Documentazione

[1 Introduzione 2](#_Toc62724879)

[1.1 Informazioni sul progetto 2](#_Toc62724880)

[1.2 Abstract 2](#_Toc62724881)

[1.3 Scopo 2](#_Toc62724882)

[2 Analisi 3](#_Toc62724883)

[2.1 Analisi del dominio 3](#_Toc62724884)

[2.2 Analisi e specifica dei requisiti 3](#_Toc62724885)

[2.3 Use case 6](#_Toc62724886)

[2.4 Pianificazione 7](#_Toc62724887)

[2.5 Analisi dei mezzi 7](#_Toc62724888)

[2.5.1 Software 7](#_Toc62724889)

[2.5.2 Hardware 7](#_Toc62724890)

[3 Progettazione 8](#_Toc62724891)

[3.1 Design dell’architettura del sistema 8](#_Toc62724892)

[3.2 Design dei dati e database 8](#_Toc62724893)

[3.3 Design delle interfacce 8](#_Toc62724894)

[3.4 Design procedurale 8](#_Toc62724895)

[4 Implementazione 9](#_Toc62724896)

[5 Test 9](#_Toc62724897)

[5.1 Protocollo di test 9](#_Toc62724898)

[5.2 Risultati test 9](#_Toc62724899)

[5.3 Mancanze/limitazioni conosciute 10](#_Toc62724900)

[6 Consuntivo 10](#_Toc62724901)

[7 Conclusioni 10](#_Toc62724902)

[7.1 Sviluppi futuri 10](#_Toc62724903)

[7.2 Considerazioni personali 10](#_Toc62724904)

[8 Bibliografia 10](#_Toc62724905)

[8.1 Bibliografia per articoli di riviste: 10](#_Toc62724906)

[8.2 Bibliografia per libri 10](#_Toc62724907)

[8.3 Sitografia 10](#_Toc62724908)

[9 Allegati 10](#_Toc62724909)

# Introduzione

## Informazioni sul progetto

|  |  |
| --- | --- |
| **Responsabile progetto** | Geo Petrini |
| **Autori** | Thaisa De Torre, Viktorija Tilevska |
| **Inizio** | 14.01.2021 |
| **Consegna** | 20.05.2021 |

## Abstract

E’ una breve e accurata rappresentazione dei contenuti di un documento, senza notazioni critiche o valutazioni. Lo scopo di un abstract efficace dovrebbe essere quello di far conoscere all’utente il contenuto di base di un documento e metterlo nella condizione di decidere se risponde ai suoi interessi e se è opportuno il ricorso al documento originale.

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?

Esempio di abstract:

*As the size and complexity of today’s most modern computer chips increase, new techniques must be developed to effectively design and create Very Large Scale Integration chips quickly. For this project, a new type of hardware compiler is created. This hardware compiler will read a C++ program, and physically design a suitable microprocessor intended for running that specific program. With this new and powerful compiler, it is possible to design anything from a small adder, to a microprocessor with millions of transistors. Designing new computer chips, such as the Pentium 4, can require dozens of engineers and months of time. With the help of this compiler, a single person could design such a large-scale microprocessor in just weeks.*

## Scopo

Lo scopo del progetto (scopi didattici/scopi operativi). Dovrebbe descrivere il mandato, ma non vanno ricopiate le informazioni del quaderno dei compiti (che va invece allegato).

# Analisi

## Analisi del dominio

Questo capitolo dovrebbe descrivere il contesto in cui il prodotto verrà utilizzato, da questa analisi dovrebbero scaturire le risposte a quesiti quali ad esempio:

* Background/Situazione iniziale
* Quale è e come è organizzato il contesto in cui il prodotto dovrà funzionare?
* Come viene risolto attualmente il problema? Esiste già un prodotto simile?
* Chi sono gli utenti? Che bisogni hanno? Come e dove lavorano?
* Che competenze/conoscenze/cultura posseggono gli utenti in relazione con il problema?
* Esistono convenzioni/standard applicati nel dominio?
* Che conoscenze teoriche bisogna avere/acquisire per poter operare efficacemente nel dominio?
* …

## Analisi e specifica dei requisiti

Il progettista, dopo aver ricevuto il mandato, in collaborazione con il committente redige una lista di requisiti. Durante questi incontri, tramite interviste (da inserire nei diari), il progettista deve cercare di rispondere alle seguenti domande:

