

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA  
CAMPUS DI CESENA

Dipartimento di Informatica - Scienza e Ingegneria  
Corso di Laurea in Ingegneria e Scienze Informatiche

Elaborato in  
Basi di Dati

**Relatore:**  
**Annalisa Franco**  
**Correlatore:**  
**Annalisa Zaccheroni**

**Presentata da:**  
**Andrea Bedei**

**Anno Accademico 2022/2023**

## **Sommario**

INSERIRE BREVE CONTENUTO DELLA TESI.

RINGRAZIAMENTI(FAMIGLIA, AMICI, PROF, ALTRI) ED EVENTUALE CITAZIONE

# Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
1.1	Diagramma dei casi d'uso . . . . .	5
<b>2</b>	<b>Documentazione Base di dati</b>	<b>6</b>
2.1	Analisi dei requisiti . . . . .	6
2.1.1	Intervista . . . . .	6
2.1.2	Rilevamento delle ambiguità e correzioni proposte	8
2.1.3	Estrazione dei concetti principali . . . . .	10
2.2	Progettazione Concettuale . . . . .	11
2.2.1	Schema scheletro . . . . .	11
2.2.2	Raffinamenti attuati . . . . .	14
2.2.3	Schema concettuale finale . . . . .	14
2.3	Progettazione Logica . . . . .	15
2.3.1	Stima del volume dei dati . . . . .	16
2.3.2	Descrizione delle operazioni principali e stima della loro frequenza . . . . .	16
2.3.3	Schemi di navigazione e tabelle degli accessi . . .	17
2.3.4	Raffinamento dello schema . . . . .	21
2.3.5	Traduzione di entità e associazioni in relazioni . .	21

2.3.6	Schema relazionale finale . . . . .	23
<b>3</b>	<b>Documentazione Applicazione Mobile</b>	<b>24</b>
<b>4</b>	<b>Documentazione Interfaccia Web</b>	<b>25</b>
4.1	Progettazione . . . . .	25
4.1.1	Usabilità . . . . .	26
4.1.2	Design . . . . .	27
4.1.3	Solution stack . . . . .	32
4.1.4	Fruizione del servizio . . . . .	34
4.1.5	Stile . . . . .	36
4.1.6	Javascript e Librerie . . . . .	37
4.1.7	Cookies . . . . .	40
4.1.8	Sicurezza . . . . .	41
4.2	Accessibilità . . . . .	43
4.2.1	WCAG . . . . .	44
4.3	Struttura delle pagine . . . . .	47
<b>5</b>	<b>Documentazione Algoritmo Di Riconoscimento</b>	<b>48</b>
<b>6</b>	<b>Documentazione Test</b>	<b>49</b>
6.1	Test con l'utente . . . . .	49
<b>7</b>	<b>Conclusioni</b>	<b>50</b>
<b>8</b>	<b>Utilità</b>	<b>51</b>
8.1	Ingegneria del software . . . . .	51
8.1.1	Verifica del software . . . . .	51
8.1.2	Analisi . . . . .	53

# Capitolo 1

## Introduzione

Durante il percorso accademico mi è stata proposta la creazione di una piattaforma per la gestione degli avvistamenti di particolari specie marine. Tale proposta è stata presa in considerazione al fine di aiutare il dipartimento di acquacoltura e igiene delle produzioni ittiche per il tracciamento degli avvistamenti delle diverse specie marine.

Ho intrapreso questo progetto al fine di mettere alla prova le conoscenze apprese durante il corso universitario con lo scopo di creare un'intera infrastruttura centralizzata.

Il sistema si suddivide in diverse parti, tra cui:

- Creazione di un database centralizzato che permetta di salvare tutte le informazioni ricavate da un avvistamento e di poterle modificare in un secondo momento.
- Creazione di applicazione smartphone che permetta di inserire i nuovi avvistamenti.

- Creazione di una applicazione web che permetta di gestire, modificare e visualizzare gli avvistamenti e i relativi dati.
- Utilizzo di script di riconoscimento che attraverso la visione artificiale e diversi algoritmi noti permetta di individuare la specie precisa di determinati esemplari acquatici.

## 1.1 Diagramma dei casi d'uso

E' stato riportato un diagramma dei casi d'uso il quale descrive le principali operazioni del sistema:

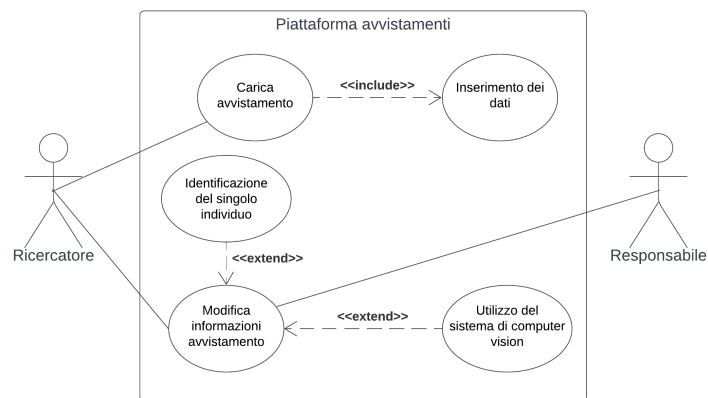


Figura 1.1: Schema dei casi d'uso per la piattaforma.

# Capitolo 2

## Documentazione Base di dati

### 2.1 Analisi dei requisiti

In questa sezione analizzeremo tutte le specifiche che il committente richiede per la piattaforma.

#### 2.1.1 Intervista

Di seguito si riporta la prima intervista col committente: "La facoltà di Acquacoltura e Igiene delle produzioni ittiche richiede un sistema informativo per la gestione di una banca dati al fine di permettere la registrazione di determinati avvistamenti durante le uscite a largo dalla costa.

Ogni utente, il quale ha a disposizione uno smartphone, una volta effettuato l'accesso con login e password, ha la possibilità di inserire un nuovo avvistamento, per il quale gli elementi da memorizzare sono la data e ora corrente(inserite in autonomia dal sistema), il numero di esemplari che sono stati avvistati e opzionalmente anche il vento in km/h, le condizioni del mare, eventuali ferite visibili. In particolare si ponga attenzione alla descri-



zione della ferita, la sua posizione e la relativa gravità. Deve anche essere possibile scrivere note che possano aiutare ad individuare la specie e/o elementi che aggiungano informazioni all'avvistamento.

Un avvistamento può far riferimento ad un particolare animale, a una specie specifica o anche a nessuna delle due: il terzo caso viene preso in considerazione quando chi sta eseguendo la rilevazione non ha la possibilità di riconoscere la specie esatta, inseribile comunque in un secondo momento.

Oltre all'applicazione smartphone deve essere realizzato un applicativo web il quale, una volta effettuato l'accesso, deve poter visualizzare tutti gli avvistamenti che sono stati effettuati dagli utenti e poter modificare i parametri. La modifica deve essere permessa per tutti i dati, eccetto gli elementi che riguardano l'utente e la data della rilevazione.

Ad ogni avvistamento il personale addetto può associare anche una o più immagini. Esse serviranno in seguito per il riconoscimento della specie di appartenenza. Dato che in un avvistamento possono essere presenti anche più animali, comunque tutti della stessa specie, la foto può essere suddivisa in più parti in modo tale da isolare ogni singolo individuo.

La base di dati deve anche tenere in considerazione la possibilità di avvistamenti dello stesso individuo in momenti diversi. Nello specifico, attraverso le sottoimmagini se si avvista più volte lo stesso esemplare si deve avere la possibilità di associargli entrambe le immagini riferite a lui, in modo tale da vedere la sua evoluzione nel tempo.

Per aiutare l'utente nella decisione della specie da assegnare a un particolare animale, il sistema deve memorizzare una descrizione, in cui vengono fornite le informazioni base su ciascuna specie analizzabile.

Infine l'applicativo web, attraverso l'uso di visione artificiale ed in particolare per l'animale denominato delfino, deve poter essere in grado in autonomia di identificare la relativa specie attraverso la foto scattata durante

il rilevamento. Nello specifico deve cercare di aiutare l'utente nella scelta della specie, stilando una classifica di compatibilità con le specie che ha presenti in database.

Un aspetto importante da tenere in considerazione è il fatto che il sistema in autonomia, siccome se ci si trova a largo della costa non è presente connessione, deve poter lo stesso mantenere i dati in memoria e caricali una volta che ha la possibilità di farlo”.

### **2.1.2 Rilevamento delle ambiguità e correzioni proposte**

Il testo dell'intervista presenta molte ambiguità. Le principali sono:

- Utilizzo di sinonimi.
- Elenchi di attributi incompleti.
- Cardinalità non specificate.

