

# Capitolo 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 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 1.1.

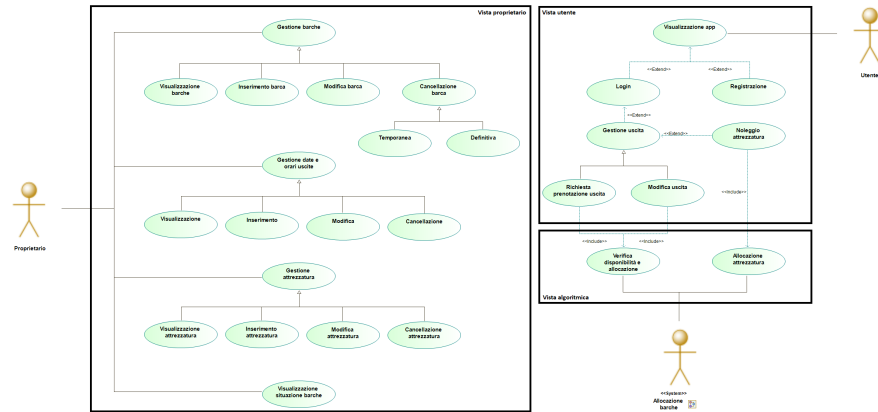


Figura 1.1: Diagramma UML dei casi d'uso

## 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. Il requisito di manutenibilità verrà rispettato mediante l'implementazione di tutte le risorse persistenti ovvero barche, orari e date escursioni, attrezzatura. In questo modo anche se dovessero cambiare nel futuro alcune condizioni sarà facile per il proprietario rifletterle all'interno del software. Il requisito dell'efficienza è l'obiettivo del progetto, ovvero l'allocazione ottimale delle risorse disponibili, i posti presenti nelle barche. 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 1.2) 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 1.2 evidenzia come sia gli utenti che il proprietario accedono alla piattaforma tramite app Android.

Questa mette a disposizione un set di API REST con cui è possibile accedere alle funzionalità dell'applicazione che salverà i dati su un database online.

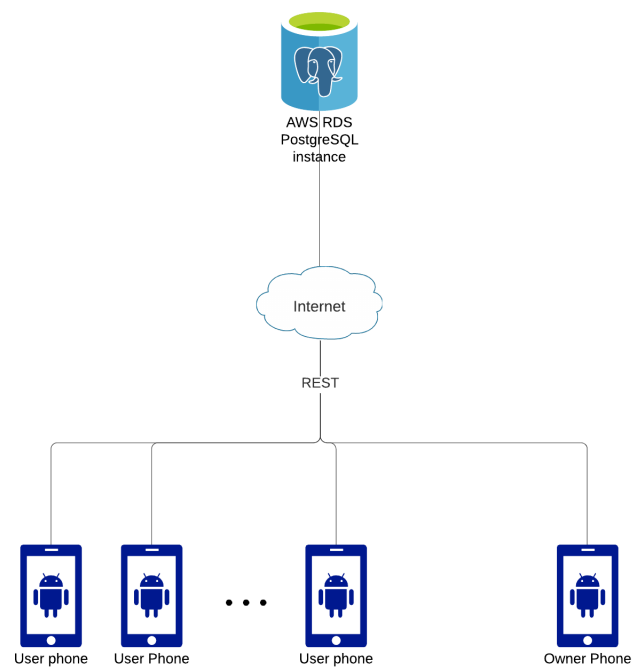


Figura 1.2: Topologia del sistema.



## Capitolo 2

# Iterazione 1

### 2.1 Introduzione

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

- Gestione barche [Proprietario]
- Calendario delle uscite [Proprietario]
- Registrazione [Utente]
- Login [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

Il proprietario del diving center deve avere una visione completa e generale delle sue barche. Accedendo all'applicazione deve poter visualizzare tutte le barche presenti e per ognuna queste caratteristiche:

- Nome della barca
- Modello della barca
- Numero posti disponibili sulla barca
- Altre caratteristiche TODO

Oltre alla visualizzazione, il proprietario deve poter modificare le imbarcazioni già presenti nel sistema. Tali modifiche possono essere dovute a cambiamenti strutturali della barca, per esempio la riduzione del numero di posti disponibili o all'eliminazione di una barca dal sistema. Quest'ultima 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.

