Documentazione Progetto  
 Astronomic Picture of the Day

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

2.2.1 Spiegazione elementi tabella dei requisiti: 6

2.3 Use case 7

2.4 Pianificazione 7

2.5 Analisi dei mezzi 8

2.5.1 Software 8

2.5.2 Hardware 8

3 Progettazione 9

3.1 Design dell’architettura del sistema 9

3.2 Design dei dati e database 9

3.3 Design delle interfacce 10

3.3.1 Interfaccia Login 10

3.3.2 Interfaccia Home page 11

3.3.3 Interfaccia Preferiti e Cronologia 11

3.4 Design procedurale 12

4 Implementazione 13

4.1 Database 13

4.1.1 Utente 13

4.1.2 Preferiti e Cronologia 13

4.2 UML 13

4.3 Codice 13

4.4 Sicurezza 13

4.5 Design delle interfacce 13

5 Test 13

5.1 Protocollo di test 13

5.2 Risultati test 15

5.3 Mancanze/limitazioni conosciute 15

6 Consuntivo 15

7 Conclusioni 15

7.1 Sviluppi futuri 15

7.2 Considerazioni personali 15

8 Glossario 15

9 Bibliografia 16

9.1 Bibliografia per articoli di riviste: 16

9.2 Bibliografia per libri 16

9.3 Sitografia 16

10 Allegati 16

# Introduzione

## Informazioni sul progetto

## Abstract

Il progetto vuole creare una pagina Web dove l’utente può vedere la foto astronomica del giorno, o di un giorno a sua scelta, i crediti, una descrizione dell’immagine, la possibilità di scaricare l’immagine e la cronologia di ricerca.

Può contenere alcuni o tutti gli elementi seguenti:

* **Background/Situazione iniziale**
* **Descrizione del problema e motivazione**: Che problema ho cercato di risolvere? Questa sezione dovrebbe includere l'importanza del vostro lavoro, la difficoltà dell'area e l'effetto che potrebbe avere se portato a termine con successo.
* **Approccio/Metodi**: Come ho ottenuto dei progressi? Come ho risolto il problema (tecniche…)? Quale è stata l’entità del mio lavoro? Che fattori importanti controllo, ignoro o misuro?
* **Risultati**: Quale è la risposta? Quali sono i risultati? Quanto è più veloce, più sicuro, più economico o in qualche altro aspetto migliore di altri prodotti/soluzioni?

## Scopo

Lo scopo didattico del progetto è riuscire a gestire nel modo più ottimale possibile un progetto IT con le risorse datoci dalla scuola. Questo servirà successivamente a prepararmi a futuri progetti e all’esame pratico di fine scuola.

Lo scopo operativo invece riguarda il creare un applicativo Web dove si possano visualizzare le *Astronomic Picture of the Day*, ovvero le foto astronomiche del giorno fatte dalla Nasa, con tutte le informazioni a riguardo. Dovrà esserci la possibilità di ricercare la foto di una specifica data e salvarla. Si dovrà anche visualizzare la cronologia delle immagini ricercate.

# Analisi

## Analisi del dominio

L’Applicativo potrà venir utilizzato da chiunque, permettendo però anche di eseguire il login così da vedere il proprio account, su qualunque Computer o dispositivo mobile. Principalmente si vuole semplificare la visione e la ricerca della Astronomical Picture of the Day (APOD). Attualmente esiste già un sito che fa questo lavoro, però non c’è la possibilità di scegliere l’APOD da vedere senza andare sul sito apposito con tutte le APOD scattate in ordine cronologico e non si può salvare le proprie foto preferite.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-1** | |
| **Nome** | Accesso |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Solo chi ha un account potrà accedere al sito |
| **Sotto requisiti** | |
| **001** | Creare un Database e gli utenti per il sito |

|  |  |
| --- | --- |
| **ID: REQ-2** | |
| **Nome** | API |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Bisogna utilizzare l’API APOD della Nasa per ricavare i dati delle foto |

|  |  |
| --- | --- |
| **ID: REQ-3** | |
| **Nome** | Download |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Sezione per scaricare una o più immagini contemporaneamente |

|  |  |
| --- | --- |
| **ID: REQ-4** | |
| **Nome** | Cronologia |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Sotto requisiti** | |
| **001** | Creare un file JSON e inserirci le informazioni su tutte le foto ricercate dall’utente |