Per quanto riguarda gli attributi parziali e le cardinalità, questi aspetti saranno corretti mediante l'uso della logica in fase di creazione dello schema concettuale. Invece per quanto concerne i sinonimi, è necessario costruire un glossario dei termini, i quali saranno considerati al fine della progettazione concettuale:

Termine	Descrizione	Sinonimi	Collegamenti
Utente	Persona che accede al portale al fine di inserire un nuovo avvistamento oppure per modificare avvistamenti già presenti nella banca dati	Persona	Avvistamento
Avvistamento	Avvistamento di un esemplare, al fine di memorizzare informazioni	Rilevazione	Utente Animale Specie Immagine
Animale	Tipologia di animale avvistato	Specie marina	Avvistamento Specie
Specie	Specie dell'animale avvistato		Animale Avvistamento Descrizione
Descrizione	Descrizione specifica al fine di una migliore selezione della specie		Specie
Immagine	Immagine che rappresenta l'avvistamento	Foto	Avvistamento Sottoimmagine
Sottoimmagine	Sezioni di immagini dei singoli individui		Immagine Ferita Esemplare
Ferita	Ferita sul corpo di un animale, con una gravità		Sottoimmagine
Esemplare	Singolo animale che si monitora		Sottoimmagine

### **2.1.3 Estrazione dei concetti principali**

Dall'intervista si ricavano anche le operazioni principali richieste:

- Creazione di un nuovo utente.
- Registrazione di un nuovo avvistamento attraverso l'applicazione mobile.
- Inserimento di ferite riconducibili al singolo animale con specifica della gravità.
- Possibilità di inserimento di una o più immagini dell'avvistamento.
- Possibilità di visualizzare, modificare ed eliminare un avvistamento.
- Possibilità di suddividere immagini in più parti, e ciascuna delle quali a seconda dell'animale analizzarla attraverso un sistema di riconoscimento della specie.
- Monitorare gli avvistamenti di particolari individui al fine di determinate verifiche.
- Possibilità di consultazione di informazioni relative alla specie che si vuole attribuire a un determinato animale, al fine di una selezione più accurata.
- Caricamento del nuovo avvistamento, compreso di dati e immagini anche in un secondo momento, nel caso non sia presente connessione.

## 2.2 Progettazione Concettuale

La fase di progettazione concettuale consiste parzialmente nella formalizzazione dei requisiti, in particolare sui termini e sulle transazioni, i quali sono stati raccolti e analizzati nella fase precedente.

### 2.2.1 Schema scheletro

Per maggiore chiarezza, di seguito riportiamo i primi esempi di schemi concettuali divisi per ambiti.

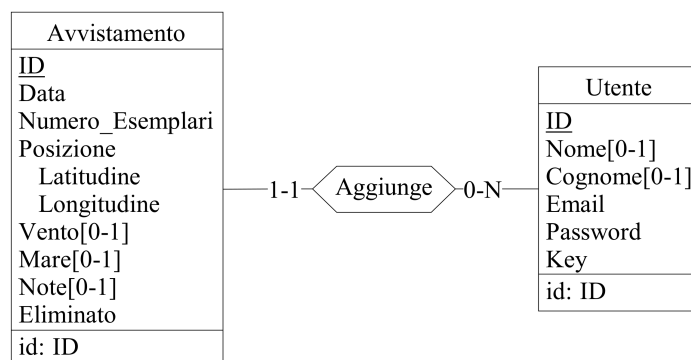


Figura 2.1: Schema concettuale per la modellazione degli avvistamenti e delle proprietà dell'utente.

Come mostrato in figura sono presenti uno o più utenti, ognuno dei quali accede alla piattaforma attraverso la propria email, password e grazie a una chiave personale univoca all'interno del database, la quale viene utilizzata al fine di criptare la password. Di tale utente possiamo tenere anche in considerazione il proprio nome e cognome. Ogni utente aggiunge degli avvistamenti, ognuno dei quali contiene tutti i dati richiesti del committente

tra cui la data di avvistamento, il numero di esemplari presenti e opzionalmente anche il vento espresso in km/h, lo stato del mare ed eventuali note informative al fine di ricerca e in particolare per riuscire ad identificare meglio la specie.

In aggiunta per ogni avvistamento è associata una posizione la quale tiene in considerazione la longitudine e la latitudine al momento dell'inserimento.

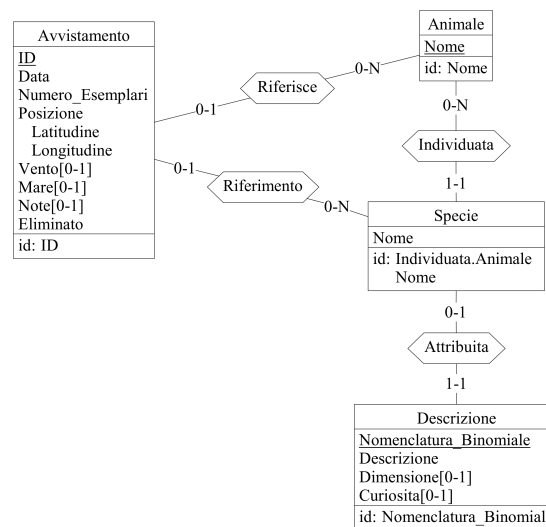


Figura 2.2: Modellazione dell'attribuzione dell'animale e della specie all'avvistamento.

Dalla figura si può notare che un avvistamento può far riferimento a un animale, direttamente alla specie o anche a nessuno dei due, in quest'ultimo caso tali dati verranno inseriti in un secondo momento. Al fine di individuare al meglio la specie ad ognuna può essere attribuita una descrizione in cui vengono elencate le caratteristiche fondamentali. Ogni specie è identificata dal nome dell'animale e dal suo nome specifico. Come si nota dalla

figura, l'avvistamento può essere direttamente associato all'animale oppure alla specie, in questo caso nasce il vincolo che se dichiaro una determinata specie essa deve fare riferimento al giusto animale. Un ulteriore vincolo da rispettare è il fatto che in un avvistamento, anche se esso include più esemplari, questi devono far riferimento allo stesso animale ed eventualmente alla relativa specie.

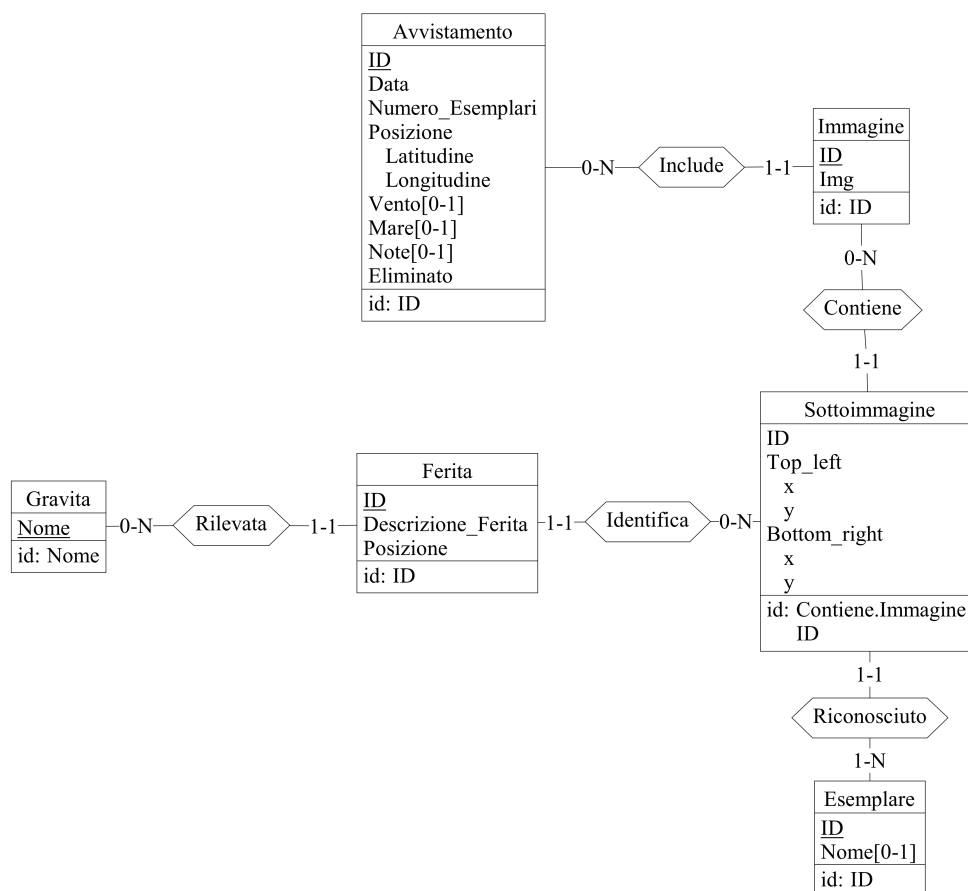


Figura 2.3: Schema concettuale raffigurante le immagini con relative sottoimmagini e la gestione delle ferite.

Navigando lo schema partendo dall'entità Avvistamento, si nota che per ogni avvistamento si possono includere una o più immagini le quali possono essere suddivise in ulteriori sottoimmagini, per fare questo si è pensato di utilizzare le coordinate cartesiane al fine di risparmiare memoria e tempo di elaborazione. Visto che ogni sottoimmagine rappresenta un elemento singolo, se esso presenta delle ferite queste possono essere inserite e ad ognuna attribuita una gravità. In aggiunta il singolo individuo viene catalogato, con un id e opzionalmente anche con un nome, in modo tale che se viene riscontrato in altri avvistamenti si abbia la possibilità di indicare che si tratta dello stesso esemplare.