## 2.3 UC2: Calendario delle uscite

L'applicazione deve poter semplificare la gestione e l'organizzazione delle uscite al proprietario del diving center.

Da questo punto di vista l'applicazione deve fungere da calendario, mettendo a disposizione le seguenti funzionalità:

- L'inserimento di una data in cui si rende disponibile la prenotazione di uscite, indicando i turni con i relativi orari e durate.
- Visualizzazione di tutte le date e turni messi a disposizione.
- La modifica o la cancellazione delle date e dei turni, in seguito a cambiamenti climatici o altri imprevisti.

## 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 2.1. 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, richiedono a loro volta al database.
- «data» rappresentato dal database in cui verranno memorizzati i dati delle barche, il calendario delle uscite e i dati dell'utente in occasione della registrazione.

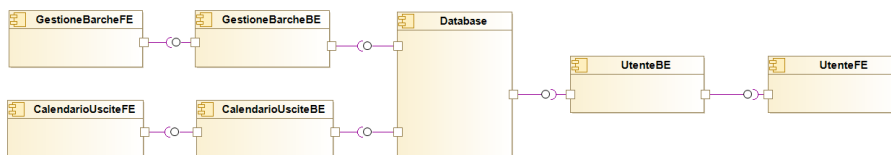


Figura 2.1: Diagramma dei componenti UML.

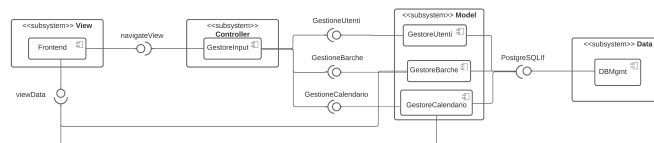


Figura 2.2: Diagramma dei componenti UML.

## 2.7 UML Class diagram per Interfacce

Il seguente diagramma serve a rappresentare le interfacce del sistema e le loro interazioni.