|  |  |
| --- | --- |
| **ID: REQ-5** | |
| **Nome** | Preferiti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Sotto requisiti** | |
| **001** | Creare un file JSON per ogni utente |
| **002** | Modificare il file JSON dell’utente |

|  |  |
| --- | --- |
| **ID: REQ-6** | |
| **Nome** | Filtraggio |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Permettere di cercare una foto per data o nome |

|  |  |
| --- | --- |
| **ID: REQ-7** | |
| **Nome** | Zoom |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Sotto requisiti** | |
| **001** | Mettere la foto in un contenitore, così che quando si passa col mouse si possa fare uno zoom |

|  |  |
| --- | --- |
| **ID: REQ-8** | |
| **Nome** | Sequenza immagini |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Le foto della settimana si alternano in modo fluido |

|  |  |
| --- | --- |
| **ID: REQ-9** | |
| **Nome** | Stampare foto |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Sotto requisiti** | |
| **001** | Selezionare le foto da stampare |
| **002** | Impaginare correttamente le foto preferite e stamparle |

### Spiegazione elementi tabella dei requisiti:

**ID**: identificativo univoco del requisito

**Nome**: breve descrizione del requisito

**Priorità**: indica l’importanza di un requisito nell’insieme del progetto, definita assieme al committente. Ad esempio, poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

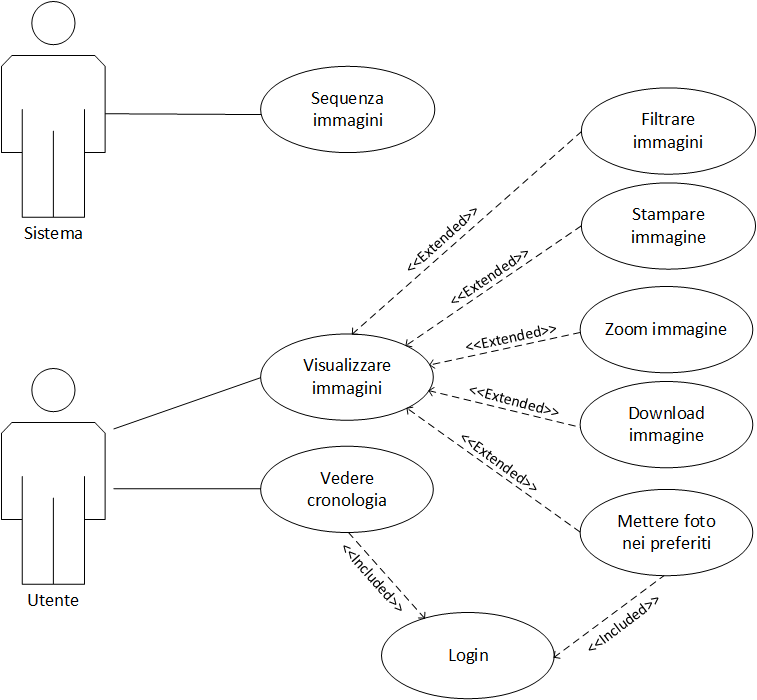
**Versione**: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l’ultima versione, mentre le vecchie dovranno essere inserite nei diari.

**Note**: eventuali osservazioni importanti o riferimenti ad altri requisiti.

**Sotto requisiti**: elementi che compongono il requisito.

## Use case



## Pianificazione

## Analisi dei mezzi

### Software

Per questo progetto avrò bisogno dell’API APOD della Nasa, MySQL e Apache (XAMPP), Project per il gantt e Visual studio Code per la parte di programmazione.

### Hardware

Per questo progetto avrò bisogno del PC di scuola e un Server dove mettere il Database.

# 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

|  |
| --- |
|  |
| C:\Users\kamil.siddiqui\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\39EF401.tmp |

## Design delle interfacce

### Interfaccia Login

Questa sarà la prima pagina che l’utente vedrà quando apre il sito, se accede potrà vedere anche la pagina preferiti e cronologia, altrimenti avrà solo accesso alla home page.