### **2.2.2 Raffinamenti attuati**

Elenchiamo in questa sezione i raffinamenti che sono stati attuati nei precedenti schemi al fine di modellare meglio entità indipendenti. In particolare:

- L'entità Descrizione è stata creata al fine di non appesantire troppo l'entità Specie con attributi opzionali, i quali come si vede dallo schema rappresentano alla fine una sezione a parte del dominio.

### **2.2.3 Schema concettuale finale**

Di seguito si riporta lo schema concettuale finale, ottenuto unendo opportunamente gli schemi scheletro presentati in precedenza.



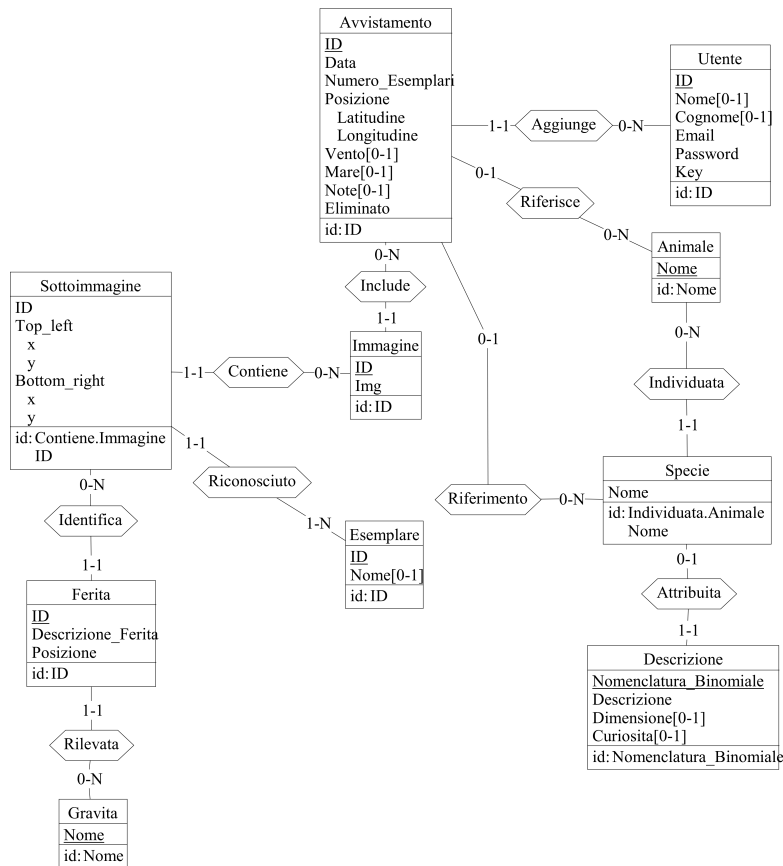


Figura 2.4: Schema concettuale finale

## 2.3 Progettazione Logica

La progettazione logica consiste nella traduzione dello schema concettuale finale in uno schema logico che rispecchi il modello relazionale. Lo schema logico progettato è indipendente dallo specifico Database manage-

ment system, che verrà scelto al termine della progettazione logica. Inoltre verranno definiti anche i vincoli di integrità sui dati.

### 2.3.1 Stima del volume dei dati

Nella tabella di seguito è riportato il volume atteso per ciascun costrutto presente nello schema concettuale. Inoltre, per garantire maggiore compattezza, sono state omesse le stime dei volumi delle associazioni 1-N, in quanto equivalenti ai volumi delle entità che partecipano alle associazioni stesse con cardinalità 1.

Concetto	Costrutto	Volume
Avvistamento	E	100
Utente	E	10
Immagine	E	200
Sottoimmagine	E	400
Ferita	E	50
Gravita	E	3
Esemplare	E	300
Animale	E	z
Specie	E	x
Descrizione	E	x-y

### 2.3.2 Descrizione delle operazioni principali e stima della loro frequenza

Di seguito si riporta una tabella contenente la frequenza prevista e una descrizione delle principali operazioni, individuate già in fase di analisi.

Codice Operazione	Descrizione Operazione	Frequenza
1	Creazione di un nuovo utente	1/settimana
2	Monitoraggio di specifici individui	5/giorno
3	Registrazione di un nuovo avvistamento compreso di operazioni opzionali	10/giorno

### 2.3.3 Schemi di navigazione e tabelle degli accessi

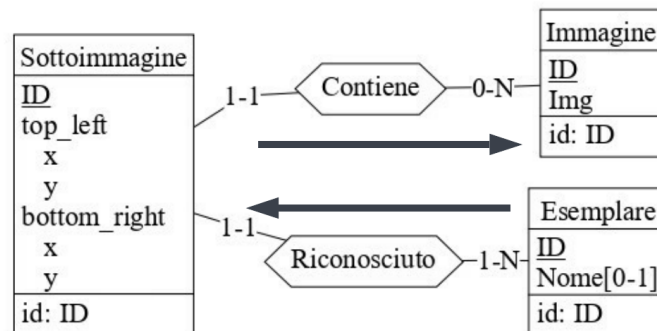
Dopo aver stimato i volumi dei principali costrutti presenti nella base di dati e la frequenza delle principali operazioni, si può procedere a disegnare i relativi schemi di navigazione e scrivere le tabelle degli accessi.

#### 1. Creazione di un nuovo utente:

Per quanto riguarda l'aggiunta dell'utente, omettiamo il corrispondente schema di navigazione in quanto coincide con l'entità stessa:

Concetto	Costrutto	Accessi	Tipo
Utente	E	1	S
Totale = 1S			

2. Monitoraggio di specifici individui al fine di determinate verifiche:



Per svolgere questa operazione occorre effettuare un totale di letture su Sottoimmagine pari al numero medio di foto a cui ogni esemplare è associato. In aggiunta, in base al numero di sottoimmagini a cui l'esemplare è associato si dovranno selezionare lo stesso numero di immagini, visto che uno stesso esemplare non può comparire più volte nella stessa immagine.

Concetto	Costrutto	Accessi	Tipo
Esemplare	E	1	L
Sottoimmagine	E	$400 \div 200 = 2$	L
Immagine	E	2	L
Totale = 5L			

3. Registrazione di un nuovo avvistamento compreso di operazioni opzionali:

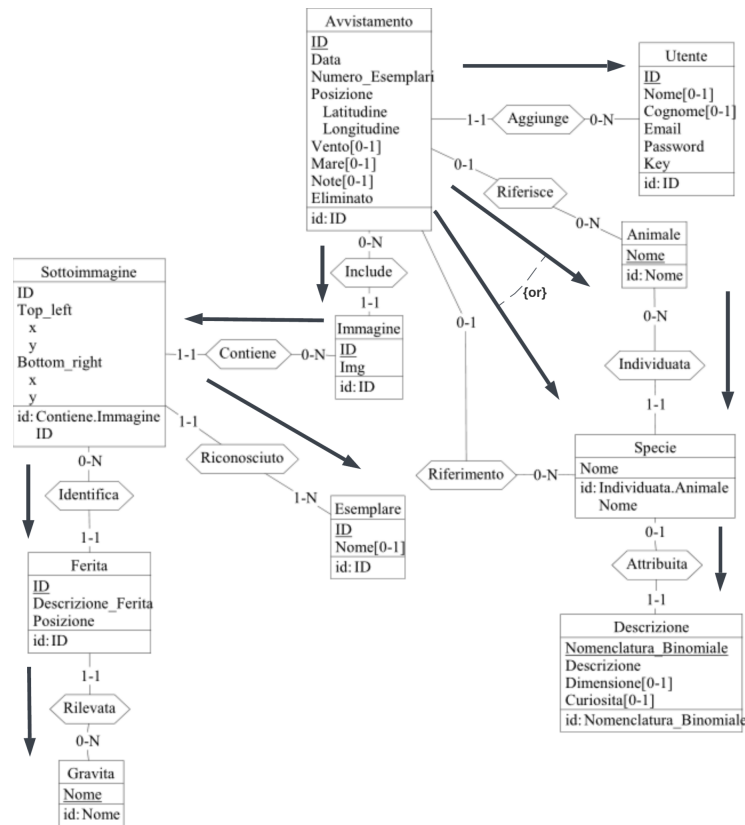


Figura 2.5: Schema concettuale raffigurante gli accessi alle tabelle per l'aggiunta di un nuovo avvistamento.

Lo schema di navigazione prevede di partire da **Avvistamento**, il quale legge l'utente, poi aggiunge più immagini le quali verranno suddivise in più sottoimmagini, ad ognuna delle quali verrà associato un esemplare. In aggiunta, ad ogni sottoimmagine si possono aggiungere delle possibili ferite, visibili in quella parte dell'immagine, ad ogni

ferita è obbligatorio associargli una gravità. Nella tabella degli accessi sottostante consideriamo il caso in cui l'esemplare e la coordinata geografica non siano ancora presenti nel database. Ogni avvistamento è associato ad un'animale e di conseguenza a una possibile specie. Al fine di individuare meglio la specie, essa è associata a una possibile descrizione.