* Quali sono i bisogni del committente?
* Quali funzioni deve svolgere il prodotto?
* Come devono essere implementate?
* L’utente, come vorrebbe/dovrebbe interagire con il prodotto?
* Come verrà utilizzato il prodotto?
* Che tipo di interfaccia si immagina?
* Che prestazioni minime deve fornire il prodotto?
* Che grado di sicurezza deve avere il prodotto?
* …é

In base alla lista dei requisiti e all’analisi degli stessi, il progettista redige una *specifica dei requisiti* in cui elenca e descrive in modo dettagliato quali sono le funzionalità che il prodotto fornirà. La specifica dovrebbe essere abbastanza dettagliata da poter essere utilizzata come base per lo sviluppo, ma non troppo; ad esempio non dovrebbe contenere dettagli di implementazione, o definizioni dettagliate dell’interfaccia grafica a meno che questi non siano considerati cruciali. Non si deve scordare che i requisiti non rappresentano delle attività bensì delle caratteristiche che il prodotto dovrà possedere.

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Acquisizione di immagini in formato PNG o JPG |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Rilevamento del testo dall’immagine con un algoritmo OCR |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Output del testo come TXT |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Non è necessario formattarlo |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Dev’essere bilingue |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Lingue da interpretare: Italiano e Inglese |
| **D: REQ-005** | |
| **Nome** | Upload file multipli con output singolo |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’elaborato di più file di input viene ritornato come output singolo.  Es: file1.png 🡪 file1.txt; file1.png, file2.jpg 🡪 file.txt;  anche con mask \*.\* 🡪 file.txt |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Statistiche scansioni |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Per ogni esecuzione dev’essere prodotta una statistica con: la quantità di parole rilevate, la percentuale di affidabilità dei rilevamenti e il tempo di elaborazione |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Visualizzazione dati statistici tramite parametro opzionale |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | I dati statistici vengono visualizzati solamente se abilitati tramite parametro |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | Guida utilizzo |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Visualizzare una guida di utilizzo tramite parametro (--help). La guida viene visualizzata automaticamente in caso di parametri assenti |

|  |  |
| --- | --- |
| **ID: REQ-009** | |
| **Nome** | Utilità (applicazione) |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il lavoro deve poter essere utilizzato in produzione nell’azienda |

|  |  |
| --- | --- |
| **ID: REQ-010** | |
| **Nome** | Gestione degli errori |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | eventuali errori sono identificati e gestiti tramite i mezzi adeguati |

|  |  |
| --- | --- |
| **ID: REQ-011** | |
| **Nome** | Registro eventi/Logging |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L'elaborazione dei dati corretta o errata viene registrata in un file di registro. Al fine di permettere un’interpretazione corretta del registro, i dati pertinenti devono essere memorizzati nel formato appropriato e raggruppati in modo utile. |

|  |  |
| --- | --- |
| **ID: REQ-012** | |
| **Nome** | Tutte le dipendenze del software devono essere incluse |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Eventuali librerie esterne devono essere incluse e facilmente installabili |

|  |  |
| --- | --- |
| **ID: REQ-013** | |
| **Nome** | Gestione delle eccezioni (batch) |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il programma deve essere in grado di funzionare senza supervisioni e di reagire alle potenziali situazioni secondo dei processi predefiniti. In caso d’errore, il programma effettua le azioni definite conformemente alla specifica, scrive i dati nel file di log in modo da permettere un’analisi della situazione che ha portato all’interruzione. |

|  |  |
| --- | --- |
| **ID: REQ-014** | |
| **Nome** | Attendibilità dei dati inseriti dall’utente |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | I campi di immissione sono contrassegnati in modo chiaro e vengono rivisti. Plausibilità: In caso di errore di digitazione, l'utente viene aiutato da indicazioni concrete e il campo corrispondente viene attivato/ evidenziato. |

|  |  |
| --- | --- |
| **ID: REQ-015** | |
| **Nome** | Organizzazione del programma |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il programma è stato strutturato in maniera intelligente senza procedure e funzioni e sotto-procedure/funzioni ridondanti. |

|  |  |
| --- | --- |
| **ID: REQ-016** | |
| **Nome** | Utilizzo di diagrammi di flusso |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | I diagrammi di flusso rappresentano lo svolgimento completo del programma. È rappresentato in modo chiaro, facilmente leggibile e contiene anche dei commenti pertinenti. |

## Use case

I casi d’uso rappresentano l’interazione tra i vari attori e le funzionalità del prodotto.

## 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 1: Esempio di diagramma di Gantt. |

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

## Analisi dei mezzi

Elencare e descrivere i mezzi disponibili per la realizzazione del progetto. Ricordarsi di sempre descrivere nel dettaglio le versioni e il modello di riferimento.

