



Università degli Studi di Salerno
Corso di Ingegneria del Software

**Ship Management
System Design Document
Versione 1.1**



Data: 08/12/2020

Progetto: Ship Management	Versione: 1.1
Documento: System Design	Data: 08/12/2020

Coordinatore del progetto:

Nome	Matricola
Andrea De Lucia	0512100000
Fabiano Pecorelli	//

Partecipanti:

Nome	Matricola
Gennaro Pio Rimoli	0512105894
Andrea Selice	0512105702
Chiara Santoro	0512105786

Scritto da:	Gennaro Pio Rimoli Chiara Santoro Andrea Selice
--------------------	---

Revision History

Data	Versione	Descrizione	Autore
08/12/2020	1.0	Creazione del documento	Andrea Selice
10/12/2020	1.0	Inserimento Introduzione, Architettura del Sistema Esistente, Decomposizione in sottosistemi, Servizi dei sottosistemi	Andrea Selice
12/12/2020	1.0	Revisione Design goal e Architettura del sistema proposto	Gennaro Pio Rimoli Chiara Santoro Andrea Selice
14/12/2020	1.1	Inserimento Gestione dei dati persistenti, Controllo degli accessi e sicurezza	Gennaro Pio Rimoli
15/12/2020	1.1	Inserimento Controllo del sistema globale, revisione Architettura del sistema proposto	Andrea Selice
17/12/2020	1.1	Inserimento Mapping Hardware/Software	Chiara Santoro
19/12/2020	1.1	Revisione documento e correzione di errori	Gennaro Pio Rimoli Andrea Selice
31/12/2020	1.1	Revisione Gestione dei dati persistenti	Gennaro Pio Rimoli



Indice

1. INTRODUZIONE.....	4
1.1 Obiettivi del Sistema	4
1.2 Design Goal.....	4
1.3 Riferimenti	5
1.4 Panoramica	5
2. ARCHITETTURA DEL SISTEMA CORRENTE.....	5
3. ARCHITETTURA DEL SISTEMA PROPOSTO	5
3.1 Panoramica	5
3.2 Decomposizione in sottosistemi.....	6
3.2.1. Uml Wae Diagram	7
3.2.1.1 Uml Wae1.....	7
3.2.1.2 Uml Wae2.....	9
3.2.1.3 Uml Wae3.....	10
3.3 Mapping Hardware/Software	11
3.4 Gestione dei dati persistenti	12
3.5 Controllo degli accessi e sicurezza.....	15
3.6 Controllo del sistema globale.....	16
4. SERVIZI DEI SOTTOSISTEMI.....	16



1- INTRODUZIONE

1.1 OBIETTIVI DEL SISTEMA

Questo sistema nasce dall'esigenza di riunire sotto un unico portale le diverse figure che compaiono durante il commercio marittimo, in modo da ridurre lo sforzo, di tempo e conseguenzialmente quello economico, necessario a eseguire le varie fasi che portano alla registrazione di un contratto di viaggio.

Le figure principali che saranno presenti all'interno del sistema sono tre:

- Il Cliente: colui che ha la necessità di spostare merci da un porto di sorgente ad un porto di destinazione con determinati vincoli
- Il Broker: colui che esercita l'attività di mediazione tra clienti e Armatori allo scopo di definire contratti di trasporto.
- L' Armatore: colui che mette a disposizione, dietro giusto compenso, le proprie imbarcazioni per il trasporto di merci.

L'obiettivo principale del sistema è adempiere all'esigenza di creare un aggregatore di clienti e armatori che può essere utilizzato per le compagnie di broker per rendere veloce e immediata la ricerca dei clienti e/o degli armatori necessari alla stipula di un contratto di viaggio.

1.2 DESIGN GOAL

Criteri di Performance

- Tempo di Risposta:
Gli standard sui tempi di risposta devono essere:
 - 3 secondi per caricare una qualsiasi pagina.
 - 4 secondi per aggiungere/modificare/eliminare un elemento tra cui:
imbarcazione/contratto/utente/mediazione.

Criteri di Affidabilità

- Robustezza:
Il sistema sarà in grado di mostrare alert di errori dovuti ad una non corretta immissione dei dati da parte degli utenti
- Tolleranza ai guasti:
In caso di problematiche lato utente, Ship Management garantirà il salvataggio delle modifiche effettuate in qualsiasi momento.
- Sicurezza:
Il sistema deve prevedere contromisure per evitare attacchi XSS, SQL injection e i dati sensibili quali le credenziali di accesso devono essere codificate in SHA1 in caso di accessi non desiderati ai database di gestione.

Criteri di Manutenzione

- Manutenzione:
Ship Management sarà facilmente modificabile a seguito della distribuzione e vi sarà la possibilità di estenderlo a nuovi scenari.



Criteri Utenti Finali

- Usabilità:

La piattaforma sarà responsive, così da poter garantire la visione su dispositivi con risoluzione diverse. Il sistema sarà costruito utilizzando il concetto del Flat Design Website e, inoltre, avrà un'interfaccia minimal per permettere all'utente di concentrarsi sulle funzionalità del sito. La piattaforma mostrerà i contenuti delle ricerche effettuate dall'utente in tabelle per permettere di trovare le informazioni ricercate in modo più semplice.

1.3 RIFERIMENTI

Problem Statement – Requisiti non funzionali

Requirements Analysis Document – Use Case

1.4 PANORAMICA

Al secondo punto del documento verrà presentata l'architettura del sistema corrente. Al terzo punto sarà presentata l'architettura del sistema proposto, in cui si tratterà:

- Decomposizione del sistema in sottosistemi
- Mapping Hardware/Software
- Gestione dei dati persistenti
- Controllo degli accessi e sicurezza
- Controllo del sistema globale

Al quarto punto saranno presentati i servizi dei vari sottosistemi individuati.

