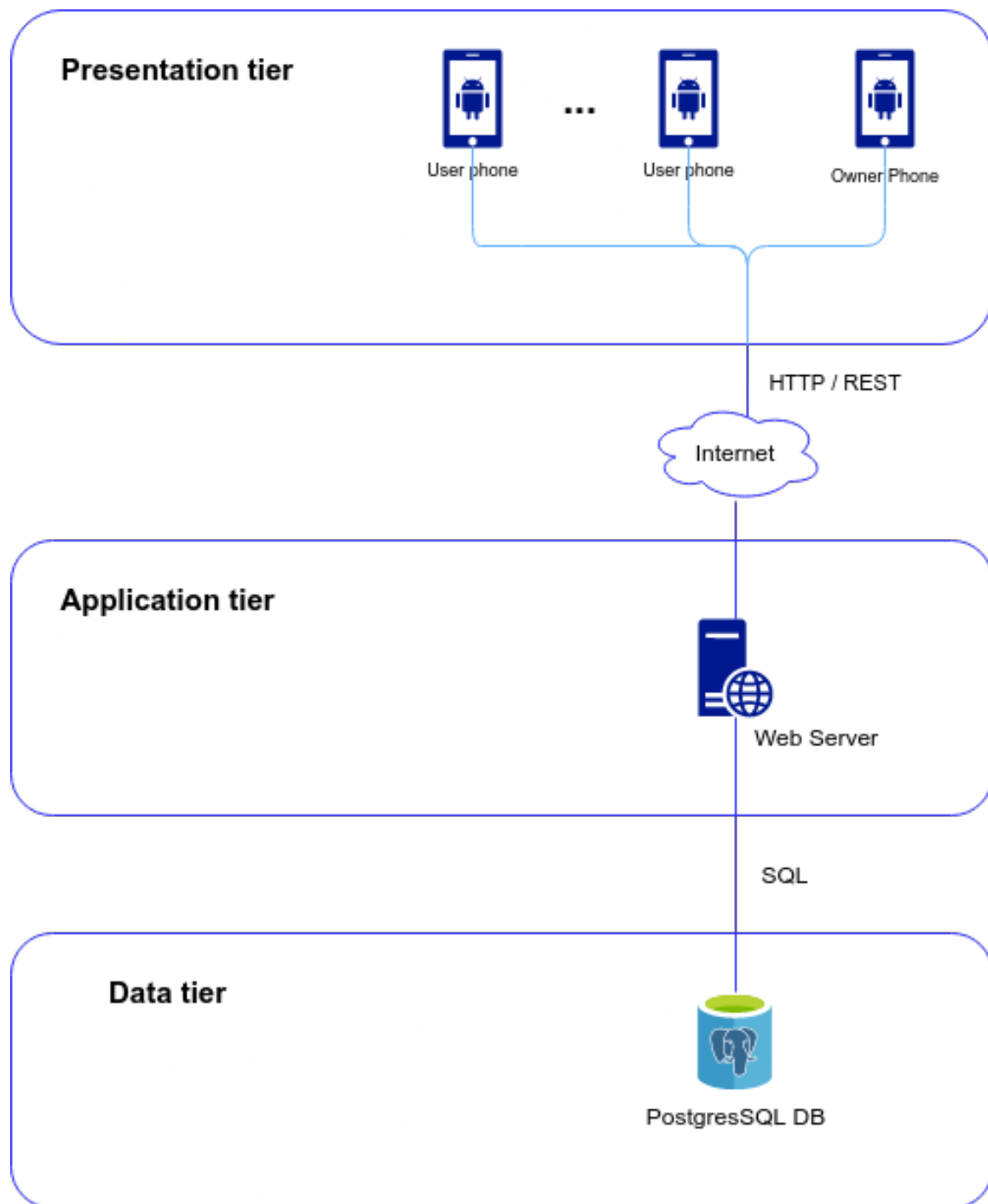


Figura 2: Topologia del sistema.

2 Iterazione 1

2.1 Introduzione

Nella prima iterazione si è scelto di implementare i seguenti casi d'uso:

- UC1: Gestione barche [*Astratto*]
 - UC1.1 Visualizzazione barche
 - UC1.2 Inserimento barca
 - UC1.3 Modifica barca
 - UC1.4 Eliminazione barca
- UC2: Gestione escursioni [*Astratto*]
 - UC2.1 Visualizzazione escursioni
 - UC2.2 Inserimento escursione
 - UC2.3 Modifica escursione
 - UC2.4 Eliminazione escursione
- UC3: Registrazione utente
- UC4: Login utente
- UC5: Logout utente

Dopo averli descritti in maniera testuale si è passati alla realizzazione del diagramma dei componenti e del deployment diagram. Infine il codice è stato implementato sulla base di questi documenti.

2.2 UC1: Gestione barche

Breve descrizione: il proprietario del diving center deve avere una visione generale e un controllo completo delle sue barche all'interno dell'applicazione. Oltre alla visualizzazione, il proprietario deve poter modificare ed eliminare le imbarcazioni già presenti nel sistema e poter aggiungerne di nuove. La gestione delle barche è quindi suddivisa in 4 casi d'uso concreti:

- UC1.1 Visualizzazione barche
- UC1.2 Inserimento barca

- UC1.3 Modifica barca
- UC1.4 Eliminazione barca

Attori coinvolti: Proprietario, Sistema.