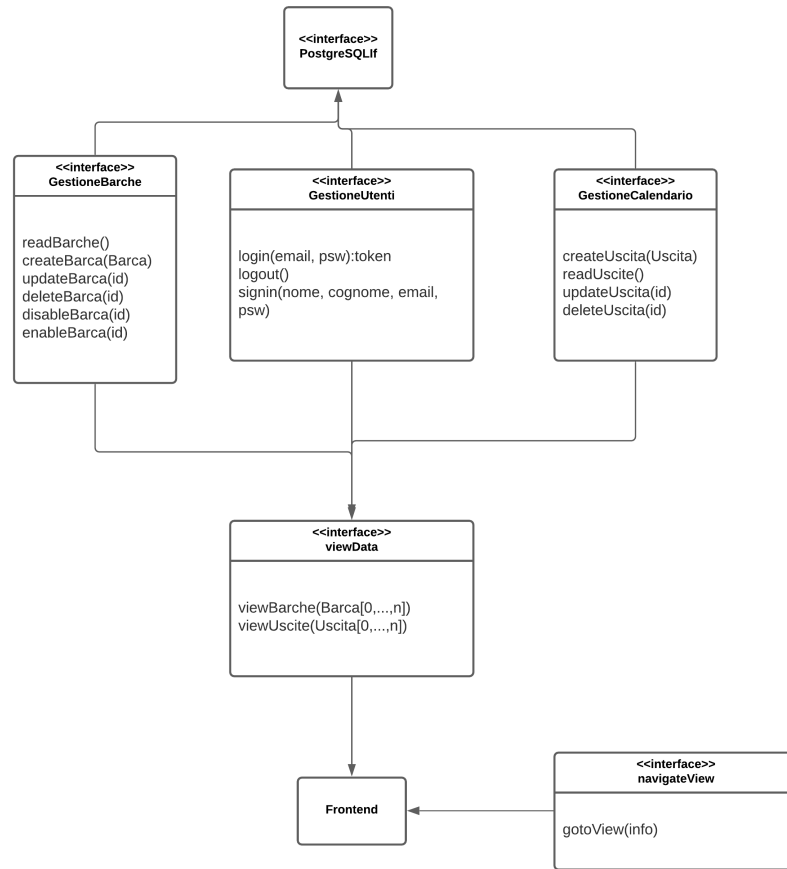


Figura 2.3: Diagramma dei componenti UML.

## 2.8 UML Class diagram per tipi di dato

Il seguente diagramma serve ad esplicitare tipi di dato particolari e i loro legami.

## 2.9 UML Deployment diagram

I componenti descritti precedentemente vengono istanziati nel Deployment diagram. In Figura 2.5 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 uscite.
- Cellulare utente è il nodo su cui una persona potrà registrarsi alla piattaforma.
- Web server fornisce le API richieste dall'applicativo lato front-end.



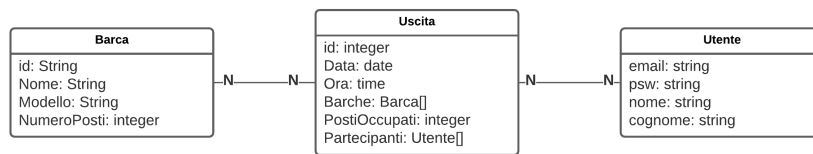


Figura 2.4: Diagramma dei componenti UML.

- Database funge da storage dei dati.

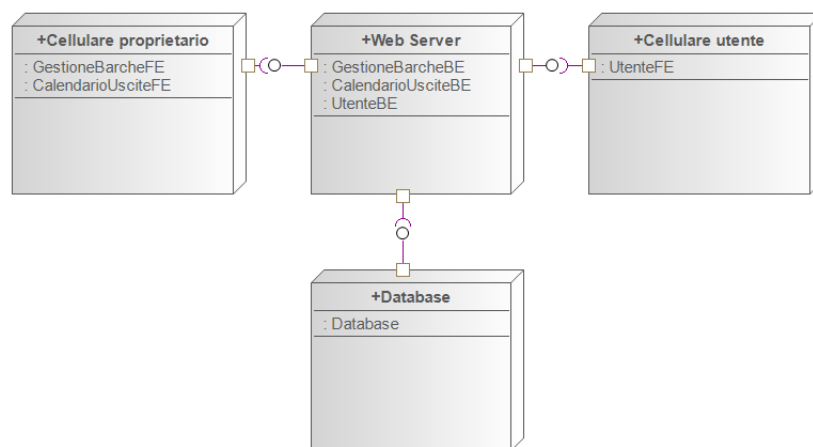


Figura 2.5: Deployment diagram UML.

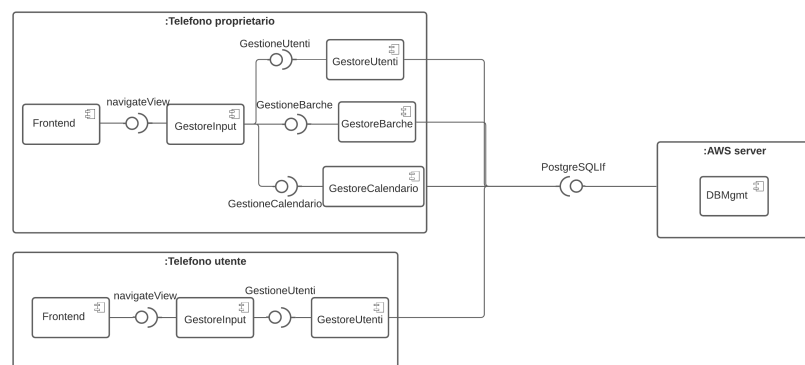


Figura 2.6: Deployment diagram UML.