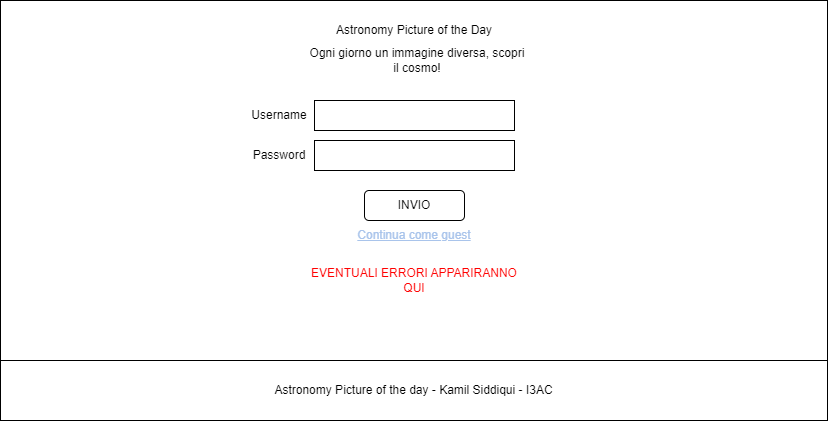


Figura : Pianificazione della GUI del Login

### Interfaccia Home page

Questa sarà la home page, da cui si potrà vedere l’immagine del giorno o quella scelta dall’utente, e se disponibili quella del giorno precedente e successivo.

Da questa pagina si potrà andare alla pagina di login premendo il menu a scomparsa in alto a sinistra.  
Da qui si aprirà un menu che permetterà anche di andare alla pagina dei preferiti o della cronologia

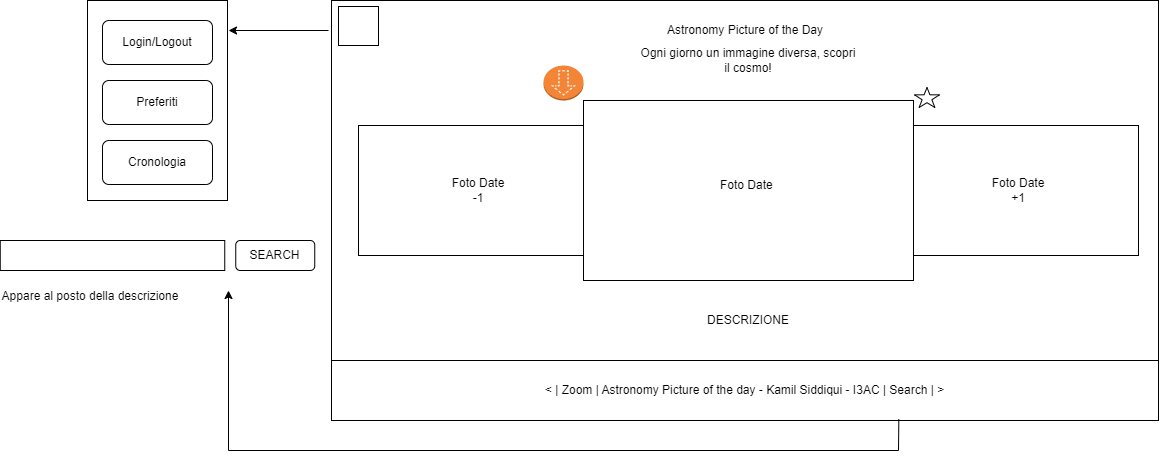


Figura : Pianificazione della GUI della Home page

### Interfaccia Preferiti e Cronologia

Questa pagina mostrerà le foto preferite dell’utente oppure le ultime 10 foto visualizzate dall’utente, da qui si potrà sempre andare alla home page,alla pagina preferiti o cronologia ed effettuare il logout, essendo questa una pagina solo per utenti loggati.