Trigger: login del proprietario avvenuto con successo.

Postcondizione: il proprietario dopo il login, viene indirizzato alla home page e può scegliere se cliccare uno dei button: *Visualizza barche* o *Aggiungi barca*. Il primo per visualizzare tutte le barche inserite, il secondo per aggiungerne una nuova.

Procedimento:

1. Proprietario effettua il login.
2. Sistema mostra la pagina iniziale che comprende un button *Visualizza barche* e un button *Aggiungi barca* con cui il proprietario può concretamente gestire le sue barche.

2.2.1 UC1.1 Visualizzazione barche

Breve descrizione: Il proprietario visualizza le barche inserite nel sistema e le relative caratteristiche.

Attori coinvolti: Proprietario, Sistema.

Trigger: Il proprietario clicca il button *Visualizza barche*.

Postcondizione: Il sistema mostra una vista con l'elenco delle barche.

Procedimento:

1. Proprietario effettua il login
2. Proprietario clicca il button *Visualizza barche* presente nella home page.
3. Sistema mostra l'elenco delle barche presenti nel database con le seguenti informazioni:
 - Nome della barca
 - Modello della barca
 - Numero posti disponibili sulla barca
 - Altre caratteristiche TODO
4. Per ogni item dell'elenco il proprietario può cliccare su *Modifica barca* o *Elimina barca* per modificare o eliminare una barca.

2.2.2 UC1.2 Inserimento barca

Breve descrizione: Il proprietario aggiunge una barca nel sistema. Questo viene fatto principalmente nella fase iniziale in cui il proprietario deve inserire tutte le sue barche all'interno dell'applicazione e poi in seguito all'acquisto di nuove barche.

Attori coinvolti: Proprietario, Sistema.

Trigger: Il proprietario clicca il button *Inserisci barca*.

Postcondizione: Il sistema mostra la nuova barca all'interno della pagina di visualizzazione delle barche.

Procedimento:

1. Il proprietario si trova sulla home page e clicca il button *Inserisci barca*.
2. Sistema mostra un form in cui è possibile inserire i dati relativi alla barca.
3. Proprietario riempie il form.
4. Proprietario clicca su *Conferma* per inserire la barca nel sistema.
5. Il sistema mostra la nuova barca all'interno della pagina di visualizzazione delle barche.

2.2.3 UC1.3 Modifica barca

Breve descrizione: Il proprietario si trova sulla pagina di visualizzazione delle barche e clicca su un button o un'icona di modifica di una delle barche. Tale modifica può essere dovuta a cambiamenti strutturali della barca, per esempio la riduzione del numero di posti disponibili.

Attori coinvolti: Proprietario, Sistema.

Trigger: Il proprietario clicca il button o l'icona *Modifica barca*.

Postcondizione: Il sistema mostra la barca modificata all'interno della pagina di visualizzazione delle barche.

Procedimento:

1. Il proprietario si trova sulla pagina di visualizzazione delle barche e clicca su un button o un'icona di modifica di una delle barche.
2. Sistema mostra una vista in cui è possibile modificare i dati relativi alla barca.
3. Proprietario modifica i dati.

4. Proprietario clicca su *Conferma* per modificare i dati della barca.
5. Il sistema mostra la barca aggiornata all'interno della pagina di visualizzazione delle barche.

2.2.4 UC1.4 Eliminazione barca

Breve descrizione: Il proprietario si trova sulla pagina di visualizzazione delle barche e clicca su un button o un'icona di eliminazione di una delle barche. L'eliminazione può essere definitiva o temporanea. L'eliminazione è definitiva se la barca non è più agibile o perché viene sostituita da un'altra; è temporanea nel caso in cui sia necessaria attività di manutenzione.

Attori coinvolti: Proprietario, Sistema.

Trigger: Il proprietario clicca il button o l'icona *Elimina barca*.

Postcondizione: Il sistema mostra la pagina di visualizzazione delle barche in cui non comparirà la barca eliminata.

Procedimento:

1. Il proprietario si trova sulla pagina di visualizzazione delle barche e clicca su un button o un'icona di eliminazione di una delle barche. Il proprietario deve inoltre scegliere se l'eliminazione è temporanea o definitiva.
2. Sistema mostra un alert che avvisa il proprietario che l'azione è irreversibile.
3. Il proprietario può scegliere se confermare l'eliminazione cliccando su *Conferma* o annullare l'azione cliccando su *Annulla*.
4. Il sistema mostra la pagina di visualizzazione delle barche in cui non comparirà la barca eliminata se il proprietario ha scelto di eliminarla in maniera non temporanea. Se invece, la barca è stata eliminata temporaneamente, essa sarà sempre visibile nella pagina di visualizzazione delle barche, ma segnalando al proprietario che il sistema organizzerà le escursioni senza tenere conto di quella barca.