2- ARCHITETTURA DEL SISTEMA CORRENTE

Non essendoci un software già in uso, lo sviluppo del sistema Ship Management rientra nel campo del Greenfield Engineering. Lo sviluppo comincia da zero e i requisiti sono ottenuti dall'utente finale e dal cliente.

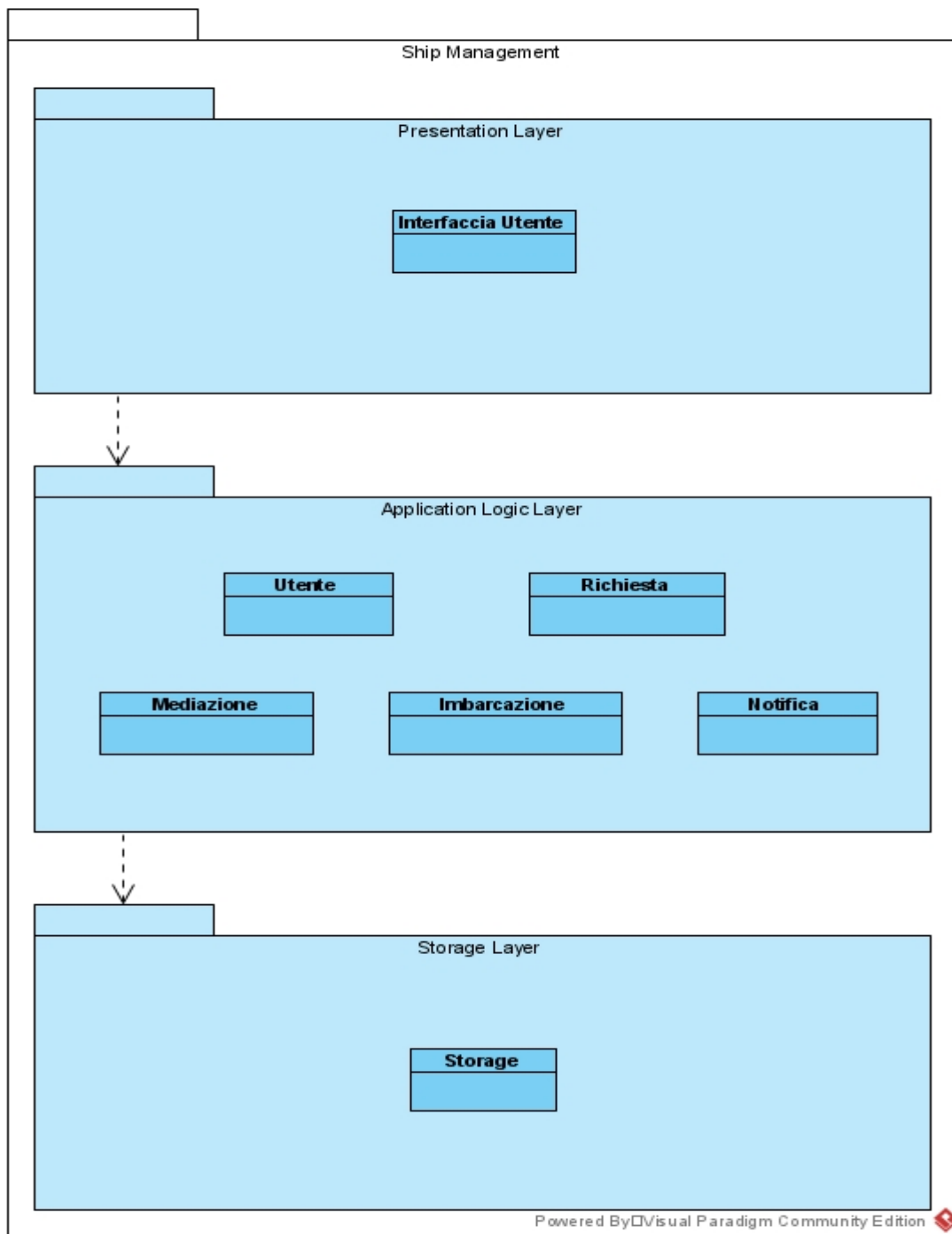
3- ARCHITETTURA DEL SISTEMA PROPOSTO

3.1 PANORAMICA

Il Sistema Ship Management sarà realizzato utilizzando l'architettura three-tier suddivisa in Presentation Layer, Application Logic Layer e Storage Layer che si occupano rispettivamente di:

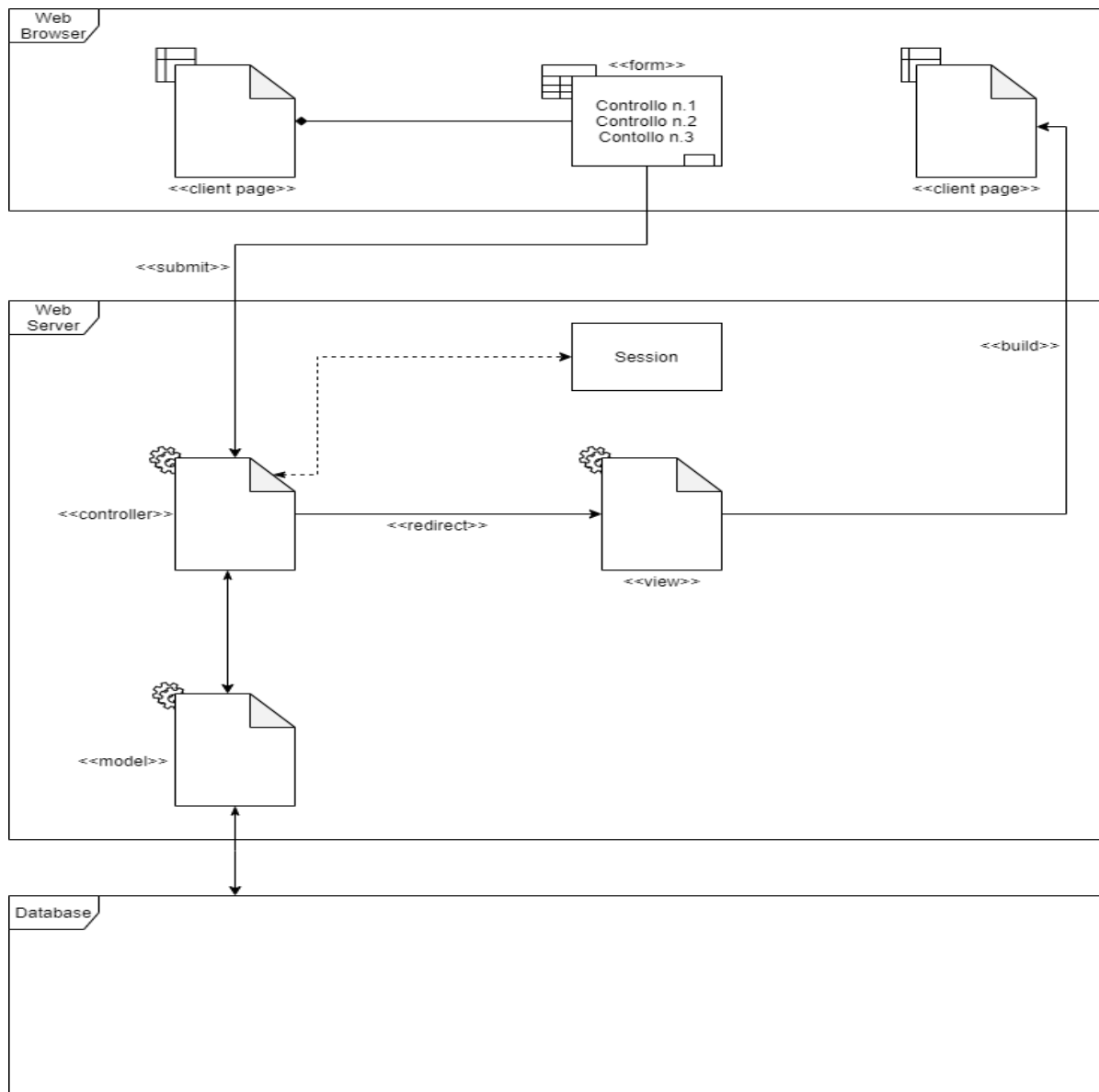
- Presentation Layer: include tutto ciò che riguarda l'interfaccia con l'utente
- Application Logic Layer: gestisce tutte le funzionalità del sistema
- Storage Layer: gestisce la memorizzazione dei dati tramite DBMS

3.2 DECOMPOSIZIONE IN SOTTOSISTEMI



3.2.1 UML WAE DIAGRAM

3.2.1.1 UML WAE1



I casi d'uso che seguono questo UML - WAE1 sono i seguenti:

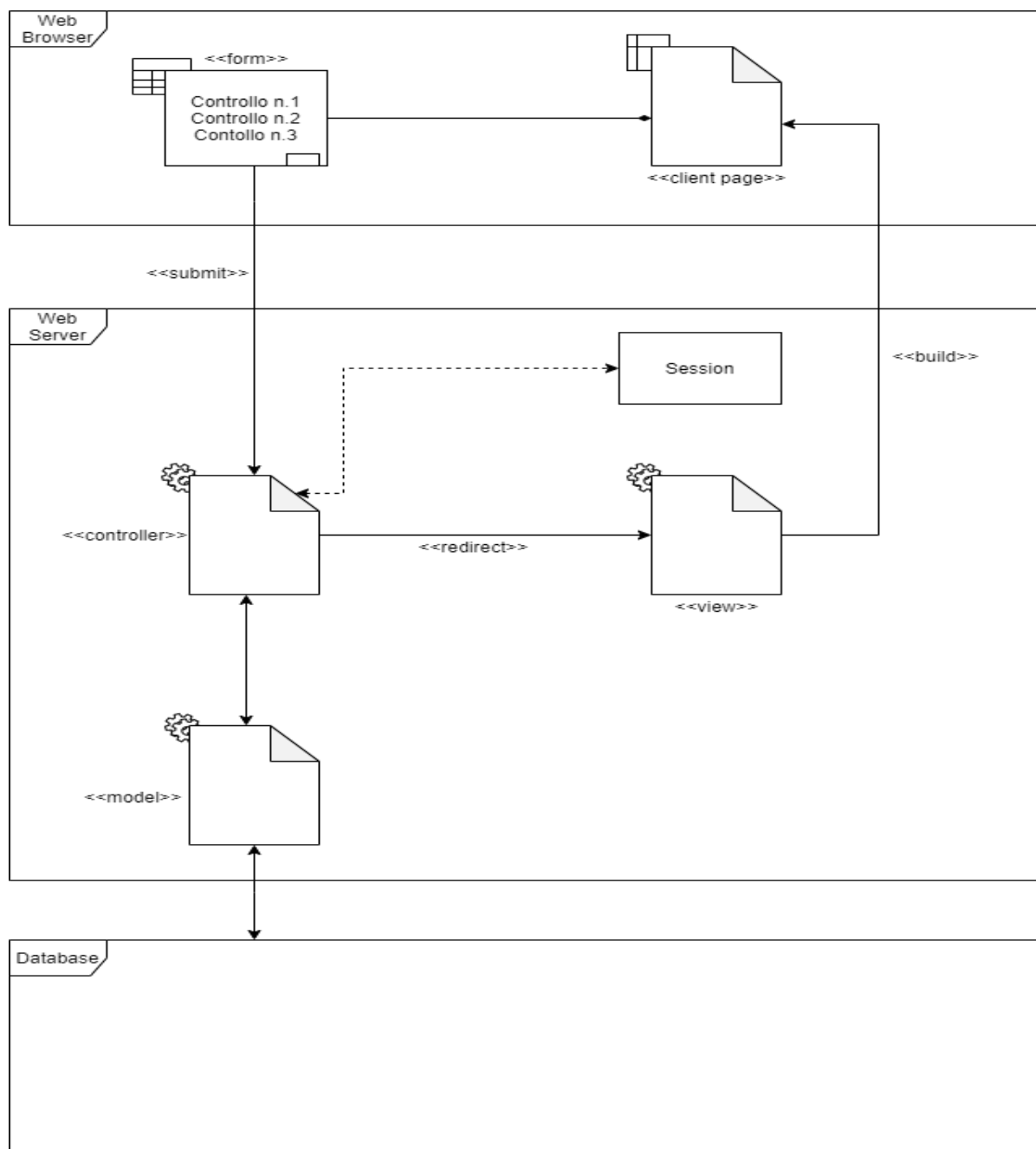
- Registrazione Broker
- Registrazione Cliente
- Registrazione Armatore
- Login
- Logout
- Modifica Profilo Armatore
- Modifica Profilo Cliente
- Modifica Profilo Broker