### Software

SDK, librerie, tools utilizzati per la realizzazione del progetto e eventuali dipendenze.

### Hardware

Su quale piattaforma dovrà essere eseguito il prodotto? Che hardware particolare è coinvolto nel progetto? Che particolarità e limitazioni presenta? Che hw sarà disponibile durante lo sviluppo?

# 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 funzionali

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-001 | **Nome:** | Acquisizione di immagini in formato PNG |
| **Descrizione:** | Passare all’OCR un’immagine in formato PNG da leggere. | | |
| **Prerequisiti:** | Avere installato l’OCR e avere un’immagine di un testo in formato PNG. | | |
| **Procedura:** | 1. Aprire la linea di commando 2. Passare il percorso dell’immagine PNG all’ocr   >ocr png\_img.png | | |
| **Risultati attesi:** | L’OCR dovrebbe accettare l’immagine PNG. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-001 | **Nome:** | Acquisizione di immagini in formato JPG |
| **Descrizione:** | Passare all’OCR un’immagine in formato JPG da leggere. | | |
| **Prerequisiti:** | Avere installato l’OCR e avere un’immagine di un testo in formato JPG. | | |
| **Procedura:** | 1. Aprire la linea di commando 2. Passare il percorso dell’immagine JPG all’ocr   >ocr jpg\_img\_eng.jpg | | |
| **Risultati attesi:** | L’OCR dovrebbe accettare l’immagine JPG. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-001 | **Nome:** | Acquisizione di immagini in un formato diverso da JPG o PNG |
| **Descrizione:** | Passare all’OCR un’immagine in un formato diverso da JPG o PNG da leggere. | | |
| **Prerequisiti:** | Avere installato l’OCR e avere un’immagine di un testo in un formato diverso da JPG o PNG. | | |
| **Procedura:** | 1. Aprire la linea di commando 2. Passare il percorso dell’immagine all’ocr   >ocr img\_eng.bmp | | |
| **Risultati attesi:** | Messaggio di errore per il formato non adeguato. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-002 e REQ-003 | **Nome:** | Rilevamento del testo dall’immagine con un algoritmo OCR con output del testo come TXT |
| **Descrizione:** | Passare all’OCR un’immagine in formato JPG o PNG da leggere e ricevere come output il testo dell’immagine come TXT. | | |
| **Prerequisiti:** | Avere installato l’OCR e avere un’immagine di un testo in formato JPG o PNG. | | |
| **Procedura:** | 1. Aprire la linea di commando 2. Passare il percorso dell’immagine all’ocr   >ocr img.jpg   1. Andare nella cartella di destinazione e controllare che il file di output esiste | | |
| **Risultati attesi:** | output file in formato txt con dentro il testo rilevato dall’immagine. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  REQ-004 | **Nome:** | Rilevamento del testo in inglese |
| **Descrizione:** | La lingua del testo dell’immagine passata può essere in inglese. | | |
| **Prerequisiti:** | Avere un’immagine con il testo in inglese. | | |
| **Procedura:** | 1. Usare l’ocr impostando il parametro per la lingua italiana   >ocr img\_eng.png –l eng | | |
| **Risultati attesi:** | La lingua è accettata | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-006  REQ-004 | **Nome:** | Rilevamento del testo in italiano |
| **Descrizione:** | La lingua del testo dell’immagine passata può essere in italiano. | | |
| **Prerequisiti:** | Avere un’immagine con il testo in italiano. | | |
| **Procedura:** | 1. Usare l’ocr impostando il parametro per la lingua italiana   >ocr img\_ita.png –l ita | | |
| **Risultati attesi:** | La lingua è accettata | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-007  REQ-004 | **Nome:** | Rilevamento del testo in una lingua non supportata |
| **Descrizione:** | La lingua del testo dell’immagine passata deve essere diversa da italiano o inglese. | | |
| **Prerequisiti:** | Avere un’immagine con il testo in una lingua diversa da italiano o inglese. | | |
| **Procedura:** | 1. Usare l’ocr impostando la lingua   >ocr img.png –l fra | | |
| **Risultati attesi:** | Messaggio di errore per la lingua non supportata | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-008  REQ-005 | **Nome:** | Inserire un file di input singolo e ricevere un file di output solo |
| **Descrizione:** | Fare il rilevamento di una sola immagine e scrivere l’output in un singolo file txt. | | |
| **Prerequisiti:** | Avere un immagine in formato png o jpg. | | |
| **Procedura:** | Utilizzare l’OCR | | |