Un'alternativa è la possibilità di scegliere direttamente la specie partendo dall'avvistamento, ma questo viene considerato un caso raro.

Concetto	Costrutto	Accessi	Tipo
Avvistamento	E	1	S
Utente	E	1	L
Animale	E	1	L
Specie	E	1	L
Descrizione	E	1	L
Immagine	E	$200 \div 100 = 2$	S
Sottoimmagine	E	$(400 \div 200) \times 2 = 4$	S
Ferita	E	$(50 \div 400) \times 4 = 0.5$	S
Gravita	E	1	L
Esemplare	E	4	S
Totale = $11.5S + 5L$			

### Accessi totali

Si riporta di seguito la tabella degli accessi totali per ogni operazione, considerando doppi gli accessi in scrittura:

Codice Operazione	Accessi	Frequenza	Totale
1	$1S = 2$	1/settimana	2/settimana
2	$5L = 5$	5/giorno	25/giorno
3	$11.5S + 5L = 28$	10/giorno	280/giorno

### 2.3.4 Raffinamento dello schema

In questa sezione ci occuperemo del raffinamento dello schema concettuale in vista della realizzazione dello schema logico. In particolare, bisognerà concentrarsi sulla seguente operazione.

- **Modifica degli attributi multipli e composti:**

Una volta individuate le entità che contengono attributi multipli o composti, si provvede a modificarli. In particolare:

- Nell'entità Sottoimmagine gli attributi top\_left e bottom\_right formati da x e y vengono trasformati in singoli attributi.
- Nell'entità Avvistamento l'attributo posizione viene scomposto in due singole attributi denominati Latitudine e Longitudine.

### 2.3.5 Traduzione di entità e associazioni in relazioni

La traduzione delle entità in relazioni è automatica e non richiede particolari passaggi.

Per quanto riguarda invece le associazioni, bisogna effettuare diversi passaggi in base alla cardinalità:

- Le associazioni 1-N vengono tradotte importando nell'entità che partecipa con cardinalità 1 la chiave dell'entità che partecipa con cardinalità N.
- Le associazioni 1-1 vengono tradotte importando nell'entità ritenuta più corretta, in base ad opzionalità e operatività, la chiave dell'altra entità.

L'operazione di traduzione porta alla creazione dello schema logico:

**Gravita**(Nome)

**Ferite**(ID, Descrizione\_Ferita, Posizione, Gravi\_Nome:Gravita,  
Sottoi\_ID:Sottoimmagini, Img\_rif:Immagini)

**Sottoimmagini**(ID, tl\_x, tl\_y, br\_x, br\_y, Immag\_ID:Immagini,  
Esemp\_ID:Esemplari)

**Esemplari**(ID, Nome\*)

**Immagini**(ID, Img, Avvis\_ID:Avvistamenti)

**Utenti**(ID, Nome\*, Cognome\*, Email, Password, Key)

**Animali**(Nome)

**Specie**((Nome, Anima\_Nome:Animali),  
Nomenclatura\_Binomiale\*:Descrizioni)

**Descrizioni**(Nomenclatura\_Binomiale, Descrizione, Dimensione\*,  
Curiosita\*)

**Avvistamenti**(ID, Data, Numero\_Esemplari, Vento\*, Mare\*, Note\*,  
Latid, Long, Utente\_ID:Utenti, Anima\_Nome\*:Animali,  
(Specie\_anima\_Nome, Specie\_Nome)\*:Specie, Eliminato)



### 2.3.6 Schema relazionale finale

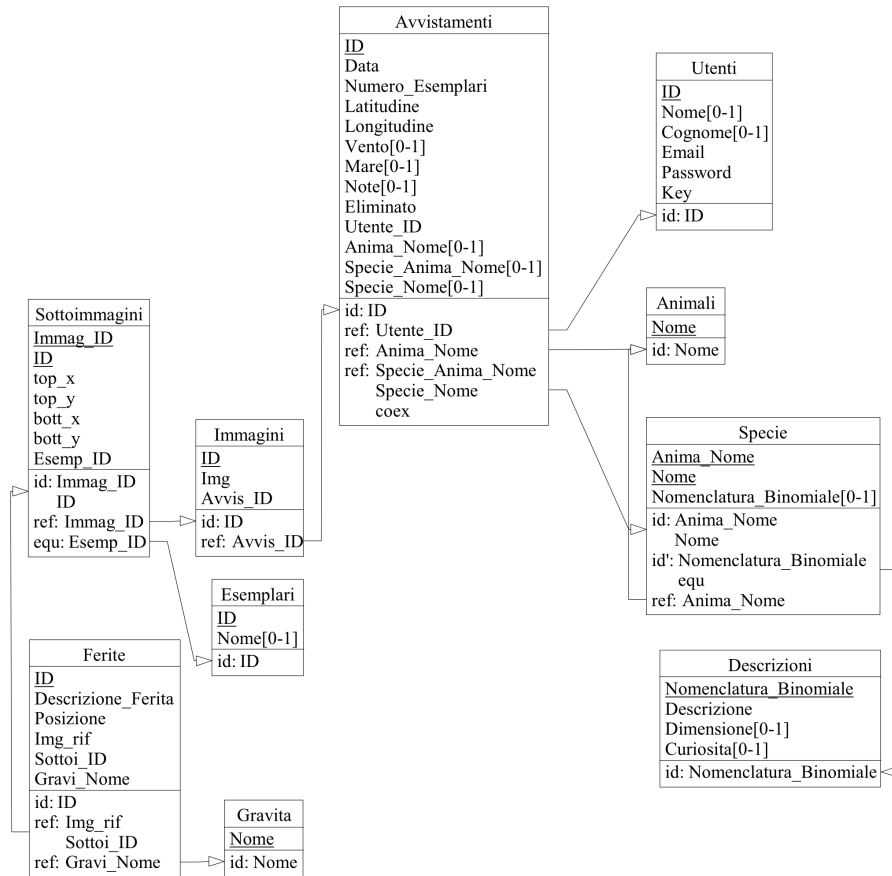


Figura 2.6: Schema logico finale.

## **Capitolo 3**

# **Documentazione Applicazione Mobile**

# Capitolo 4

## Documentazione Interfaccia Web

### 4.1 Progettazione

Nella progettazione sia dell'applicativo web che mobile si è cercato di dare particolare attenzione alla comunicazione visiva della GUI, nello specifico si è cercato di rispettare i seguenti principi:

- **Affordance:** enfatizzare aspetti di un oggetto che invitano a manipolarlo in un certo modo.  
Tali aspetti possono essere: Tridimensionalità, ombreggiatura e puntamento.
- **Metafora:** una parola, una frase o una figura la quale dipinge un oggetto o un concetto attraverso una somiglianza con un altro oggetto del mondo reale. Un esempio è: un bottone rappresenta un comando.
- **Layout:** la posizione degli elementi all'interno della pagina è uno strumento importante di comunicazione.
- **Colori:** utili per focalizzare l'attenzione o suscitare emozioni.

- **Font:** leggibilità in relazione al tipo e alle caratteristiche del carattere.

#### 4.1.1 Usabilità

Col termine usabilità si intende l'efficacia, l'efficienza e la soddisfazione con cui determinati utenti eseguono determinati compiti in particolari ambienti. Più nello specifico in questo applicativo tali termini rappresentano:

- **Efficacia:** tutti i compiti richiesti in fasi di analisi possono essere eseguiti senza problematiche.
- **Efficienza:** si è cercato di usare al meglio le risorse disponibili per svolgere i compiti richiesti.
- **Soddisfazione:** si è avuta prova dell'accettabilità del funzionamento da parte dell'utente.

#### Personas e scenari

Sono state definite delle personas, cioè persone che rappresentano dei gruppi di utenti, i quali saranno gli utilizzatori del programma ed a ognuno di loro è stato associato uno scenario, il quale rappresenta le operazioni, il modo di muoversi nel sito e le modalità di utilizzo di ciascuno di loro. Le personas che sono state riscontrate:

- **Responsabile avvistamenti:** persona addetta alla modifica delle informazioni al fine di modificare, aggiungere o eliminare i dati.
- **Ricercatore:** persona che carica nuove informazioni, di un nuovo avvistamento direttamente dal portale.

A ciascuna di esse è associato uno scenarios:

- Responsabile avvistamenti: iscrizione al sito con possibilità di modifica degli avvistamenti, aggiunta e cancellazione.
- Ricercatore: Aggiunta di nuovi avvistamenti.

### **4.1.2 Design**

In questa sezione viene descritto il processo che ha portato alla creazione dell'interfaccia web. Al fine di progettare la migliore interfaccia con le rispettive caratteristiche che sono state elencate in fase di analisi sono stati fatti:

- Focus Group: è stato chiesto agli utilizzatori del sistema come prevedevano la realizzazione di determinate parti di interfaccia, in particolare per le schede dei singoli avvistamenti sono state proposte diverse modalità disposizione dei contenuti e sulla base di un sondaggio è stata scelta la migliore strategia.
- Mockup: sulla base delle informazioni sono stati creati degli schemi base in cui danno un'idea della struttura del sistema.
- Experience Prototyping: E' stata realizzata un semplice prototipo di tipo usa e getta al fine di verificare alcune parti incerte del programma in particolare la tabella degli avvistamenti e le specifiche di ogni avvistamento.