2.3 UC2: Gestione escursioni

L'applicazione deve poter semplificare la gestione e l'organizzazione delle escursioni al proprietario del diving center. L'applicazione deve fungere da calendario, mettendo a disposizione le seguenti funzionalità:

- UC2.1 Visualizzazione escursioni
- UC2.2 Inserimento escursione
- UC2.3 Modifica escursione
- UC2.4 Eliminazione escursione

2.3.1 UC2.1 Visualizzazione escursioni

Breve descrizione: Il proprietario visualizza le date e i turni in cui gli utenti possono prenotarsi.

Attori coinvolti: Proprietario, Sistema.

Trigger: Il proprietario clicca il button *Visualizza escursioni*.

Postcondizione: Il sistema mostra una vista con l'elenco delle escursioni.

Procedimento:

1. Proprietario effettua il login.
2. Proprietario clicca il button *Visualizza escursioni* presente nella home page.
3. Sistema mostra l'elenco delle escursioni rese disponibili con le seguenti informazioni:
 - Data dell'escursione.
 - Orario di inizio escursione.
 - Orario di fine escursione.
 - Altro TODO
4. Per ogni data presente nell'elenco il proprietario può cliccare su *Modifica escursione* o *Elimina escursione* per modificare o eliminare un'escursione.

2.3.2 UC2.2 Inserimento escursione

Breve descrizione: Il proprietario aggiunge un'escursione al calendario indicandone data e orari in cui verranno rese disponibili le prenotazioni.

Attori coinvolti: Proprietario, Sistema.

Trigger: Il proprietario clicca il button *Inserisci escursione*.

Postcondizione: Il sistema mostra la nuova escursione all'interno della pagina di visualizzazione delle escursioni.

Procedimento:

1. Il proprietario si trova sulla home page e clicca il button *Inserisci escursione*.
2. Sistema mostra un form in cui è possibile inserire i dati relativi all'escursione.
3. Proprietario riempie il form.
4. Proprietario clicca su *Conferma* per inserire l'escursione nel sistema.
5. Il sistema mostra la nuova escursione all'interno della pagina di visualizzazione delle escursioni.

2.3.3 UC2.3 Modifica escursione

Breve descrizione: Il proprietario si trova sulla pagina di visualizzazione delle escursioni e clicca su un button o un'icona di modifica di una delle escursioni. Tale modifica può essere dovuta a cambiamenti climatici o ad altri imprevisti.

Attori coinvolti: Proprietario, Sistema.

Trigger: Il proprietario clicca il button o l'icona *Modifica escursione*.

Postcondizione: Il sistema mostra l'escursione modificata all'interno della pagina di visualizzazione delle escursioni.

Procedimento:

1. Il proprietario si trova sulla pagina di visualizzazione delle escursioni e clicca su un button o un'icona di modifica di una delle escursioni.
2. Sistema mostra una vista in cui è possibile modificare i dati relativi all'escursione.
3. Proprietario modifica i dati.
4. Proprietario clicca su *Conferma* per modificare i dati dell'escursione.
5. Il sistema mostra l'escursione aggiornata all'interno della pagina di visualizzazione delle escursioni.

2.3.4 UC2.4 Eliminazione escursione

Breve descrizione: Il proprietario si trova sulla pagina di visualizzazione delle escursioni e clicca su un button o un'icona di eliminazione di una delle escursioni.

Attori coinvolti: Proprietario, Sistema.

Trigger: Il proprietario clicca il button o l'icona *Elimina escursione*.

Postcondizione: Il sistema mostra la pagina di visualizzazione delle escursioni in cui non comparirà l'escursione eliminata.

Procedimento:

1. Il proprietario si trova sulla pagina di visualizzazione delle escursioni e clicca su un button o un'icona di eliminazione di una delle escursioni.
2. Sistema mostra un alert che avvisa il proprietario che l'azione è irreversibile.
3. Il proprietario può scegliere se confermare l'eliminazione cliccando su *Conferma* o annullare l'azione cliccando su *Annulla*.
4. Il sistema mostra la pagina di visualizzazione delle escursioni in cui non comparirà l'escursione eliminata.

2.4 UC3: Registrazione dell'utente

Breve descrizione: L'utente compila il form per la registrazione all'app e, se non si è già registrato, viene aggiunto al database.

Attori coinvolti: Utente, Sistema?

Trigger: L'utente preme su "Registrazione".

Postcondizione: L'utente è stato inserito nel database e ha ricevuto la conferma dell'operazione.

Procedimento:

1. Utente preme su "Registrazione" nella pagina iniziale dell'app
2. Utente fornisce Nome, username, indirizzo email e password nel form di registrazione
3. Il sistema verifica se esiste già un utente con quella mail:
 - (a) se esiste già, il sistema lo comunica all'utente
 - (b) se non esiste allora il sistema aggiunge l'utente nel database

2.5 UC4: Login dell'utente

Breve descrizione: L'utente (utente normale o il proprietario) compila il form per il login e se le credenziali sono giuste il sistema gli consente l'accesso.

Attori coinvolti: Utente/Proprietario, Sistema?

Trigger: L'utente preme su "Login"

Postcondizione: L'utente ha accesso alla vista del suo profilo.