Laurea Triennale in Informatica-Università degli Studi di Salerno
Corso di Ingegneria del Software - Professore A. De Lucia

- Disattiva Profilo
- Inserimento Imbarcazione
- Modifica Imbarcazione
- Elimina Imbarcazione
- Trasferimento Imbarcazione
- Inserimento Richiesta
- Modifica Richiesta
- Elimina Richiesta
- Crea Mediazione
- Modifica Mediazione
- Elimina Mediazione
- Elimina Notifica

3.2.1.2 UML WAE2

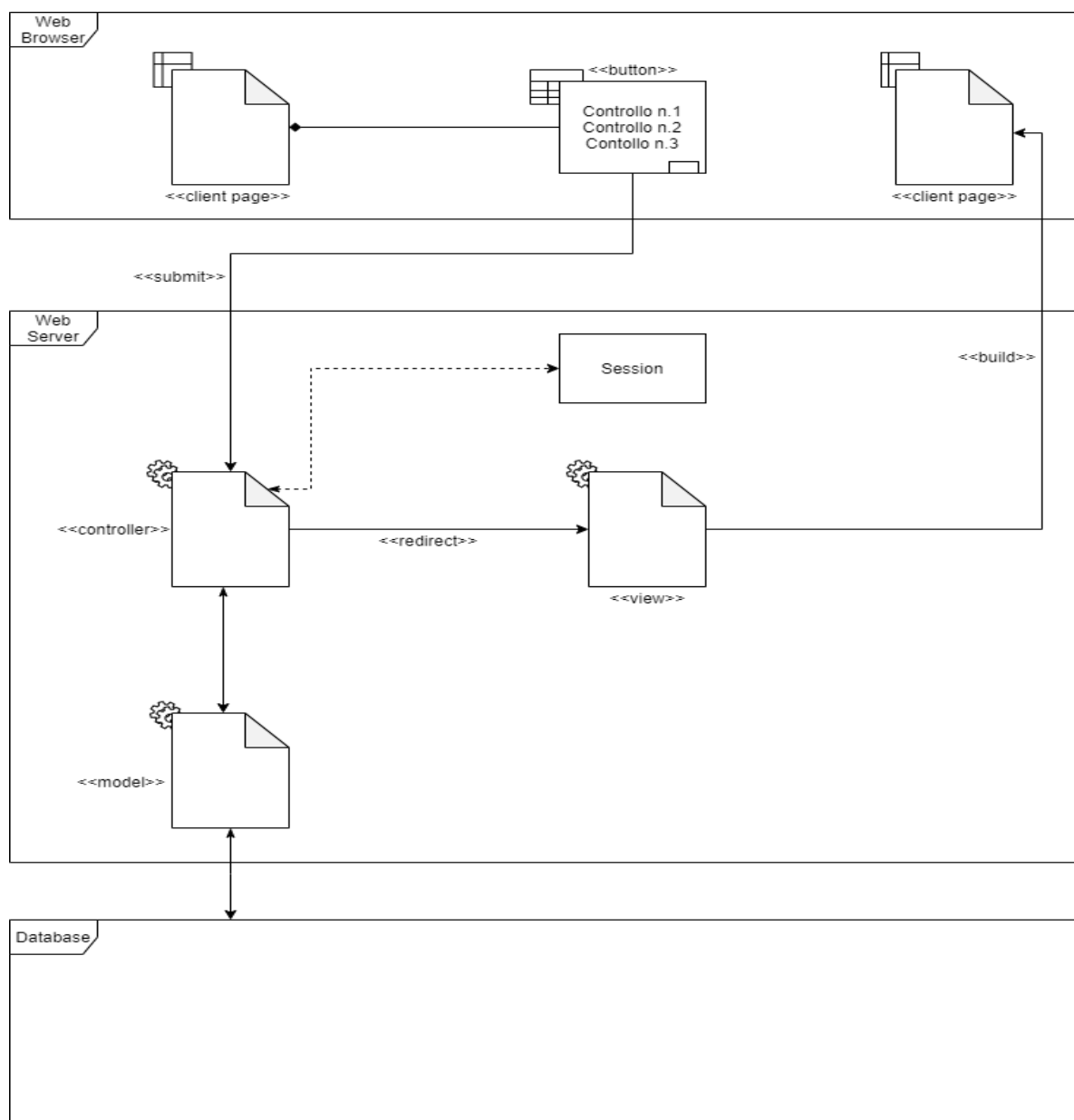


I casi d'uso che seguono questo UML – WAE2 sono i seguenti:

- Ricerca Imbarcazione/Richiesta
- Ricerca Profilo
- Aggiungi Richiesta alla Mediazione
- Aggiungi Imbarcazione alla Mediazione
- Rimuovi Contratto dalla Mediazione
- Elimina Richiesta dalla Mediazione

- Elimina Imbarcazione dalla Mediazione
- Finalizza Mediazione
- Firma Mediazione
- Rifiuta Mediazione
- Accetta Terminazione
- Rifiuta Terminazione

3.2.1.3 UML WAE3



I casi d'uso che seguono questo UML – WAE3 sono i seguenti:

- Visualizza Profilo

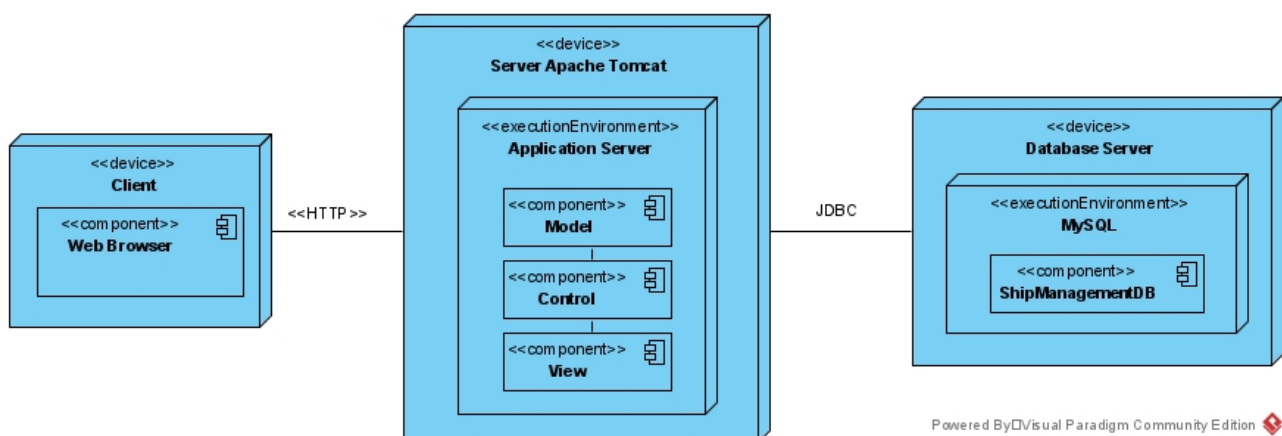
- Visualizza Profilo Broker
- Visualizza Imbarcazione
- Visualizza Imbarcazione Broker
- Visualizza Richiesta
- Visualizza Richiesta Broker
- Visualizza Tabella Ricerca
- Visualizza Profilo Utente
- Visualizza Mediazione Armatore
- Visualizza Mediazione Cliente
- Visualizza Mediazione Broker
- Visualizza Mediazioni Proposte
- Visualizza Storico
- Visualizza Liste Notifiche
- Visualizza Notifica

3.3 MAPPING HARDWARE/SOFTWARE

Il sistema che si intende sviluppare inizialmente prevederà due server:

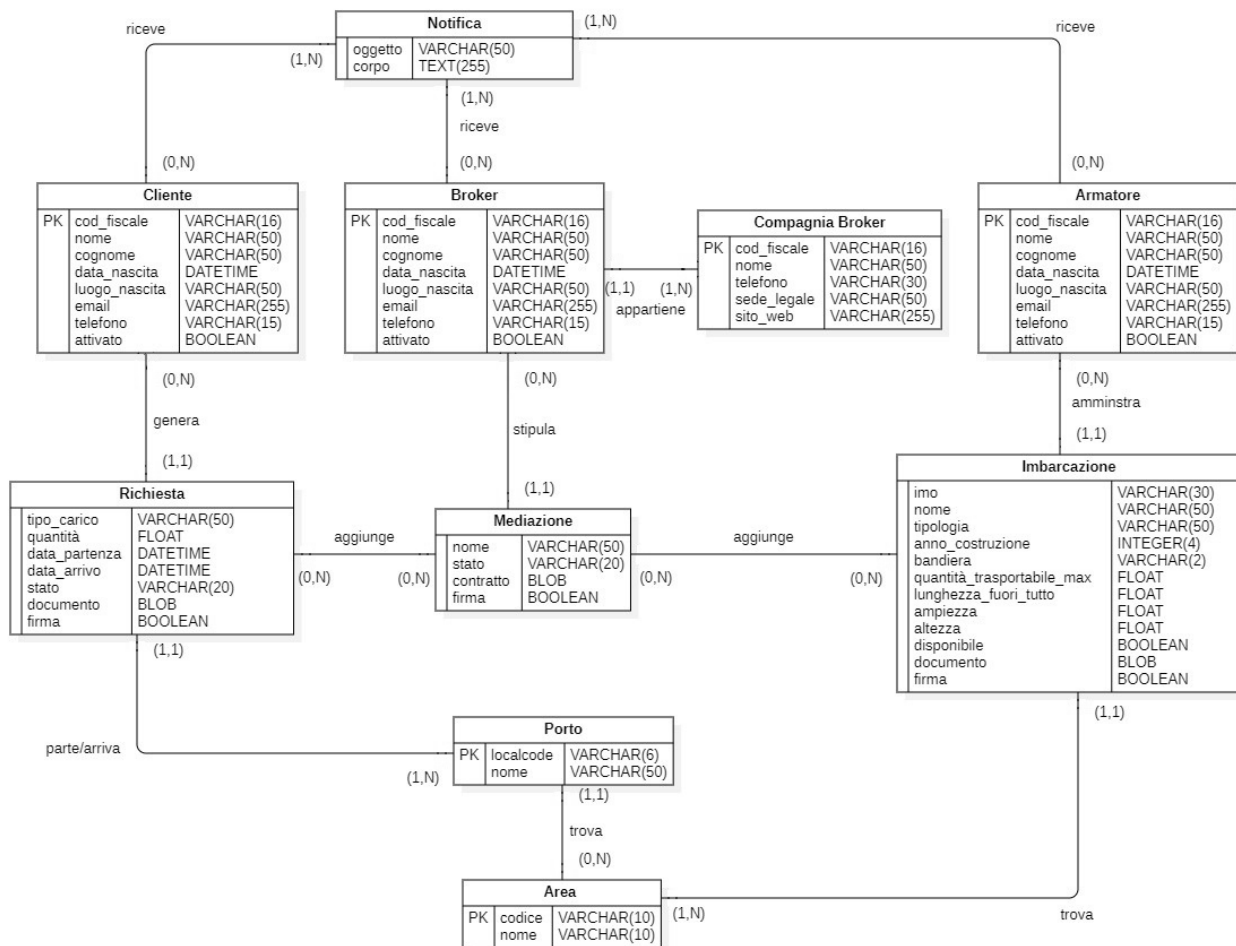
- Sul primo sarà installato un database DBMS Mysql per la gestione dei dati persistenti;
- Sul secondo un web server Apache Tomcat che servirà per la gestione delle Java Servlet, JavaServer Pages, Java Expression Language e Java WebSocket.