Gli obiettivi del design sono principalmente di essere:

- Mobile First

- User Centered
- Accessibile

## Mockup

Prima dell'inizio della realizzazione sono stati creati i mockup principali al fine definire la struttura delle pagine. In particolare sono stati creati i mockup per la pagina principale in cui vengono visualizzati tutti i gli avvistamenti e la pagina del singolo avvistamento in cui si definiscono le operazioni chiave. Il primo mockup che viene mostrato è quello della pagina principale in modalità mobile:

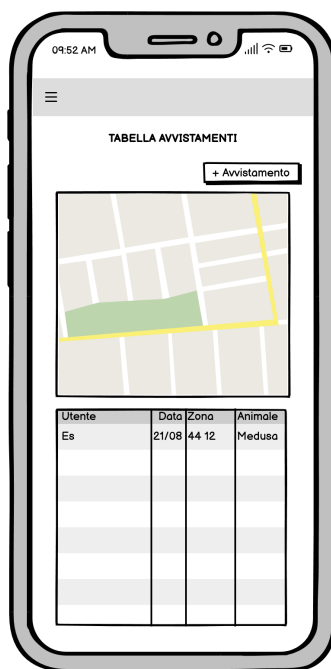


Figura 4.1: Mockup homepage del sito in versione mobile.

Da come si può vedere dell'immagine, si tratta di un design molto semplice in cui è presente un menù, una mappa e una tabella in cui saranno elencati gli avvistamenti che sono caricati attraverso l'apposito pulsante oppure attraverso l'applicazione mobile.

Nel secondo mockup viene mostrata stessa pagina web in versione desktop:

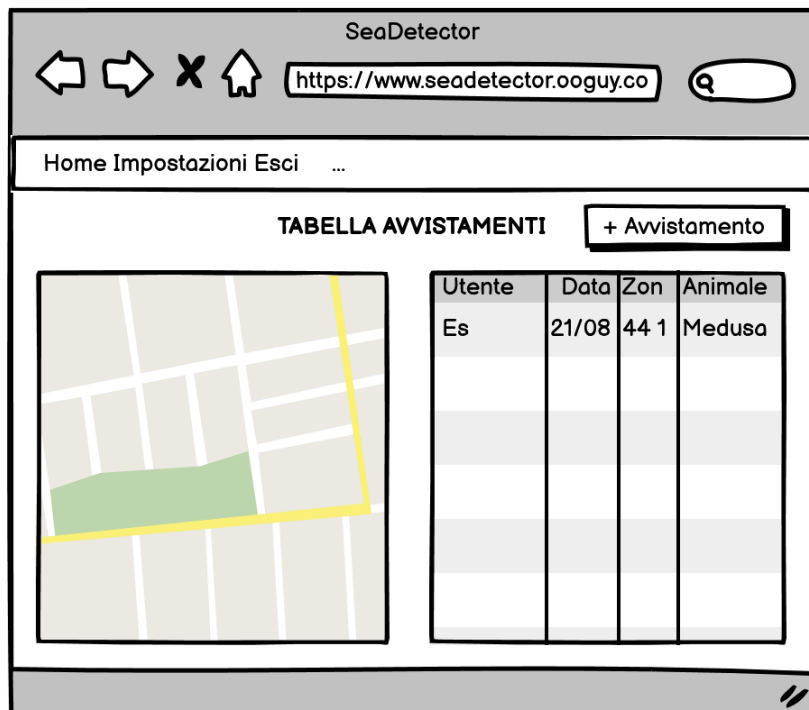


Figura 4.2: Mockup homepage del sito in versione desktop.

La pagina web contiene gli stessi elementi descritti nell'immagine precedente ma disposti solo in modo differente al fine di adattare lo spazio. Nel terzo mockup invece viene rappresentata la struttura della pagina degli avvistamenti in versione mobile:

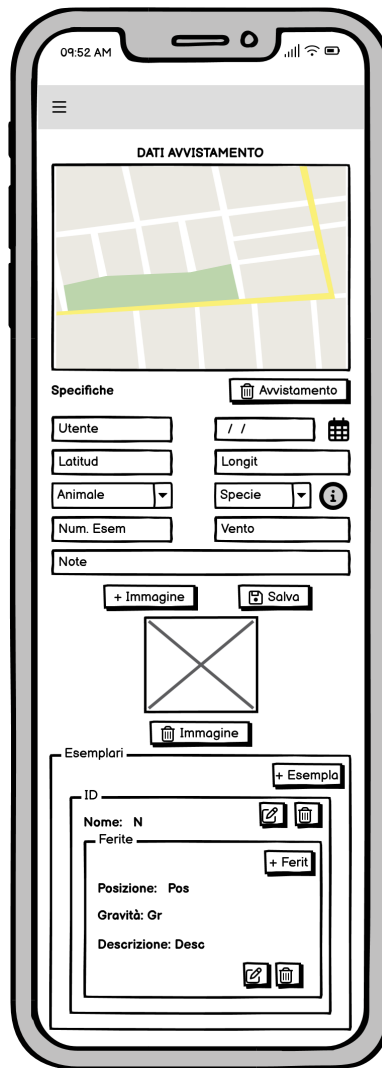


Figura 4.3: Mockup avvistamento del sito in versione mobile.



La pagina web contiene una semplice mappa in cui viene raffigurato il luogo dell'avvistamento, per ogni avvistamento vengono mostrati i dati, i quali sono modificabili al fine di variare le informazioni precedentemente inserite. In aggiunta per ogni avvistamento c'è la possibilità di aggiungere immagini e di cancellarle. Per ogni immagine possono essere aggiunte più sottoimmagini ognuna delle quali rappresenta un individuo al quale possono essere aggiunte o eliminare le ferite.

Nel quarto ed ultimo mockup viene raffigurata la struttura della pagina degli avvistamenti:

The mockup shows a web browser window titled 'SeoDetector' with the URL 'https://www.seodetector.ooguy.com/'. The page has a navigation bar with 'Home', 'Impostazioni', and 'Esci'. The main content area is titled 'DATI AVVISTAMENTO' and contains a map on the left and a form on the right. The form includes fields for 'Utente', 'Latitud', 'Longit', 'Animale', 'Specie', 'Num. Esem', and 'Vento'. There is also a 'Note' field and a '+ Immagine' button. Below the map is a placeholder for an image with an 'X' and a '+ Immagine' button. The bottom section is titled 'Esemplari' and contains a table with columns for 'ID', 'Nome', 'Ferite', 'Posizione', 'Gravità', and 'Descrizione'. There are buttons for '+ Esemplare', '+ Ferita', and '+ Immagine'.

Figura 4.4: Mockup avvistamento del sito in versione desktop.

Il contenuto della pagina è lo stesso della versione mobile solo con una disposizione diversa degli elementi.

### **4.1.3 Solution stack**

Un solution stack è un insieme di tecnologie software che vengono utilizzate per costruire un'applicazione o un sistema. È composto da una combinazione di software di sviluppo, linguaggi di programmazione, framework, database e altri strumenti necessari per creare un'applicazione completa e funzionante. Un solution stack ben sviluppato può rendere molto più semplice e veloce lo sviluppo di un'applicazione, poiché fornisce tutti gli strumenti necessari in un pacchetto completo.

Esistono molti solution stack differenti disponibili, alcuni dei quali sono sviluppati per specifici sistemi operativi o piattaforme, mentre altri sono più generici e possono essere utilizzati su molte piattaforme diverse.

### **LAMP**

Nel nostro caso è stato utilizzato il solution stack denominato LAMP (Linux, Apache, MySQL, PHP) è un popolare solution stack per lo sviluppo di applicazioni web su sistemi operativi Linux. Questo perché il database il server web Apache2 sono situati in una raspberry4 la quale ha un indirizzo di rete fisso al quale chiunque collegato via internet può collegarsi ed accedere al sito e ai suoi contenuti previa registrazione.

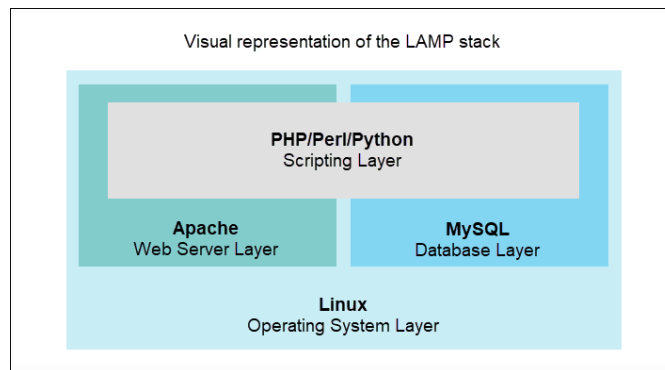


Figura 4.5: Stack LAMP

## PHP

Lato server si ha un linguaggio di programmazione denominato PHP esso è open-source, utilizzato per lo sviluppo di siti web dinamici e interattivi. PHP è un linguaggio di scripting server-side, il che significa che il codice PHP viene eseguito sul server, non sul browser del client. Questo rende il processo di sviluppo più efficiente, poiché l'elaborazione avviene sul server e i dati vengono inviati al browser solo una volta elaborati. Questo rende possibile creare siti web dinamici e interattivi, che possono essere personalizzati in base alle esigenze dell'utente.