Procedimento:

1. Utente preme su "Login" nella pagina iniziale dell'app
2. Utente fornisce username e password nel form di registrazione
3. Il sistema controlla le credenziali inserite:
 - (a) se sono corrette, il sistema invia una conferma di accesso all'utente
 - (b) se sono errate, il sistema lo comunica all'utente

2.6 UML Component diagram

I casi d'uso scelti in questa prima iterazione vengono rappresentati sottoforma di componenti nel diagramma in Figura 3. Si è scelto di suddividere i caso d'uso in componenti:

- «boundary» rappresentati dai componenti lato front-end con cui gli attori si interfacciano direttamente. Tali componenti richiedono delle interfacce al back-end.
- «control» rappresentati dai componenti lato back-end che forniscono delle API al front-end, richiedonone a loro volta al database.
- «data» rappresentato dal database in cui verranno memorizzati i dati delle barche, il calendario delle escursioni e i dati dell'utente in occasione della registrazione.

TODO spiega che usi MVC e 3-tier.

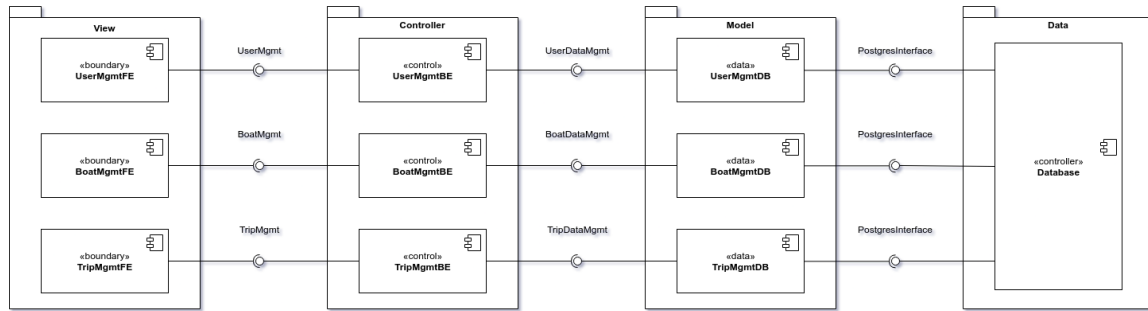


Figura 3: Diagramma dei componenti UML.

2.7 UML Class diagram per Interfacce

Il diagramma in Figura 4 serve a rappresentare le interfacce del sistema e le loro interazioni.