Il Client può essere qualunque dispositivo che abbia un client web installato. Il protocollo usato per trasferire i dati tra client e server sarà HTTP, mentre per le comunicazioni tra il server Database e quello Web si utilizzano le API JDBC

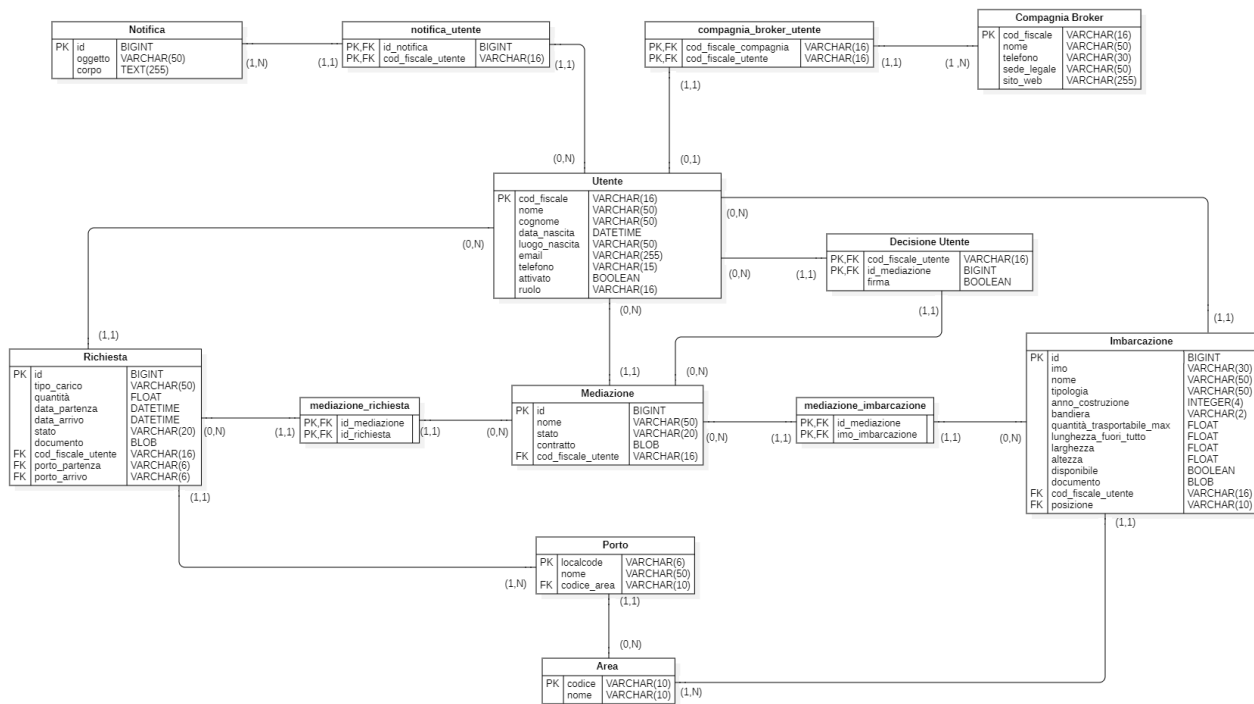


3.4 GESTIONE DEI DATI PERSISTENTI

I dati persistenti saranno memorizzati attraverso il DBMS Mysql seguendo questo diagramma entità relazione (Notazione UML)



