Documentazione Fotomap

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

2 Analisi 4

2.1 Analisi del dominio 4

2.2 Analisi e specifica dei requisiti 4

2.3 Use case 6

2.4 Pianificazione 6

2.5 Analisi dei mezzi 6

2.5.1 Software 6

2.5.2 Hardware 6

3 Progettazione 7

3.1 Design dell’architettura del sistema 7

3.2 Design dei dati e database 7

3.3 Design delle interfacce 7

3.4 Design procedurale 7

4 Implementazione 8

5 Test 8

5.1 Protocollo di test 8

5.2 Risultati test 9

5.3 Mancanze/limitazioni conosciute 9

6 Consuntivo 9

7 Conclusioni 9

7.1 Sviluppi futuri 9

7.2 Considerazioni personali 9

8 Bibliografia 9

8.1 Bibliografia per articoli di riviste: 9

8.2 Bibliografia per libri 9

8.3 Sitografia 10

9 Allegati 10

# Introduzione

## Informazioni sul progetto

* Titolo Progetto: Fotomap
* Allievo: Gioele Chiodoni
* Docente responsabile: Maurizio di Florio
* Scuola: Arti e Mestieri Trevano, sezione Informatica
* Data di inizio e fine: 27.01.2025 – 04.04.2025
* Data di presentazione: 07.04.2025 – 17.04.2025

## Abstract

The growing focus on biodiversity and the need to monitor wildlife and plant species require advanced digital tools for data collection and analysis. This project aims to develop a web platform dedicated to sharing sightings of wild animals and plants, targeting biologists, nature photographers, and enthusiasts.

The main goal is to provide an efficient system for collecting and consulting georeferenced data, facilitating scientific analysis and the creation of environmental statistics. The platform will allow users to upload photographs with detailed information, enhancing the quality of data available for research.

The implementation is based on a modern architecture, utilizing Java Spring Boot for the backend, Angular for the frontend, and MySQL as the relational database. The application will ensure high performance, secure data access, and an intuitive user interface.

The adoption of these technologies will guarantee scalability, reliability, and a seamless user experience, contributing to the preservation and enhancement of biodiversity through the intelligent use of data.

## Scopo

Questo progetto ha un duplice scopo: operativo e didattico. Operativamente, mira a sviluppare una piattaforma web per raccogliere e consultare avvistamenti di animali e piante selvatiche, facilitando l’analisi scientifica e la creazione di statistiche ambientali. Gli utenti potranno inserire segnalazioni georeferenziate con immagini e dettagli, rendendo l’accesso ai dati rapido ed efficace.

Dal punto di vista didattico, il progetto serve a migliorare la conoscenza di Spring Boot e Angular, approfondendo lo sviluppo di applicazioni web moderne, scalabili e sicure.

## Analisi del dominio

Ho scelto di sviluppare un'applicazione web per la raccolta di dati sugli avvistamenti di fauna e flora selvatica, poiché attualmente non esiste una piattaforma unica, accessibile e intuitiva per questo scopo. Il sistema permetterà agli utenti di registrare e consultare i dati in modo semplice, offrendo strumenti per generare statistiche e ottenere informazioni rapidamente, senza dover cercare tra forum o documenti sparsi.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Accesso e login |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Un utente per poter inserire avvistamenti deve effettuare l’accesso. |
| **Sotto requisiti** | |
| **001** | Maschera di login. |
| **002** | Si devono poter vedere gli avvistamenti senza effettuare il login. |
| **003** | Dovrà essere possibile cercare avvistamenti senza effettuare il login. |
| **004** | Nel caso in qui l’utente dimentichi la password dovrà poterla reimpostare. |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Registrazione |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Un utente senza account deve poter registrarsi tramite email. |
| **Sotto requisiti** | |
| **001** | Maschera di registrazione. |
| **002** | L’email dovrà essere verificata dell’email. |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Visualizzazione avvistamenti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’utente dovrà poter visualizzare gli avvistamenti. |
| **Sotto requisiti** | |
| **001** | Dovrà essere possibile applicare dei filtri di ricerca. |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Inserimento avvistamenti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’utente dovrà poter inserire degli avvistamenti |
| **Sotto requisiti** | |
| **001** | Dovrà essere possibile se presenti prendere i dati dalla fotografia. |
| **002** | L’utente dovrà poter inserire i dati manualmente anche se presenti nella foto. |