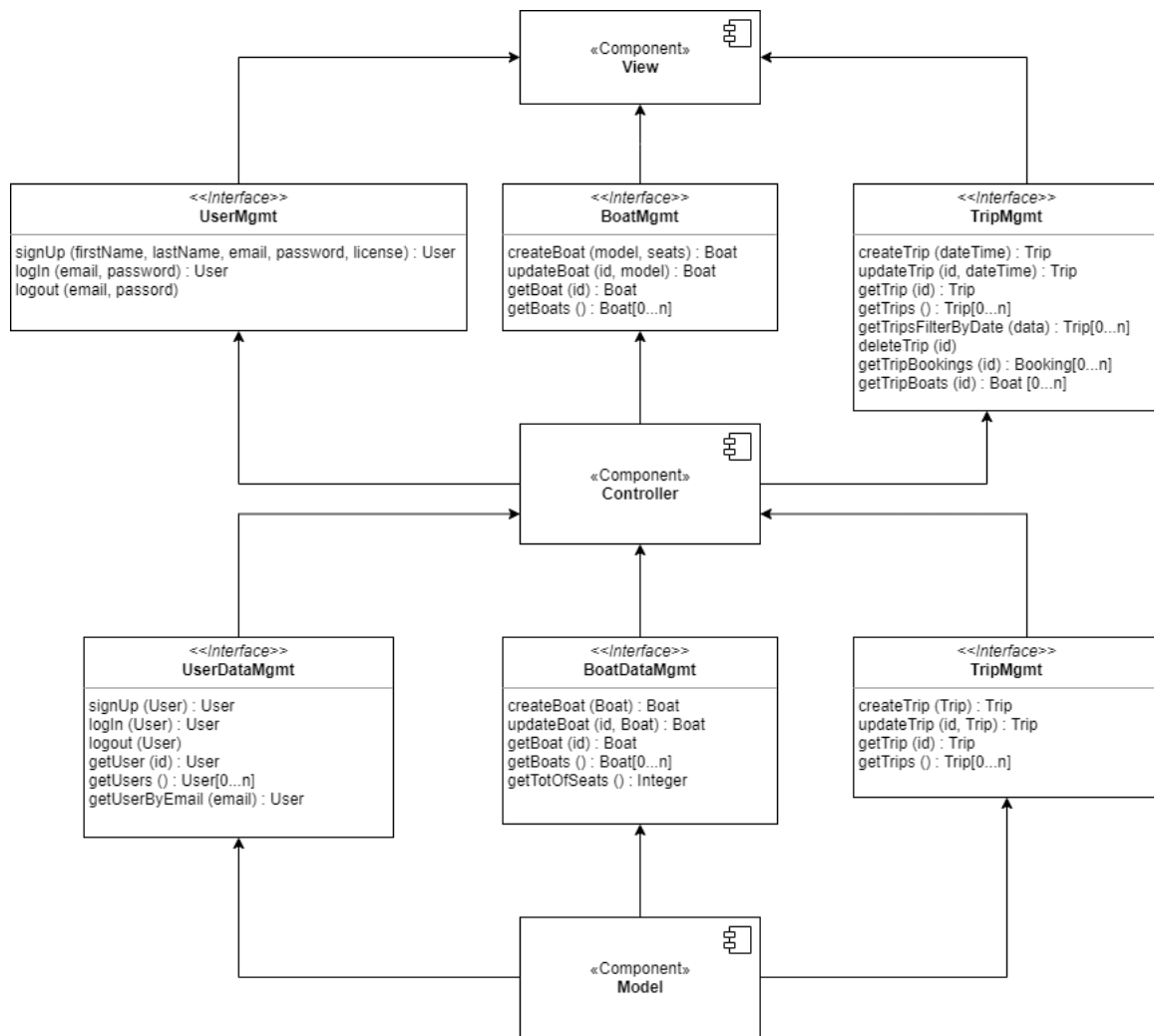


Figura 4: Diagramma delle classi per interfacce UML.

2.8 UML Class diagram per tipi di dato

Il diagramma in Figura 5 serve ad esplicitare tipi di dato particolari e i loro legami.

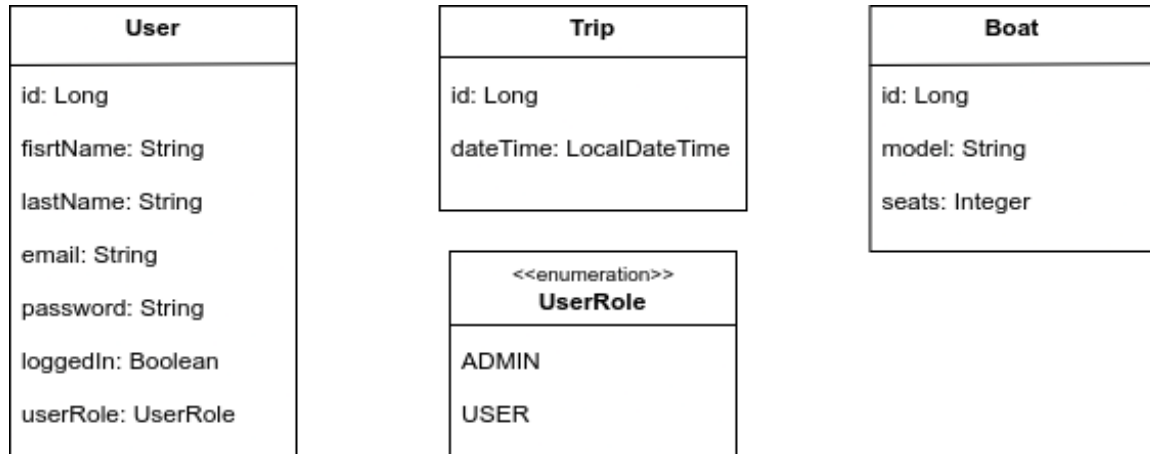


Figura 5: Diagramma delle classi per tipi di dato UML.

2.9 UML Deployment diagram

I componenti descritti precedentemente vengono istanziati nel Deployment diagram. In Figura 6 vengono mostrati i componenti contenuti nei seguenti nodi:

- Cellulare proprietario è il nodo su cui l'admin del sistema potrà gestire le barche e il calendario delle escursioni.
- Cellulare utente è il nodo su cui una persona potrà registrarsi alla piattaforma.
- Web server fornisce le API richieste dall'applicativo lato front-end.
- Database funge da storage dei dati.

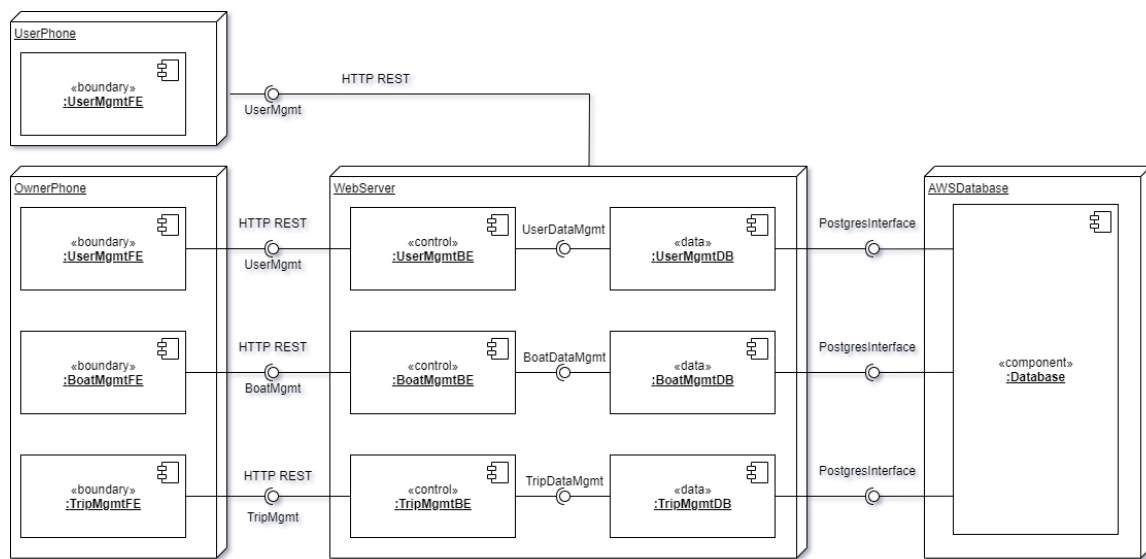


Figura 6: Deployment diagram UML.