| **Risultati attesi:** | Si ottiene un singolo file di output. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-009  REQ-005 | **Nome:** | Inserire file multipli come input e ricevere un file di output singolo. |
| **Descrizione:** | Fare il rilevamento di più immagini e scrivere l’output in un singolo file txt. | | |
| **Prerequisiti:** | Avere più immagini in formato png o jpg. | | |
| **Procedura:** | Utilizzare l’OCR passando come sorgente 2 immagini in formato png o jpg separate da una virgola  >ocr img1.png, img2.jpg | | |
| **Risultati attesi:** | Si ottiene un singolo file di output con il contenuto delle due immagini scannerizzate. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-010  REQ-005 | **Nome:** | Inserire un file mask come input e ricevere un file di output solo. |
| **Descrizione:** | Fare il rilevamento di tutte le immagini in formato accettato e scriverne l’output in un singolo file txt. | | |
| **Prerequisiti:** | Avere più immagini in formato png o jpg. | | |
| **Procedura:** | Utilizzare l’OCR passando come sorgente file mask  >ocr \*.\* | | |
| **Risultati attesi:** | Si ottiene un singolo file di output con il contenuto di tutte le immagini trovate in formato accettabile. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-011  REQ-006 | **Nome:** | Statistiche scansioni |
| **Descrizione:** | Controllare che dopo ogni esecuzione dell’OCR vengano calcolate le statistiche. | | |
| **Prerequisiti:** | Avere accesso ai file di log | | |
| **Procedura:** | 1. Utilizzare l’OCR   >ocr lorem\_ipsum.png   1. Controllare nei file di log le statistiche | | |
| **Risultati attesi:** | Nei file di log le statistiche scritte devono coincidere con **<…>** | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-012  REQ-007 | **Nome:** | Dati statistiche opzionali |
| **Descrizione:** | Le statistiche vengono stampate a terminale solamente se specificato da parametro –stats | | |
| **Prerequisiti:** | Utilizzare l’OCR | | |
| **Procedura:** | 1. Utilizzare l’OCR   >ocr lorem\_ipsum.png --stats | | |
| **Risultati attesi:** | Alla fine dell’esecuzione a terminale dev’essere stampata la statistica del file | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-013  REQ-007 | **Nome:** | Dati statistiche opzionali senza parametro |
| **Descrizione:** | Le statistiche non devono venire stampate a terminale | | |
| **Prerequisiti:** | Utilizzare l’OCR | | |
| **Procedura:** | 1. Utilizzare l’OCR   >ocr lorem\_ipsum.png | | |
| **Risultati attesi:** | Alla fine dell’esecuzione a terminale non viene stampato nulla | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-014  REQ-008 | **Nome:** | Guida utilizzo tramite parametro |
| **Descrizione:** | Con il parametro –h/-help viene stampata a terminale la guida di utilizzo del comando. | | |
| **Prerequisiti:** | Utilizzare l’OCR | | |
| **Procedura:** | 1. Utilizzare il comando OCR con il parametro –h o -help   >ocr –help | | |
| **Risultati attesi:** | Viene stampata a terminale la guida | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-015  REQ-008 | **Nome:** | Guida utilizzo per parametri mancanti |
| **Descrizione:** | Se non si inserisce come minimo la sorgente del file l’OCR solleva un errore per parametri mancanti e stampa a terminale la guida per mostrare all’utente il corretto utilizzo del comando. | | |
| **Prerequisiti:** | Utilizzare l’OCR | | |
| **Procedura:** | 1. Utilizzare il comando OCR senza parametri   >ocr | | |
| **Risultati attesi:** | Viene stampata a terminale un errore per i parametri mancanti e la guida per il corretto utilizzo | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-016  REQ-012 | **Nome:** | Dipendenze SW incluse |
| **Descrizione:** | Tutte le dipendenze necessarie devono essere incluse e installate nel software finale. | | |
| **Prerequisiti:** | Avere l’eseguibile dell’OCR e poterlo eseguire | | |
| **Procedura:** | 1. Eseguire il .exe dell’OCR 2. Controllare che tutte le dipendenze siano installate | | |
| **Risultati attesi:** | Tutte le dipendenze devono essere incluse e installate con l’exe | | |

### Test non funzionali

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  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. | | |

## 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.

**Legenda**: test riuscito test fallito

|  |  |  |
| --- | --- | --- |
| ID Test Case | Nome | Risultato |
| TC-001 |  |  |

## 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
* …