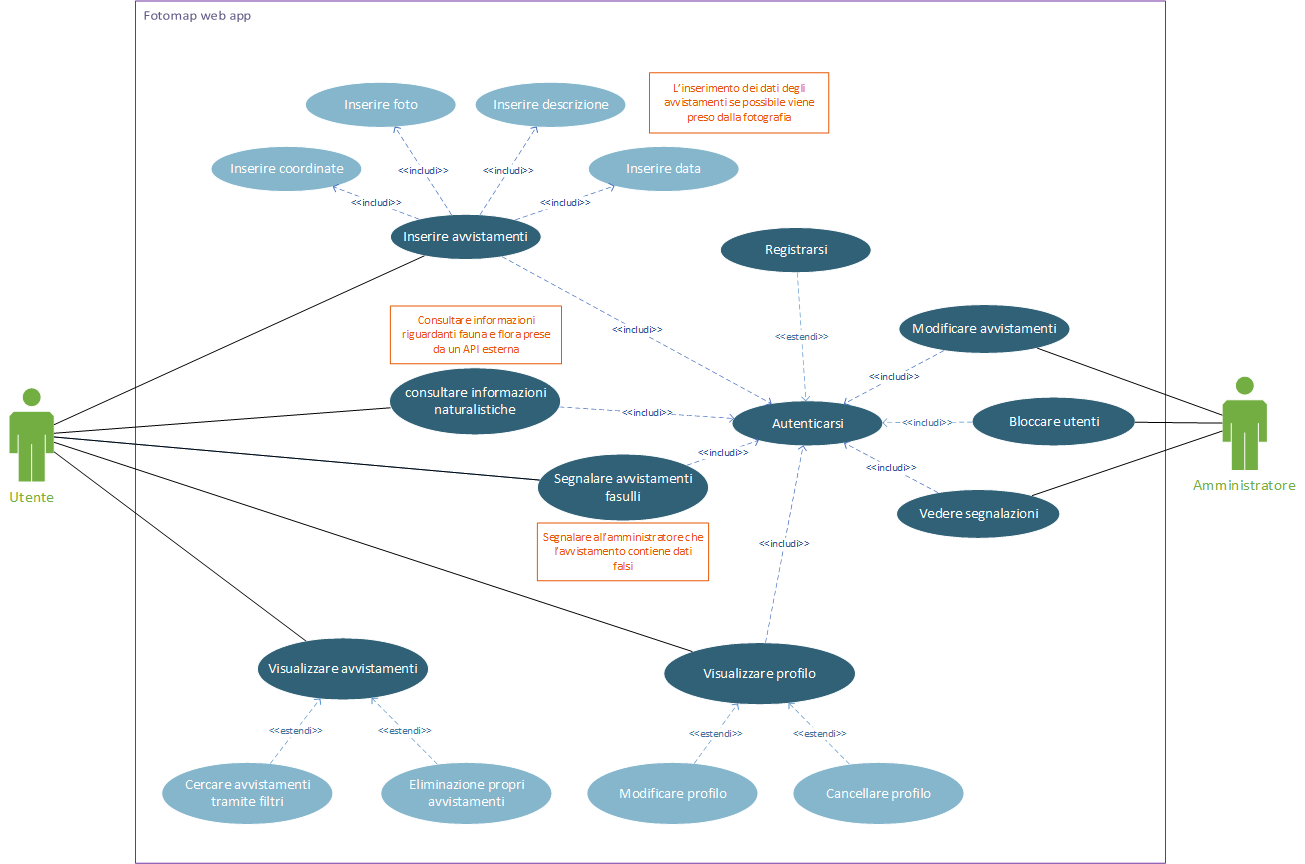
|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Segnalazione avvistamenti fasulli |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Gli utenti registrati potranno segnalare degli avvistamenti fasulli o con dati errati. |
| **Sotto requisiti** | |
| **001** | L’amministratore dovrà poter visualizzare e modificare gli avvistamenti segnalati. |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Consultare informazioni |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’utente dovrà poter visualizzare informazioni su fauna e flora. |
| **Sotto requisiti** | |
| **001** | Dovrà poter cercare animali o piante in modo specifico. |
| **002** | Dovrà essere possibile visualizzare avvistamenti correlati alla ricerca fatta. |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Modifica informazioni personali |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’utente dovrà poter modificare i propri dati personali. |
| **Sotto requisiti** | |
| **001** | L’utente dovrà poter modificare il proprio username, password ed immagine profilo. |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | Gestione segnalazioni |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’amministratore dovrà poter gestire le segnalazioni degli avvistamenti fasulli. |
| **Sotto requisiti** | |
| **001** | Dovrà poter modificare o eliminare gli avvistamenti segnalati |
| **002** | Dovrà poter bloccare utenti che inseriscono dati falsi. |

## Use case



Come si può notare dallo schema l’utente può eseguire diverse operazioni sull’applicativo, molte di queste però richiedono all’utente di autenticarsi. L’unica operazione che non richiede una registrazione è la visualizzazione degli avvistamenti.

