Deduplicator

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

Analisi 4

1.4 Analisi del dominio 4

1.5 Analisi e specifica dei requisiti 4

1.6 Use case 6

1.7 Pianificazione 6

1.8 Analisi dei mezzi 6

1.8.1 Software 6

1.8.2 Hardware 6

2 Progettazione 7

2.1 Design dell’architettura del sistema 7

2.2 Design dei dati e database 7

2.3 Design delle interfacce 7

2.4 Design procedurale 7

3 Implementazione 8

4 Test 8

4.1 Protocollo di test 8

4.2 Risultati test 9

4.3 Mancanze/limitazioni conosciute 9

5 Consuntivo 9

6 Conclusioni 9

6.1 Sviluppi futuri 9

6.2 Considerazioni personali 9

7 Bibliografia 9

7.1 Bibliografia per articoli di riviste: 9

7.2 Bibliografia per libri 9

7.3 Sitografia 9

8 Allegati 10

# Introduzione

## Informazioni sul progetto

Questo progetto è estensione del progetto del primo semestre del anno scolastico 2019/2020 dove è stato creato un servizio per eseguire la ricerca e la gestione di duplicati di uno o più percorsi definiti dall’utente. Nel progetto del primo semestre è stata creata una GUI primitiva è incompleta.

Il progetto ha inizio in data 23.01.2020 e finisce il 06.04.2020, il docente responsabile è Geo Petrini.

## Abstract

The project that was terminated in the first semester of the 2019/2020 year consisted in the creation of a service to discover and manage duplicate files in one or more paths defined by the user, it had a primitive and incomplete GUI.   
This project has as the primary objective the creation of a modern GUI in order to use and manage all of the funcionalities of the service. The service has to have a modular approach, allowing additions to be implemented in the future. Furthermore there are some bug fixes to be made to the service.

## Scopo

Con questo progetto si vuole creare una interfaccia grafica, moderna, per l‘utente con il scopo di facilitare l’utilizzo e la gestione del servizio. L’implementazione deve avere un approccio modulare per permettere l’aggiunta di ulteriori moduli in futuro come la scansione di tracce audio o immagini.

## Analisi

## Analisi del dominio

Al momento il servizio creato nel primo progetto del semestre ha una GUI primitiva e incompleta, al mercato non esiste nessuna soluzione che potrebbe essere collegarsi e utilizzare le caratteristiche del servizio.

Il prodotto sarà sviluppato in modo da poterlo utilizzare tramite il browser e ogni servizio avrà una sua GUI che lavora in parallelo.

Gli utenti di questo prodotto saranno utenti singoli che useranno il programma per scopo personale ma si può utilizzare anche per lo scopo di piccole aziende con un paio di server.

L’interfaccia utente sarà utilizzata da utenti con conoscenze minime quindi dovrà eseguire tutti i controlli sui dati inseriti dall’utente.

## Analisi e specifica dei requisiti

I requisiti per questo progetto sono stati definiti dal docente responsabile Geo Petrini.

L’interfaccia utente dovrà avere un aspetto moderno ispirandosi al material design, un layout responsive e dovrà essere semplice e intuitiva per l’utilizzatore. Tramite l’interfaccia utente si dovranno inoltre poter configurare le impostazioni del servizio, quest’ultima parte è anche da aggiungere al servizio.

Il progetto deve essere strutturato in modo da permettere delle future espansioni di funzionalità.

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | GUI moderna |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’interfaccia utente deve essere moderna e ispirata al material design |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Layout responsive |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’interfaccia utente si deve adeguare a ogni grandezza di schermo del utente per permettere sempre una visuale utilizzabile. |
| **Sotto requisiti** | |
| **01** | Visuale desktop |
| **02** | Visuale tablet |
| **03** | Visuale mobile |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | GUI chiara e semplice |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’interfaccia utente deve essere chiara e semplice |

|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | Lo stato delle scansioni |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Nella GUI deve essere possibile visualizzare lo stato e una stima del tempo. |
| **Sotto requisiti** | |
| **01** | Modificare la parte del servizio per permettere di ricavare lo stato della scansione |
| **02** | Rappresentare lo stato con una barra di caricamento |
| **03** | Implementare una stima del tempo necessario al completamento della scansione |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Panello per la gestione del servizio |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Nella GUI ci deve essere la possibilità di gestire completamente il servizio |

|  |  |
| --- | --- |
| **ID: REQ-06** | |
| **Nome** | Architettura modulare |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il servizio deve essere sviluppato in modo modulare per permettere dei sviluppi e espansioni future. |

|  |  |
| --- | --- |
| **ID: REQ-07** | |
| **Nome** | Avvio semplice della GUI |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | La interfaccia utente deve essere semplice da avviare/eseguire. |