Esso supporta anche molte funzionalità avanzate, come l'elaborazione di form HTML, la connessione a database, l'elaborazione di cookie e sessioni, e la gestione dei file. Questo rende possibile creare siti web dinamici che possono offrire molte funzionalità avanzate, come ad esempio la personalizzazione dei contenuti in base alle preferenze dell'utente.

## MYSQL

MySQL è un sistema di gestione del database relazionale open-source, utilizzato per gestire grandi quantità di informazioni. Il fatto che sia un sistema di gestione del database relazionale, significa che i dati sono organizzati in tabelle e ogni tabella ha una relazione con le altre. Questo rende il processo di gestione dei dati molto efficiente e semplice, poiché è possibile accedere rapidamente a informazioni specifiche utilizzando delle query. Esso è un sistema di gestione del database altamente scalabile, il che significa che può gestire grandi quantità di informazioni senza compromettere le prestazioni. Inoltre, è molto flessibile e facile da usare, il che significa che gli sviluppatori possono facilmente creare e gestire grandi quantità di informazioni.

### 4.1.4 Fruizione del servizio

Il servizio essendo di tipo centralizzato su un unico server è fruibile utilizzando le tre tecnologie che sono alla base del web.

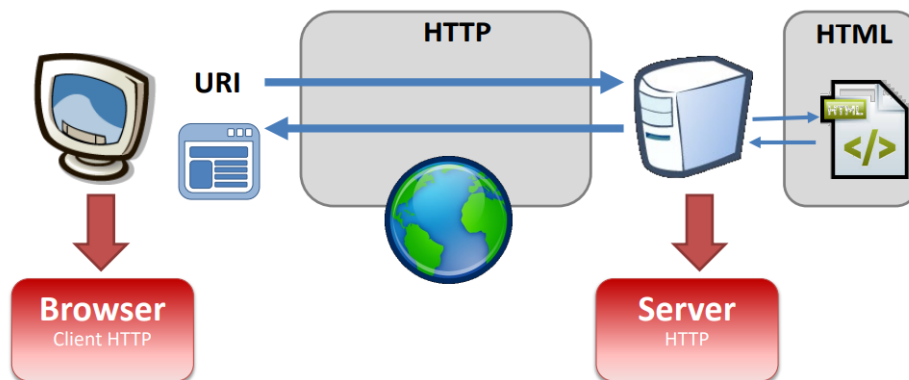


Figura 4.6: Tecnologie alla base del web

## URI

Lo Uniform Resource Identifier è utilizzato per identificare risorse in modo univoco e permanente. Questo è importante perché rende possibile accedere alle risorse su Internet in modo affidabile e coerente, indipendentemente dalla loro posizione. Nel nostro caso si può utilizzare l'URL *www.seadetector.ooguy.com* al fine di raggiungere il sito.

## HTTPS

L'Hypertext Transfer Protocol Secure è un'estensione di HTTP, il protocollo di base utilizzato per la trasmissione di informazioni su internet. Il funzionamento di HTTPS si basa sulla crittografia. Quando un utente accede a un sito web che utilizza HTTPS, i dati inviati e ricevuti vengono crittografati utilizzando un certificato SSL (Secure Sockets Layer) o TLS (Transport Layer Security). Questo certificato garantisce che i dati siano protetti da eventuali attacchi esterni e che solo l'utente e il sito web abbiano accesso ai dati scambiati. L'utilizzo di HTTPS è molto importante per la sicurezza delle informazioni personali e sensibili degli utenti. Ad esempio, quando si effettua un login i dati sensibili come le password vengono trasmessi attraverso internet. Se il sito web utilizza HTTPS, questi dati vengono protetti da eventuali attacchi esterni e non possono essere intercettati o utilizzati impropriamente.

In particolare nel nostro caso una volta installato e configurato il server web sulla Raspberry pi4 è stato utilizzato un software denominato certbot il quale facilita la creazione, l'installazione e l'utilizzo di certificati SSL per i siti web. Inoltre, Certbot si occupa automaticamente della gestione dei certificati, inclusa la rinnovazione automatica dei certificati prima della loro scadenza. Questo rende molto più semplice la gestione dei certificati

SSL, poiché gli sviluppatori non devono preoccuparsi di rinnovare manualmente i certificati.

Certbot permette di ottenere certificati gratuiti tramite Let's Encrypt, un'organizzazione che fornisce certificati SSL gratuiti a tutti.

## **HTML**

Quindi una volta contattato il server attraverso l'identificatore e utilizzando il protocollo di file transfert in modalità sicura il client e il server si scambiano le pagine in formato HTML.

HTML (Hypertext Markup Language) è un linguaggio di markup utilizzato per creare documenti web. HTML definisce la struttura e il contenuto della pagina web, utilizzando un insieme di tag per identificare diversi elementi della pagina, come il testo, le immagini, i link e molto altro.

### **4.1.5 Stile**

Per quanto riguarda lo stile del sito sono state utilizzate diverse tecnologie combinate insieme per cercare di avere la massima resa possibile.

## **CSS**

CSS (Cascading Style Sheets) è un linguaggio di marcatura utilizzato per descrivere l'aspetto di un documento HTML. Esso è un componente importante della creazione di siti web, poiché permette ai designer e agli sviluppatori di separare il contenuto di una pagina web dalla sua presentazione. In questo modo il contenuto di una pagina web può essere modificato senza influire sulla sua presentazione, rendendo più semplice la manutenzione e

l'aggiornamento del sito. CSS viene utilizzato per la creazione di layout complessi, animazioni, transizioni e effetti grafici.

## **Bootstrap**

Bootstrap è un framework di design per il web, che fornisce un insieme di componenti e strumenti per la creazione di siti web. Bootstrap è progettato per aiutare i designer e gli sviluppatori a creare progetti web più velocemente e con maggiore facilità, fornendo una serie di componenti predefiniti e stili per la creazione di interfacce utente. Bootstrap è basato su HTML, CSS e JavaScript, e offre una vasta gamma di componenti, tra cui barre di navigazione, moduli, tabelle, pulsanti, icone, griglie e molto altro. Esso rende molto più semplice la creazione di progetti web, invece di dover scrivere da zero il codice CSS e HTML per ogni progetto, i designer e gli sviluppatori possono utilizzare i componenti predefiniti di Bootstrap, che sono già ben testati e ottimizzati per una buona esperienza utente. Un altro aspetto fondamentale è che esso è basato sul principio denominato Mobile first e questo permette di creare pagine web adatte oltre che per pc anche per cellulari. Grazie alla combinazione di righe e colonne che insieme formano una griglia, permettono di modificare il layout della pagina in base alla grandezza del dispositivo che si utilizza.

### **4.1.6 Javascript e Librerie**

Per l'applicazione è stato usato Javascript come linguaggio di programmazione lato client esso è dinamico e interpretato, utilizzato principalmente per creare interazioni ed effetti visivi sulle pagine web. Esso è un linguaggio di programmazione completamente differente rispetto a HTML e CSS, che sono utilizzati per la creazione della struttura e del design delle pagine

web. JavaScript, d'altra parte, è utilizzato per creare interazioni e effetti dinamici all'interno della pagina, come ad esempio la validazione dei moduli e la creazione di animazioni.

## JQUERY

Nel progetto al fine di semplificare la creazione di determinate funzionalità è stato utilizzata una libreria JavaScript open source utilizzata per rendere più semplice e veloce lo sviluppo di pagine web interattive e applicazioni web. Essa è denominata JQUERY la sua principale caratteristica è la sua capacità di semplificare le attività di sviluppo web, come la manipolazione del DOM (Document Object Model), l'aggiunta di effetti e animazioni e la gestione degli eventi. Essa ha la caratteristica di essere molto compatibile con i browser più diffusi e offre un'ottima compatibilità anche con i dispositivi mobile.