Per quanto riguarda la visualizzazione degli avvistamenti l’utente può filtrare gli avvistamenti già presenti o cancellare i propri.

Mentre per l’inserimento degli avvistamenti l’utente avrà la possibilità di inserire i dati manualmente o ricavarli automaticamente grazie ai metadati della fotografia.

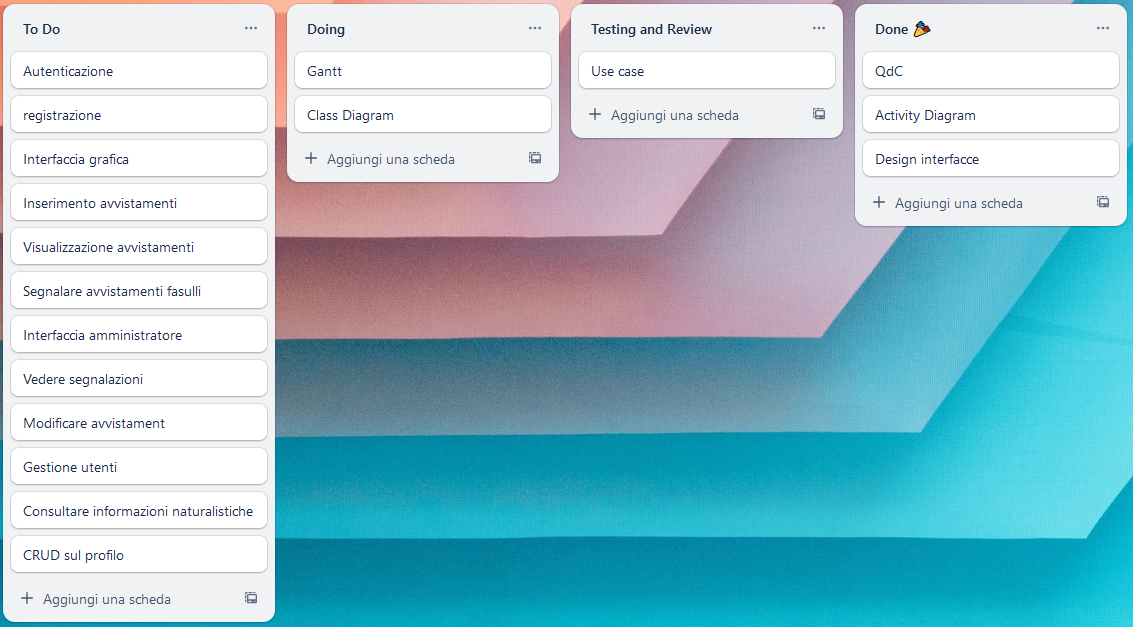
Inoltre avrà la possibilità di consultare informazioni sulla fauna e la fora di modo da poter ottenere informazioni rapidamente.

Per quanto riguarda gli avvistamenti gli utenti che avranno eseguito l’accesso nel caso in cui visualizzino un avvistamento sospetto potranno segnalarlo all’amministratore. In seguito l’amministratore potrà vedere le segnalazioni e modificare gli avvistamenti segnalati e nel caso bloccare gli utenti.

## Pianificazione

Per quanto riguarda la pianificazione ho deciso di usare una metodologia di lavoro Kanban, ho scelto di progettare il mio lavoro in questo modo per poter essere più flessibile essendo che ho scelto di usare tecnologie e framework che non conosco benissimo e quindi non so esattamente quanto tempo mi richiederà ogni attività. Per approcciare al meglio la metodologia Kanban ho usato un Gantt in cui ho suddiviso il progetto in macro task. Queste le ho riportate tali nella mia Kanban board. Per pianificare al meglio il lavoro ho inoltre deciso di suddividere ogni task in micro attività all’inizio di ogni task. Questo mi consente di essere più flessibile in base all’evoluzione del progetto. Alla fine delle task per testarle al meglio ho pensato di svolgere dei test sul API e sul frontend. Questo mi consente di essere più sicuro del corretto completamento di ogni task.

Per organizzare al meglio le task mi faccio aiutare da una Kanban board su Trello di modo da tener traccia delle task e delle micro attività.



## Analisi dei mezzi

Per realizzare questo progetto ho usato il computer fornito dalla scuola.