Il diagramma entità relazione, qui presente, rappresenta la ristrutturazione del diagramma precedentemente mostrato



utente (cod_fiscale, nome, cognome, data_nascita, luogo_nascita, email, telefono, attivato, ruolo)
porto (localcode, nome, codice_area)

area (codice, nome)

notifica (id, oggetto, corpo)

notifica_utente (id_notifica, cod_fiscale_utente)

compagnia_broker (cod_fiscale, nome, telefono, sede_legale, sito_web)

compagnia_broker_utente (cod_fiscale_compagnia, cod_fiscale_utente)

imbarcazione (id, cod_fiscale_utente, imo, nome, tipologia, anno_costruzione, bandiera, lunghezza_fuori_tutto, ampiezza, altezza, posizione, disponibile, documento)

richiesta (id, cod_fiscale_utente, tipo_carico, data_partenza, porto_partenza, data_arrivo, porto_arrivo, stato, documento)

mediazione (id, nome, stato, contratto, cod_fiscale_utente)

mediazione_richiesta (id_mediazione, id_richiesta)

mediazione_imbarcazione (id_mediazione, imo_imbarcazione)

decisione_utente(cod_fiscale_utente, id_mediazione, firma)

Durante l'analisi del sistema è stato deciso di utilizzare un DBMS relazionale (MYSQL) per le seguenti motivazioni:

- Accesso ai dati tramite un linguaggio universale (SQL).

- Accesso efficiente ai dati. Un DBMS ha molti modi per ottimizzare l'accesso all'informazione. La base di dati è solitamente memorizzata in memoria secondaria. Un DBMS permette di creare dei file ausiliari (indici) che permettono l'accesso veloce ai dati su disco
- Indipendenza dei dati. Un DBMS mantiene diversi livelli di astrazione dei dati, permettendo di accedere ai dati logici indipendentemente dalla loro rappresentazione fisica.
- Controllo della ridondanza dei dati. Ogni dato logico dovrebbe essere memorizzato in un solo posto nella base di dati. Avere più copie della stessa informazione ha i seguenti svantaggi:
 - o Maggior uso di memoria;
 - o Le modifiche della stessa informazione debbono essere effettuate diverse volte;
 - o Ci possono essere fenomeni di inconsistenza dei dati qualora gli aggiornamenti dei dati vengano eseguiti in modo indipendente;
- Atomicità delle operazioni. Un DBMS permette di effettuare sequenze di operazioni in modo atomico. Ciò significa che l'intera sequenza di operazioni viene eseguita con successo oppure nessuna di queste operazioni ha alcun effetto sui dati della base.
- Accesso concorrente ai dati. Un DBMS permette a più utenti di accedere contemporaneamente alla base di dati. Più utenti possono accedere nello stesso istante a dati diversi.
- Privatezza dei dati. Un DBMS permette un accesso protetto ai dati. Utenti diversi possono avere accesso a diverse porzioni della base di dati e possono essere abilitati a diverse operazioni su di esse.
- Affidabilità dei dati. Un DBMS offre dei metodi per salvare copie dei dati (backup) e per ripristinare lo stato della base di dati in caso di guasti software e hardware (recovery).

Per effettuare la ristrutturazione dello schema E-R si è proceduto innanzitutto all'eliminazione della generalizzazione Utente: Broker-Cliente-Armatore. L'alternativa più adatta alla nostra situazione, in quanto generalizzazione totale, è stata di mantenere il genitore ed eliminare i figli della generalizzazione. La scelta è dovuta ai vantaggi che questa soluzione porta in termini di tempi di accesso alle informazioni e migliora la gestione della memoria.

In seguito, si è proceduto all'individuazione delle chiavi primarie per ogni entità e qualora un'entità ne era sprovvista abbiamo creato degli indici auto incrementali che permettono una migliore indicizzazione. Infine, si è provveduto ad effettuare un'analisi delle ridondanze e degli attributi.

Durante questa fase si è individuato l'attributo documento che si è deciso di salvare come dato blob e non come file sul disco rigido per le seguenti motivazioni:

- Sicurezza: Uno dei target principali dell'implementazione di una base di dati è riuscire a mantenere la consistenza dei dati a fronte di qualsiasi problema che sia il crash di sistema o del database, in modo che i dati siano sempre coerenti con la logica dell'applicazione. Cosa succede se il sistema va in crash mentre è in corso l'operazione di inserimento nel database? Potremmo aver caricato il dato all'interno del filesystem ma non il riferimento nel database. Inoltre, se dovessimo caricare i documenti nel filesystem i file stessi potrebbero essere esposti (accessibili al di fuori del sito), mentre caricandoli nel database i dati non possono essere acceduti se non si hanno i permessi necessari.
- Spazio occupato: In tutti i database che supportano i campi blob, l'informazione non è mai immagazzinata all'interno della struttura della tabella, ma sempre al di fuori: i campi a lunghezza limitata vengono allocati nello stesso spazio di allocazione del record, mentre al posto del campo blob viene inserito un puntatore ad una zona di allocazione separata.

- Backup: Effettuando backup centralizzati in un solo punto si diminuiscono i punti di failure, inoltre, effettuando backup separati di filesystem e database possiamo riscontrare incongruenze quando si va ad effettuare il ripristino dei dati perché quest'ultimi non sono effettuati in contemporanea.