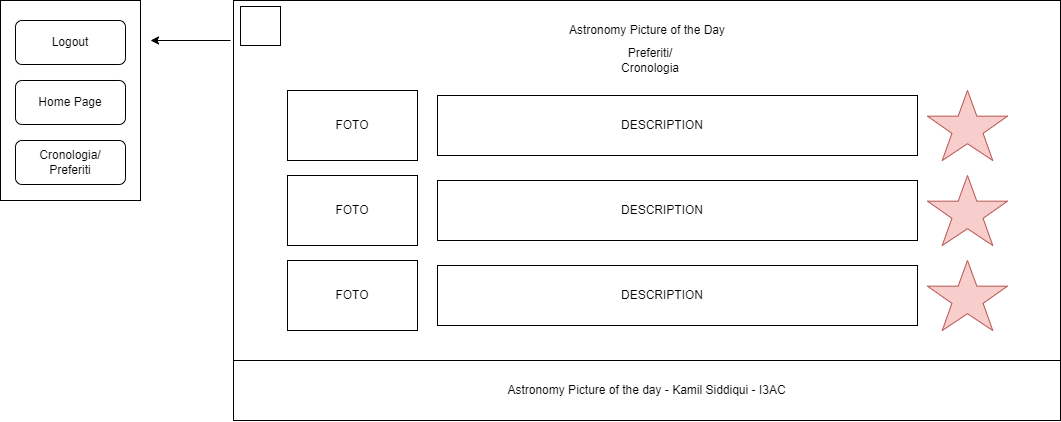


Figura : Pianificazione della page dei Preferiti e cronologia

## Design procedurale

# Implementazione

## Database

### Utente

|  |
| --- |
|  |

Il sito viene utilizzato dagli utenti, questi devono poter vedere le loro foto preferite e le foto viste recentemente all’interno del sito, per permettere questo ho create la tabella utente, in modo che ogni utente possa avere un account personale. Per fare il login utilizzo un Username e una password che verrà hashata usando MD5, tutti gli utenti vengono aggiunti direttamente dalla pagina di registrazione di PHP. ID viene usato come key e verrà anche usato come foreign key per le altre tabelle del Database.

### Preferiti

|  |
| --- |
|  |

La tabella “Preferito” conterrà le informazioni importanti sulle foto che l’utente metterà tra i preferiti.

Data e Utente\_Id sono le chiavi essendo che un utente può avere più foto preferite diverse, e la stessa foto può essere la preferità di più utenti.

Ci sarà un limite di foto preferite, quando raggiungerà il massimo (15/20 foto) l’utente avrà un avviso e dovrà togliere delle foto se ne vorrà aggiungere altre. Url serve per salvare l’URL della foto e mettere un’anteprima della foto. Date, oltre che venir usato come chiave, serve per fare un eventuale richiesta all’API e avere tutte le informazioni sull’immagine.

### Cronologia

|  |
| --- |
|  |

La tabella “Cronologia” conterrà le informazioni importanti sulle ultime 15/20 foto viste dall’utente. A differenza della tabella “Preferito”, questa come chiave usa Id e Utente\_Id, questo perché a differenza di “Preferito”, se superi il numero massimo di foto elimino io la foto più vecchia vista (quella con l’ID più basso).

Url serve per salvare l’URL della foto e mettere un’anteprima della foto. Date serve per fare un eventuale richiesta all’API e avere tutte le informazioni sull’immagine.

## UML

### Struttura cartelle

|  |
| --- |
| * APOD\_API-2024-2025   + 1\_Qdc   + 2\_Abstract   + 3\_Documentazione (word e pdf)   + 4\_Diari     - 12. Dicembre     - 11. Novembre     - 10. Ottobre     - 9. Settembre   + 5\_Applicativo     - css     - img     - js   + 7\_Allegati     - DB     - Pianificazione\_Pagine   + 8\_Manuali |

Questa è la strutta delle cartelle. Inizialmente c’è la cartella del progetto **APOD\_API-2024-2025**, questa cartella contiene 7 cartelle, **1\_Qdc** che contiene il quaderno dei compiti. **2\_Abstract** contiene l’abstract del progetto. **3\_Documentazione (word e pdf)** contiene la documentazione del progetto in formato .docx e .pdf.  
**4\_Diari** contiene una cartella per ogni mese, dentro di esse ci sono i relativi diari di quel mese. **5\_Applicativo** contiene delle sottocartelle, css contiene tutti i file di stile per la pagina php, img contiene tutte le immagini usate nel sito, come la favicon, lo sfondo e l’immagine che uso se manca l’url di una foto e js contiene due file di tipo javascript, uno che gestise il layout della pagina html e uno che gestisce le richieste all’api.

## Codice

## Sicurezza

## Design delle interfacce

# 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.

# Glossario

Inserite una semplice tabella con due colonne che spieghi i termini specifici del progetto (lista dei termini in ordine alfabetico A-Z)

Esempio:

|  |  |
| --- | --- |
| **Termine** | **Descrizione** |
| AJAX | **Asynchronous JavaScript And XML**: una tecnica che permette di eseguire richieste ed ottenere dati da una pagina web in modo asincrono. |
| CSS | **Cascading Style Sheets**: linguaggio che permette di definire il layout e la grafica di una pagina web. |

# 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
* …