### Software

* Visual Studio Code v. 1.96.4
* IntelliJ IDEA 2024.3.2.2 (Ultimate Edition), come IDE per il backend
* WebStorm 2024.3.2.1, come IDE per il backend
* Firefox v. 129.0.2, per testare la visualizzazione del sito
* GanttProject v. 3.3.3312, per la realizzazione del gantt
* Microsoft Visio, per fare lo use case
* Microsoft Word e Obsidian, per redigere la documentazione, il QdC e il diario
* Trello.com, per la gestione delle task
* Postman v. 11.30.4 per testare gli endpoint dell’API

### Hardware

Computer scolastico:

* Processore: Intel(R) Core (TM) i7-9700 CPU @ 3.00GHz 3.00 GHz
* RAM: 32.0 GB (31.8 GB utilizzabile)
* Sceda video: NVIDIA GeForce RTX 2060
* SSD: 512 GB
* SSD personale: Samsung T7 1TB
* Monitor 1: HP E24i G4 1920x1200 60Hz
* Monitor 2: ACER KA220HQ 1920x1080 60Hz

# Progettazione

Questo capitolo descrive esaustivamente come deve essere realizzato il prodotto fin nei suoi dettagli. Una buona progettazione permette all’esecutore di evitare fraintendimenti e imprecisioni nell’implementazione del prodotto.

## Design dell’architettura del sistema

Descrive:

* La struttura del programma/sistema lo schema di rete...
* Gli oggetti/moduli/componenti che lo compongono.
* I flussi di informazione in ingresso ed in uscita e le relative elaborazioni. Può utilizzare *diagrammi di flusso dei dati* (DFD).
* Eventuale sitemap

## Design dei dati e database

Descrizione delle strutture di dati utilizzate dal programma in base agli attributi e le relazioni degli oggetti in uso.

Schema E-R, schema logico e descrizione.

Se il diagramma E-R viene modificato, sulla doc dovrà apparire l’ultima versione, mentre le vecchie saranno sui diari.

## Design delle interfacce

Descrizione delle interfacce interne ed esterne del sistema e dell’interfaccia utente. La progettazione delle interfacce è basata sulle informazioni ricavate durante la fase di analisi e realizzata tramite mockups.

## Design procedurale

Descrive i concetti dettagliati dell’architettura/sviluppo utilizzando ad esempio:

* Diagrammi di flusso e Nassi.
* Tabelle.
* Classi e metodi.
* Tabelle di routing
* Diritti di accesso a condivisioni …

Questi documenti permetteranno di rappresentare i dettagli procedurali per la realizzazione del prodotto.

# Implementazione

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato.

Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.

In questa parte è richiesto l’inserimento di codice sorgente - Print Screen - di maschere solamente per quei passaggi particolarmente significativi e/o critici.

Inoltre, dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.

Non deve apparire nessuna forma di guida d’uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.

Per eventuali dettagli si possono inserire riferimenti ai diari.

# Test

## Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l’adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-012 | **Nome:** | Import a card with KIC, KID and KIK keys, but not shown with the GUI |
| **Descrizione:** | Import a card with KIC, KID and KIK keys with no obfuscation, but not shown with the GUI | | |
| **Prerequisiti:** | Store on local PC: Profile\_1.2.001.xml (appendix n\_n) and Cards\_1.2.001.txt (appendix n\_n).  PIN (OTA\_VIEW\_PIN\_PUK\_KEY) and ADM (OTA\_VIEW\_ADM\_KEY) user right not set. | | |
| **Procedura:** | 1. Go to “Cards manager” menu,  in main page click “Import Profiles” link, Select the “1.2.001.xml” file, Import the Profile 2. Go to “Cards manager” menu,  in main page click “Import Cards” link, Select the “1.2.001.txt” file, Delete the cards,  Select the “1.2.001.txt” file, Import the cards 3. Research the “41795924770” Card, Click the imsi card link Check the card details 4. Execute the SQL: SELECT imsi, dir, keyset, cntr, rawtohex(kickey), rawtohex(kidkey), rawtohex(kikkey), rawtohex(chv), rawtohex(dap)FROM otacardkey a where imsi='340041795924770' ORDER BY keyset; | | |
| **Risultati attesi:** | Keys visible in the DB (OtaCardKey) but not visible in the GUI (Card details) | | |

## Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l’errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l’errore con eventuali ipotesi di correzione.

## Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

# Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap. 1.7) (ad esempio Gantt consuntivo).

# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc.

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc.

# Bibliografia

## Bibliografia per articoli di riviste:

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo dell’articolo (tra virgolette),
3. Titolo della rivista (in italico),
4. Anno e numero
5. Pagina iniziale dell’articolo.

## Bibliografia per libri

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo del libro (in italico),
3. ev. Numero di edizione,
4. Nome dell’editore,
5. Anno di pubblicazione,
6. ISBN.

## Sitografia

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (GG-MM-AAAA).

**Esempio:**

* http://standards.ieee.org/guides/style/section7.html, *IEEE Standards Style Manual*, 07-06-2008.

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Codici sorgente/documentazione macchine virtuali
* Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
* Documentazione di prodotti di terzi
* Eventuali guide utente / Manuali di utilizzo
* Mandato e/o QdC
* Prodotto
* …