## Use case

Il prodotto avrà diversi casi d’uso, essi sono rappresentati con il seguente diagramma

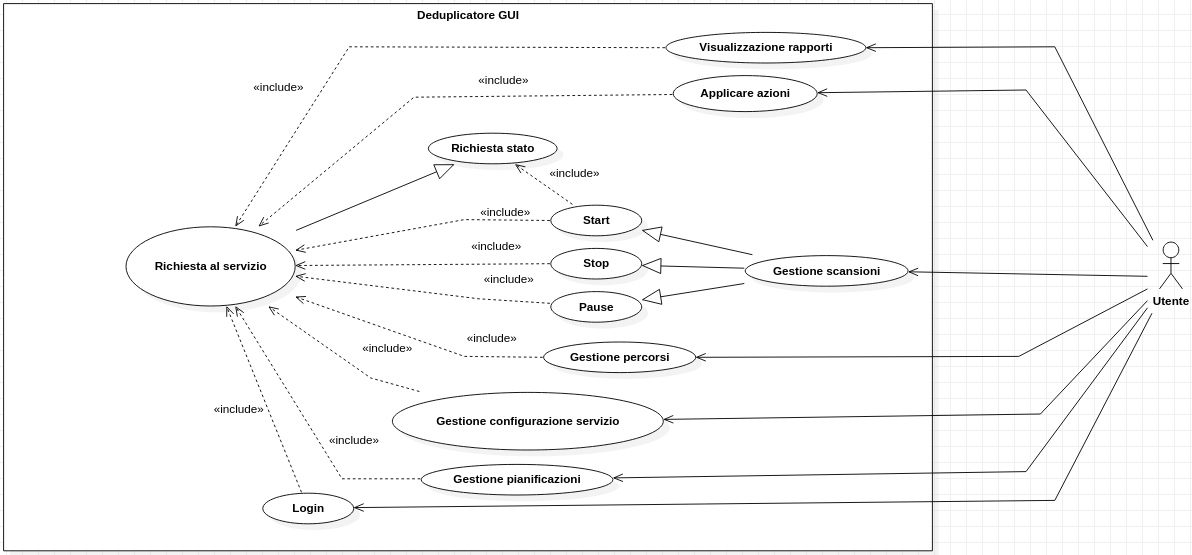


Figura 1 Diagramma dei casi d'uso

## Pianificazione

Prima di stabilire una pianificazione bisogna avere almeno una vaga idea del modello di sviluppo che si intende adottare. In questa sezione bisognerà inserire il modello concettuale di sviluppo che si seguirà durante il progetto. Gli elementi di riferimento per una buona pianificazione derivano da una scomposizione top-down della problematica del progetto.

La pianificazione può essere rappresentata mediante un diagramma di Gantt:

|  |
| --- |
| gantt  Figura 2: Esempio di diagramma di Gantt. |

Se si usano altri metodi di pianificazione (es scrum), dovranno apparire in questo capitolo.

## Analisi dei mezzi

Per la creazione di questo progetto ho a disposizione tutti i programmi che sono messi a disposizione dalla scuola e 1 accesso presso l'hosting interno nel caso che sia necessario caricare il progetto sull’FTP della scuola.

### Software

Vaadin 14

VSCode 1.41.1

MySQL 8.0

Java 11

Apache Maven 3.6.1

Postman 7.8.0

TODO:mettere tool e versioni complete

### Hardware

Per lo sviluppo verrà utilizzato il mio portatile personale che ha le seguenti specifiche:

HP Pavilion 15-0800nz

CPU: i7-8550U

RAM: 16 GB DDR4

OS: Pop\_OS! 19.10 / Kernel: 5.3.0-7625-generic

# 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 Gannt 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
* …