## AJAX

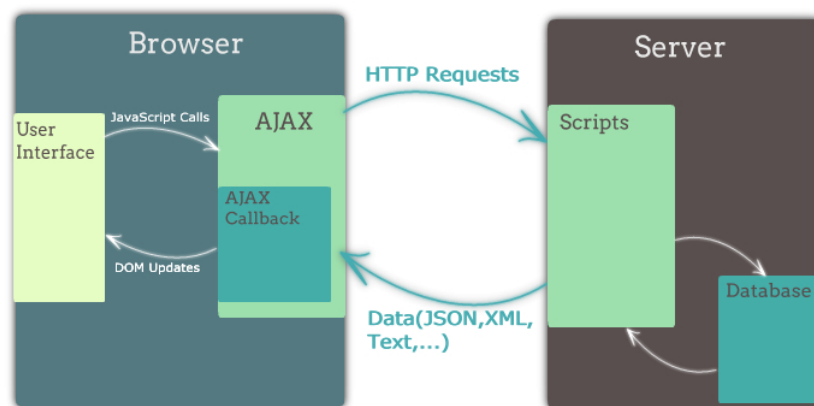


Figura 4.7: Funzionamento di AJAX



Nell'applicazione per richiedere dati dal database vengono eseguite delle chiamate Ajax asincrone al server. AJAX (Asynchronous JavaScript and XML) è una tecnologia utilizzata per creare applicazioni web dinamiche e interattive. Esso si basa sull'utilizzo di JavaScript e permette di aggiornare parti di una pagina web senza dover ricaricare l'intera pagina. Invece di inviare una richiesta al server per ottenere l'intera pagina web, AJAX consente di inviare richieste al server per ottenere solo i dati necessari. Questo significa che, ad esempio, se un utente clicca su un link o modifica una form, solo i dati necessari vengono inviati al server, e solo la parte della pagina che deve essere aggiornata viene effettivamente modificata. Questo rende l'interazione più veloce e fluida per l'utente, poiché non è necessario attendere il caricamento dell'intera pagina ogni volta che viene effettuata un'azione. Il server invierà i dati richiesti dal client in formato JSON.

### **PHPMailer**

Nel sito al fine di recuperare la password nel caso qualcuno se la dimentichi in fase di login è stata implementata una funzione che permetta di inviare email con all'interno la password temporanea dell'account. Per farlo è stata utilizzata la libreria PHP Mailer. Essa è una libreria open-source per PHP che permette di inviare email tramite il protocollo SMTP (Simple Mail Transfer Protocol). Questa libreria è stata sviluppata per semplificare il processo di invio di email da un sito web, e offre molte funzionalità avanzate, come ad esempio la gestione della codifica dei caratteri e l'invio di email con allegati. La libreria PHPMailer è molto semplice da utilizzare e offre un'interfaccia intuitiva per l'invio di email da un sito web.

Al fine di non dover comprare un dominio per le mail e dover gestire il server mail è stato utilizzato il servizio di posta elettronica di Libero.

### 4.1.7 Cookies

Esistono diverse tecniche utilizzate per raccogliere i dati online e costituire, attraverso esse, un profilo dell'utente. Una di queste tecniche prevede l'utilizzo di cookies, essi sono considerati delle stringhe di testo che il browser crea all'apertura di una pagina web sul computer dell'utente e che salvano dati dell'utente durante la navigazione di un sito web agevolandone l'utilizzo. Dei possibili esempi possono essere:

- Memorizzare le preferenze linguistiche.
- I dati di login.

I dati vengono memorizzati per essere poi ritrasmessi ai medesimi siti alla visita successiva dello stesso utente. Mediante i cookies è anche possibile monitorare la navigazione e raccogliere dati inerenti alle abitudini e le scelte personali degli utenti, consentendo così la creazione di profili dettagliati degli utenti. Un esempio è la personalizzazione delle inserzioni pubblicitarie sul browser. Generalmente un cookie contiene un attributo che indica la durata di vita e un numero generato in modo casuale che consente il riconoscimento dell'utente.

Esiste una specifica normativa, di derivazione europea, che disciplina l'utilizzo dei cookies al fine di tutelare le persone da forme di profilazione definite occulte e di consentirgli un minimo controllo sulla circolazione dei dati inerenti alla propria navigazione online.

Esistono tre tipologie di cookies:

1. **Cookies tecnici:** necessari per motivi tecnici e comportano una forma indispensabile (e molte volte temporanea) di memorizzazione di dati. Essi consentono la normale navigazione di un sito o la imple-

mentazione di un servizio, salvando solo le preferenze ed i criteri di navigazione di ogni utente.

2. **Cookies di profilazione:** elementi che non sono tecnicamente necessari per il funzionamento del sito. Alcuni esempi sono i cookies di tracciamento, oppure ai cookies che creano profili dell'utente per finalità pubblicitarie e vengono utilizzati per finalità di marketing.
3. **Cookies analitici:** consentono il monitoraggio dell'uso del sito da parte degli utenti e consentono il miglioramento del sito stesso. Talvolta anche i cookies analitici possono essere di terze parti.

Nel nostro applicativo sono stati utilizzati solo cookies tecnici, al fine di memorizzare i dati di login. Quindi secondo la normativa, i cookies tecnici possono essere usati anche senza chiedere il consenso dell'utente. Al fine dell'applicazione si è deciso di non utilizzare cookies di profilazione e nemmeno quelli analitici.

#### **4.1.8 Sicurezza**

Oltre all'utilizzo di un protocollo https si è data una particolare attenzione alla sicurezza sia per quanto riguarda la trasmissione dei dati tecnici degli avvistamenti che quelli personali come ad esempio le password.

##### **Invio dati**

Tutte le form, chiamate ajax inviano i dati in modalità POST. Il metodo "POST" è una delle due modalità principali per inviare dati tramite una richiesta HTTP. Quando si inviano dati utilizzando il metodo POST, i dati sono inclusi nella corpo della richiesta HTTP, anziché nell'URL come con

il metodo GET. Questo rende il metodo POST adatto per l'invio di dati sensibili o confidenziali, poiché i dati non sono visibili nell'URL e non possono essere facilmente intercettati o modificati.

### Criptaggio password

Per quanto riguarda la password si è usata una tecnica leggermente complicata:

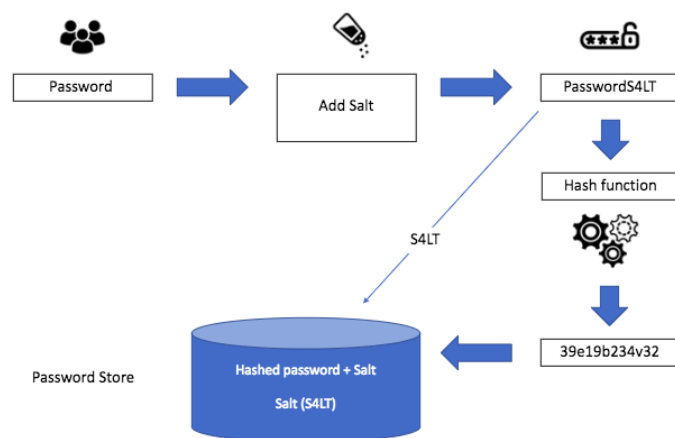


Figura 4.8: Struttura alla base della fase di criptaggio

1. In fase di registrazione l'utente inserisce due volte la password in modo da evitare errori di battitura.
2. Quando conferma i dati inseriti viene creata una chiave univoca basata sul timestamp del momento e criptata utilizzando una funzione hash SHA1 (SALT).
3. Una volta calcolata la chiave criptata viene utilizzata come chiave di codifica per cifrare la password inserita dell'utente attraverso

l'algoritmo di crittografia a chiave simmetrica denominato HMAC-SHA512.

4. Avendo la password e la chiave entrambe criptate, oltre a tutti gli altri dati, verrà salvato tutto nel database.
5. Una volta che l'utente esegue l'accesso, il client richiederà al server la chiave al fine di criptare la password e la invierà al server, a questo punto il server confronterà le due password in caso di uguaglianza permetterà l'accesso al servizio.

In questo modo la password dell'utente per l'accesso al sistema non verrà mai trasmessa in rete permettendo maggiore sicurezza in caso di attacchi di tipo Sniffing, man in the middle, o Spoofing.

## 4.2 Accessibilità

L'accessibilità informatica è la capacità per i sistemi informatici di fornire servizi e informazioni utilizzabili senza discriminazioni per tutti, compresi coloro che hanno bisogno di tecnologie assistive o configurazioni speciali a causa di disabilità. Questo viene realizzato nei limiti delle attuali conoscenze tecnologiche. Per quanto riguarda il sito si sono prese in considerazione i seguenti tipi di disabilità:

- Visive
  - Daltonismo
  - Ipovisione (tool assistivi di ingrandimento)
  - Cecità (accesso attraverso screen reader e voice browser)
- Motorie (Utilizzo di strumenti di input diversi)

### 4.2.1 WCAG

L'intero sito internet segue le linee guida create dal W3C per i diversi livelli di accessibilità. In particolare il WAI è l'acronimo di Web Accessibility Initiative ed è un'iniziativa del W3C. Il cui obiettivo è di promuovere la creazione di contenuti web accessibili e di garantire che tutti gli utenti, indipendentemente dalla loro abilità, possano accedere e interagire con i contenuti del web in modo efficiente e significativo. Il WAI lavora in stretta collaborazione con organizzazioni governative, industriali e di advocacy per garantire che le linee guida e le tecnologie sviluppate siano adatte alle esigenze di tutti gli utenti del web. Il suo impegno a supportare l'accessibilità del web ha contribuito a rendere il web un luogo più inclusivo e accessibile per tutti gli utenti.