3 Iterazione 2

3.1 Introduzione

Nella seconda iterazione si è deciso di implementare i seguenti casi d'uso:

- UC1 Prenotazioni [Astratto]
 - UC1.1 Nuova prenotazione
 - UC1.2 Visualizzazione prenotazioni
 - UC1.3 Modifica prenotazione
 - UC1.4 Elimina prenotazione
- UC2 Algoritmo barche ed escursioni

Seguono una descrizione testuale di questi casi e i diagrammi dei componenti e di deployment.

3.2 UC1: Prenotazioni

Breve descrizione: L'utente, dopo essersi loggato, può effettuare la prenotazione per una delle escursioni organizzate dal proprietario. Gli è possibile prenotare un posto non solo per sé stesso ma anche per altre persone, indicando il numero di partecipanti totale. Oltre alla creazione di una prenotazione, l'utente deve poter visualizzare le prenotazioni effettuate ed eventualmente modificarle o eliminarle. Il caso d'uso Prenotazioni è suddiviso in 4 casi d'uso concreti:

- UC1.1 Visualizzazione prenotazioni
- UC1.2 Nuova prenotazione
- UC1.3 Modifica prenotazione
- UC1.4 Elimina prenotazione

Attori coinvolti: Utente, Sistema.

Trigger: login dell'utente avvenuto con successo.

Postcondizione: l'utente dopo il login viene indirizzato alla home page e può scegliere se cliccare uno dei bottoni: Nuova prenotazione o Visualizza prenotazioni.

Procedimento:

1. l'utente effettua il login
2. Il sistema mostra la pagina iniziale che comprende i bottoni Visualizza e Nuova Prenotazione con cui l'utente può gestire le sue prenotazioni.

3.2.1 UC1.1 Visualizzazione prenotazioni

Breve descrizione: L'utente visualizza le prenotazioni effettuate e i relativi dati.

Attori coinvolti: Utente, Sistema.

Trigger: L'utente clicca il bottone Visualizza.

Postcondizione: Il sistema mostra una vista con l'elenco delle prenotazioni.

Procedimento:

1. L'utente effettua il login
2. L'utente clicca il bottone Visualizza presente nella home page.
3. Il sistema mostra l'elenco delle prenotazioni presenti nel database a suo nome con le seguenti informazioni:
 - id prenotazione
 - id escursione
 - data e ora escursione
 - numero di persone per cui si è prenotato
 - disposizione persone sulle barche (idbarca e numPersone)
4. Per ogni item dell'elenco l'utente può cliccare su Modifica prenotazione o Elimina prenotazione

3.2.2 UC1.2 Nuova prenotazione

Breve descrizione: L'utente effettua una nuova prenotazione per un'escursione organizzata dal proprietario.

Attori coinvolti: Utente, Sistema.

Trigger: L'utente clicca il bottone Nuova prenotazione.

Postcondizione: Il sistema mostra la nuova prenotazione all'interno della pagina di visualizzazione delle prenotazioni.

Procedimento:

1. L'utente si trova sulla home page e clicca il bottone Nuova prenotazione.
2. Il sistema mostra un form in cui è possibile specificare i dati relativi alla prenotazione.

3. L'utente riempie il form e clicca il pulsante Conferma.
4. Il sistema mostra la prenotazione effettuata all'interno della pagina di visualizzazione delle prenotazioni.

3.2.3 UC1.3 Modifica prenotazione

Breve descrizione: L'utente si trova sulla pagina di visualizzazione delle prenotazioni e clicca sul bottone Modifica di una di queste. Tale modifica può essere effettuata per aumentare o ridurre il numero di persone legate alla prenotazione (possibile fino a 3gg prima dell'escursione).

Attori coinvolti: Utente, Sistema.

Trigger: L'utente clicca su un bottone Modifica prenotazione.

Postcondizione: Il sistema mostra la prenotazione modificata all'interno della pagina di visualizzazione delle prenotazioni.

Procedimento:

1. L'utente si trova sulla pagina di visualizzazione delle prenotazioni e clicca su un bottone Modifica.
2. Il sistema mostra una vista in cui è possibile modificare i dati relativi alla prenotazione.
3. L'utente modifica i dati e clicca su Conferma.
4. Il sistema mostra la prenotazione aggiornata all'interno della pagina di visualizzazione delle prenotazioni.

3.2.4 UC1.4 Elimina prenotazione

Breve descrizione: L'utente si trova sulla pagina di visualizzazione delle prenotazioni e clicca sul bottone Elimina di una di queste. Dopo aver confermato la scelta, la prenotazione viene eliminata definitivamente.

Attori coinvolti: Utente, Sistema.

Trigger: L'utente clicca su un bottone Elimina prenotazione.