3.5 CONTROLLO DEGLI ACCESSI E SICUREZZA

Essendo il sistema formato da diverse tipologie di attori, ognuno con funzionalità differenti, si è organizzato all'interno della tabella di accesso globale, in seguito riportata, le operazioni che ogni utente può usufruire

Attori Oggetti	Utente non Loggato	Cliente	Armatore	Broker
Utente	Registrazione() Login()	Logout() VisualizzaProfilo() ModificaProfilo() EliminaProfilo()	Logout() VisualizzaProfilo() ModificaProfilo() EliminaProfilo()	Logout() VisualizzaProfilo() ModificaProfilo() EliminaProfilo()
Notifica		VisualizzaListaNotifiche() VisualizzaNotifica() EliminaNotifica()	VisualizzaListaNotifiche() VisualizzaNotifica() EliminaNotifica()	VisualizzaListaNotifiche() VisualizzaNotifica() EliminaNotifica()
Mediazione		VisualizzaListaMediazioni() VisualizzaMediazione() FirmaMediazione() AccettaTerminazione() RifiutaMediazione() RifiutaTerminazione()	VisualizzaListaMediazioni() VisualizzaMediazione() FirmaMediazione() AccettaTerminazione() RifiutaMediazione() RifiutaTerminazione()	VisualizzaListaMediazioni() VisualizzaMediazione() CreaMediazione() ModificaMediazione() EliminaMediazione() CaricaContratto() EliminaContratto() FinalizzaMediazione() TerminaMediazione()
Imbarcazione			VisualizzaImbarcazione() CreaImbarcazione() ModificaImbarcazione() TrasferisciImbarcazione() EliminaImbarcazione() RendiIndisponibile() RendiDisponibile()	VisualizzaListaImbarcazione() VisualizzaImbarcazione() AggiungiAllaMediazione() RimuoviDallaMediazione()
Richiesta		VisualizzaRichiesta() CreaRichiesta() ModificaRichiesta() EliminaRichiesta()		VisualizzaListaRichieste() VisualizzaRichiesta() AggiungiAllaMediazione() RimuoviDallaMediazione()

3.6 CONTROLLO DEL SISTEMA GLOBALE

Il sistema è caratterizzato da un sito web accessibile dal browser. Nel momento in cui un utente effettua il Login, vi è un accesso al Database per controllare l'effettiva esistenza di quest'ultimo. Una volta avuta la conferma, l'utente può accedere alle diverse funzionalità messe a disposizione da parte del sistema.

L'utente può richiedere una risorsa attivando una URL. Il browser costruisce una HTTP Request con le informazioni prese dall'URL e viene inviato al server HTTP, in particolare viene catturata da un Controller.

Il Controller riceve l'HTTP Request tradotta in oggetto Java, dopo averlo letto utilizza i dati estratti per eseguire delle operazioni, opzionalmente utilizzando i Model di riferimento dell'oggetto desiderato. Termina con la realizzazione di una risposta inviata sotto forma di Response ad una View, che avrà il compito di leggere i dati e formattarli in pagina Web.

4- SERVIZI DEI SOTTOSISTEMI

- **Utente**

- Registrazione - consente all'utente di potersi registrare in Ship Management
- Login - consente all'utente di effettuare il login
- Logout - consente all'utente di effettuare il logout
- Recupero Password - consente all'utente di poter recuperare la password
- Attivazione Profilo - consente all'utente di poter attivare il profilo per effettuare le operazioni all'interno del sistema
- Ricerca Profilo – consente al broker di poter ricercare i profili presenti nel sistema
- Modifica Profilo - consente all'utente di poter modificare i dati del proprio profilo
- Disattiva Profilo – consente all'utente di poter eliminare il proprio profilo
- Visualizza Profilo – consente all'utente di poter visualizzare i dati del proprio profilo

- **Imbarcazione**

- Inserimento Imbarcazione – consente all'armatore di poter inserire un'imbarcazione nel sistema
- Modifica Imbarcazione – consente all'armatore di poter modificare i dati di un'imbarcazione
- Visualizza Imbarcazione – consente all'armatore di poter visualizzare i dati di un'imbarcazione
- Elimina Imbarcazione – consente all'armatore di poter eliminare un'imbarcazione dal sistema
- Trasferisci Imbarcazione – consente all'armatore di poter trasferire l'imbarcazione ad un nuovo armatore



- **Richiesta**

- Inserimento Richiesta – consente al cliente di poter inserire una richiesta nel sistema
- Modifica Richiesta – consente al cliente di poter modificare i dati di una richiesta
- Elimina Richiesta – consente al cliente di poter eliminare una richiesta
- Visualizza Richiesta – consente al cliente di poter visualizzare i dati di una richiesta

- **Mediazione**

- Crea Mediazione – consente al broker di poter creare una mediazione
- Modifica Mediazione – consente al broker di poter modificare i dati di una mediazione, e il contratto
- Elimina Mediazione – consente al broker di poter eliminare una mediazione
- Ricerca Imbarcazione – consente al broker di poter ricercare le imbarcazioni presenti nel sistema
- Ricerca Richiesta – consente al broker di poter ricercare le richieste presenti nel sistema
- Aggiungere Imbarcazioni/Richieste nella Mediazione – consente al broker di poter aggiungere imbarcazioni, richieste nella mediazione
- Eliminare Imbarcazioni/Richieste/Contratto nella Mediazione – consente al broker di poter eliminare imbarcazioni, richieste o contratto nella mediazione
- Finalizza Mediazione – consente al broker di finalizzare una mediazione dopo aver inserito tutti i dati necessari
- Firmare/Rifiutare Mediazione – consente al cliente e all'armatore di poter firmare o rifiutare una mediazione
- Termina Mediazione – consente al broker di poter proporre di terminare una mediazione
- Accetta/Rifiuta Terminazione Mediazione – consente al cliente e all'armatore di poter accettare o rifiutare la terminazione della mediazione proposta dal broker
- Visualizza Mediazione – consente al broker, al cliente e all'armatore di visualizzare i dati di una mediazione

- **Notifica**

- Visualizza Notifica/che – consente all'utente di poter visualizzare una o più notifiche
- Elimina Notifica – consente all'utente di poter eliminare una notifica

- **Storage**

- Salvare Dati
- Aggiornare Dati
- Eliminare Dati