Le linee guida prese in considerazione sono le WCAG che è l'acronimo di Web Content Accessibility Guidelines. Esse si basano su 4 principi:

- Percepibile
- Utilizzabile
- Comprensibile
- Robusto

Dai 4 principi discendono le 12 linee guida che forniscono indicazioni per rendere il contenuto più accessibile. per ciascuna di esse sono stati identificati una serie di criteri di successo che possono essere facilmente verificati. Questi criteri sono suddivisi in tre livelli di conformità: A (minimo), AA e AAA (massimo). Il livello A rappresenta il livello minimo di conformità, mentre il livello AAA rappresenta il massimo livello di conformità possibile.

Principles	Guidelines	Level A	Level AA	Level AAA
1. Perceivable	1.1 Text Alternatives	1.1.1		
	1.2 Time-based Media	1.2.1 – 1.2.3	1.2.4 – 1.2.5	1.2.6 – 1.2.9
	1.3 Adaptable	1.3.1 – 1.3.3		
	1.4 Distinguishable	1.4.1 – 1.4.2	1.4.3 – 1.4.5	1.4.6 – 1.4.9
2. Operable	2.1 Keyboard Accessible	2.1.1 – 2.1.2		2.1.3
	2.2 Enough Time	2.2.1 – 2.2.2		2.2.3 – 2.2.5
	2.3 Seizures	2.3.1		2.3.2
	2.4 Navigable	2.4.1 – 2.4.4	2.4.5 – 2.4.7	2.4.8 – 2.4.10
3. Understandable	3.1 Readable	3.1.1	3.1.2	3.1.3 – 3.1.6
	3.2 Predictable	3.2.1 – 3.2.2	3.2.3 – 3.2.4	3.2.5
	3.3 Input Assistance	3.3.1 – 3.3.2	3.3.3 – 3.3.4	3.3.5 – 3.3.6
4. Robust	4.1 Compatible	4.1.1 – 4.1.2		

Figura 4.9: Struttura delle linee guide coi rispettivi livelli.

Il WCAG è stato sviluppato con la collaborazione di una vasta comunità di esperti in accessibilità, tecnologia e advocacy e viene continuamente aggiornato per tener conto delle evoluzioni delle tecnologie e delle esigenze degli utenti. Esso è diventato un punto di riferimento per gli sviluppatori di siti web e per le organizzazioni che cercano di garantire l'accessibilità dei loro contenuti web.

## Linee guida

Nel progetto le linee guida a cui si è data maggiore importanza sono state:

*Principio Percettibile:*

- Linea guida 1.1 Alternative testuali: Si necessita di fornire alternative testuali per qualsiasi contenuto non di testo in modo che questo possa essere trasformato in altre forme fruibili secondo le necessità degli

utenti come Braille, stampa a caratteri ingranditi, sintesi vocale o in un linguaggio più semplice.

- Criterio di successo 1.1.1; Contenuti non testuali: Tutti i contenuti non testuali presentati all'utente hanno un'alternativa testuale equivalente. Questo viene fatto con l'utilizzo in modo opportuno dei tag `<figure>`, `<figcaption>` oppure con gli attributi `alt` oppure `longdesc` nei tag `<img>`.
- Linea guida 1.4 Distinguibile: Rendere più semplice agli utenti la visione dei contenuti, separando i contenuti in primo piano dallo sfondo.
  - Criterio di successo 1.4.3 Contrasto (minimo): La rappresentazione visiva del testo e di immagini contenenti testo ha un rapporto di contrasto di almeno 4.5:1.
- Linea guida 1.3 Adattabile: Creare contenuti che possano essere rappresentati in modalità differenti (ad esempio, con layout più semplici), senza perdere informazioni o la struttura.
  - Questa linea guida tratta molte componenti che sono fondamentali nella costruzione della struttura di un documento, in particolare gli elementi della categoria Sectioning hanno funzione strutturale. La struttura del documento è importante non solo per supportare la navigazione con l'ausilio di uno screen reader, ma anche per ottimizzare l'indicizzazione da parte dei motori di ricerca (SEO). La struttura del documento permette una navigazione più fluida tra sezioni, titoli e link.



In particolare gli screen reader permettono all'utente non vedenti di percorrere la pagine attraverso i tag di heading. Per questo motivo i tag non devono assolutamente avere uno scopo presentazionale ma la loro sequenza deve essere corretta al fine di aiutare gli utenti che utilizzano mezzi assistivi.

Un altro aspetto da tenere in considerazione sono le tabelle che sono presenti nel sito web, queste tabelle per essere accessibili devono utilizzare l'attributo id nei tag di heading(`<th>`) e l'attributo headers nei tag `<td>` al fine di permettere l'associazione di quella cella con il rispettivo titolo. In aggiunta nei tag di intestazione deve essere utilizzato l'attributo scope, esso permette di identificare lo scopo di quella cella. Per finire, al fine di far capire all'utente quali dati tale tabella rappresenta è utile utilizzare il tag `<caption>` dove viene scritta la didascalia della tabella.

*Principio Percettibile:*

- – Linea guida 3.3 Assistenza nell'inserimento: Aiutare gli utenti ad evitare gli errori ed agevolarli nella loro correzione. In particolare, Fornire etichette o istruzioni quando il contenuto richiede azioni di input da parte dell'utente.

La soluzione è l'utilizzo di label con attributo for e input con attributo id. Oppure inserire il tag input all'interno della label.

## **4.3 Struttura delle pagine**

## **Capitolo 5**

# **Documentazione Algoritmo Di Riconoscimento**

# Capitolo 6

## Documentazione Test

### 6.1 Test con l'utente

Al fine di effettuare operazioni di test direttamente con l'utente si è deciso di creare un prototipo per far interagire direttamente l'utente sull'intero sistema in beta-release. Gli aspetti testati sono:

- Il modello concettuale è sufficientemente rappresentato.
- Rispetto al progetto l'interfaccia è adatta e sono stati rispettati tutti gli standard.
- Possibilità di uso alternativo tra mouse e tastiera.
- Livello di interazione tra l'utente e l'interfaccia.
- Adeguato bilanciamento tra flusso predefinito e flessibilità.

## **Capitolo 7**

## **Conclusioni**

# Capitolo 8

## Utilità

Capitolo di elementi da inserire nei capitoli precedenti.

### 8.1 Ingegneria del software

#### 8.1.1 Verifica del software

Questa fase ha lo scopo di verificare che il sistema contenga le specifiche di progetto. Tale operazione non viene eseguita solo sul prodotto finale ma segue il progetto ad ogni suo passo.

Le tecniche di verifica che sono state utilizzate sono:

- **Di testing:** che attraverso delle prove sperimentali, su un insieme rappresentativo di situazioni, verificano il corretto funzionamento del sistema.
- **Di analisi:** attraverso l'analisi della struttura dei moduli e del codice che li realizza viene verificato il corretto funzionamento.

## Testing

”Le operazioni di testing possono individuare la presenza di errori nel software ma non possono dimostrarne la correttezza.”[1]

Come primo passo si sono individuati i casi significativi in cui applicare tale processo.

Le operazioni di testing si suddividono in:

1. **Testing in the small:** Riguardano porzioni specifiche di codice a cui è stata attribuita una particolare importanza.
2. **Testing in the large:** riguardano tutto il sistema.

Nel primo caso si è cercato di seguire il seguente criterio di copertura:

**Criterio di copertura delle decisioni e delle condizioni:** selezionare un insieme di test T tali che, a seguito dell'esecuzione del programma P su tutti i casi di T, ogni arco del grafo di controllo di P sia attraversato e tutti i possibili valori delle condizioni composte siano valutati almeno una volta. Nel secondo caso il sistema viene considerato come una black-box. L'insieme di test che verranno usati vengono selezionati sulla base delle specifiche di progetto, e grazie ad esse definiti i valori di input a cui corrisponderanno determinati valori di output.

I casi di testing in the large presi in considerazione sono:

- **Test di modulo:** controlla se l'implementazione di un modulo è corretta in base al comportamento esterno.
- **Test d'integrazione:** sottoparti del sistema vengono verificate sulla base del loro comportamento esterno.
- **Test di sistema:** controllo il comportamento dell'intero sistema sulla base del suo comportamento esterno.

### **8.1.2 Analisi**

Analizzare il software consiste nel capire le caratteristiche e le funzionalità. L'approccio per l'analisi del software è stato il Code Inspection.

In particolare si è posta molta attenzione all'analisi di flusso dei dati. Infatti, l'evoluzione del valore associato alle variabili durante l'esecuzione del programma è estremamente dinamica. Quindi si è usata la tecnica di associare ad ogni comando staticamente il tipo di operazioni eseguite sulle variabili. In questo modo avremo delle sequenze di comandi corrispondenti a possibili esecuzioni, le quali sono riconducibili staticamente a sequenze di tali operazioni. Quest'ultime verranno analizzate al fine di individuare delle possibili anomalie nel programma.

# Bibliografia

- [1] Khan, K. & Yadav, S. 2022, A Literature Review on Software Testing Techniques.