Postcondizione: Il sistema mostra la pagina di visualizzazione delle prenotazioni in cui non comparirà la prenotazione eliminata.

Procedimento:

1. L'utente si trova sulla pagina di visualizzazione delle prenotazioni e clicca su un bottone Elimina.
2. Il sistema mostra una finestra di dialogo per chiedere conferma.
3. L'utente clicca su Conferma o Annulla.
4. In caso di conferma il sistema mostra la pagina di visualizzazione delle prenotazioni senza la prenotazione appena eliminata.

3.3 UC2: Algoritmo barche ed escursioni

Breve descrizione: L'utente ha richiesto di effettuare una prenotazione specificando l'escursione e il numero di persone nel proprio gruppo. Il sistema deve allocare in modo ottimale sulle barche i posti appena richiesti e quelli di altri utenti già prenotati. Lo scopo dell'algoritmo è quello di minimizzare il numero di barche impiegate per l'escursione, così da ridurre i costi del proprietario, e cercare di non separare i gruppi.

Attori coinvolti: Utente, Sistema.

Trigger: L'utente conferma la richiesta di prenotazione.

Postcondizione: La disposizione dei posti dell'escursione è stata aggiornata e l'utente riceve una conferma di prenotazione.

Procedimento:

1. L'utente clicca su conferma prenotazione.
2. Il sistema esegue l'algoritmo di allocazione. Viene osservata la disposizione attuale dei gruppi:
 - Se il numero di posti liberi è minore delle persone del nuovo gruppo viene allocata una nuova barca
 - Se possibile, il gruppo viene allocato interamente su una delle barche già presenti
 - ...
3. La nuova disposizione viene aggiornata nel database
4. Viene mandata una conferma di prenotazione all'utente.

3.4 UML Component Diagram

I casi d'uso scelti in questa prima iterazione vengono rappresentati sottoforma di componenti nel diagramma in Figura 7. Come per l'iterazione precedente, si è scelto di suddividere i casi d'uso in componenti boundary, control e data.

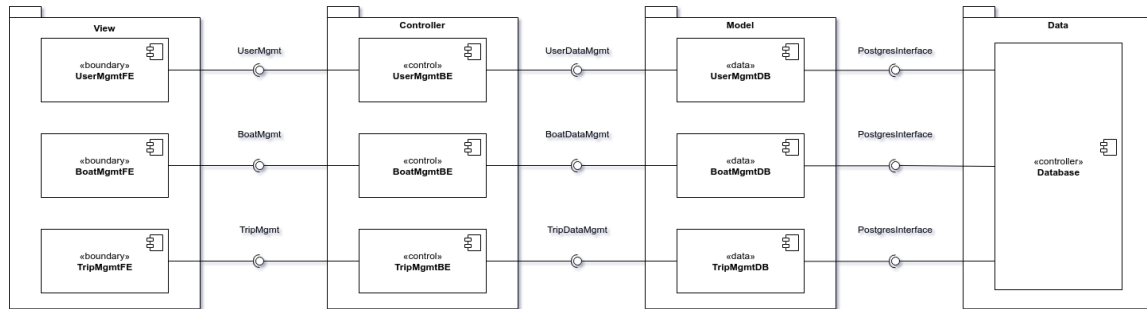


Figura 7: Diagramma dei componenti UML.

3.5 UML Class Diagram per interfacce

Il diagramma in Figura 8 serve a rappresentare le interfacce del sistema e le loro interazioni.

3.6 UML Class Diagram per tipi di dato

Il tipo Booking è stato già esplicitato nell'iterazione precedente nel Class diagram per tipi di dato. Viene riportato di seguito in Figura 9.

3.7 UML Deployment diagram

I componenti descritti precedentemente vengono istanziati nel Deployment diagram. In Figura 10 vengono mostrati i componenti contenuti nei seguenti nodi:

- Cellulare utente è il nodo su cui una persona potrà gestire le prenotazioni.
- Web server fornisce le API richieste dall'applicativo lato front-end.
- Database funge da storage dei dati.

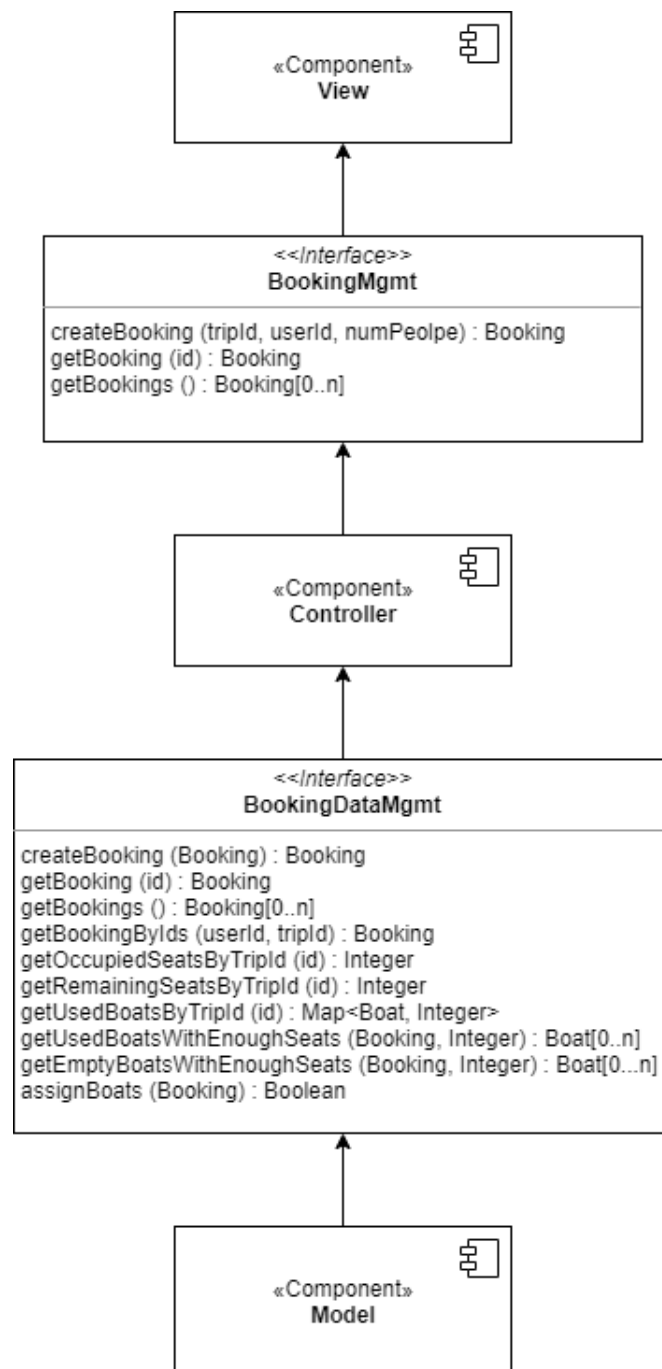


Figura 8: Diagramma delle classi per interfacce UML.

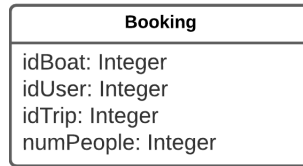


Figura 9: Tipo di dato Booking

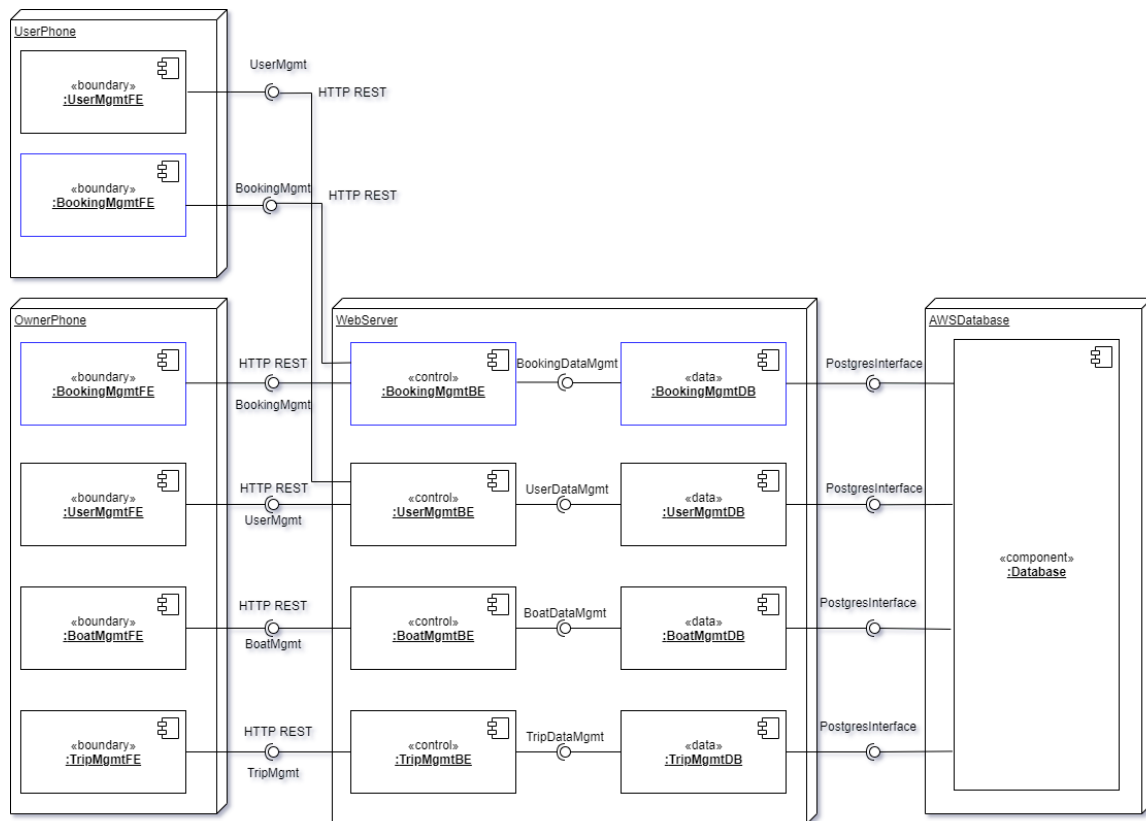


Figura 10: Deployment diagram UML.