Gestione Chiavi Armadietti (GCA)

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

In questo capitolo sono raccolte le informazioni relative al progetto:

* Allievo: Valerio Ratti
* Docente responsabile: Michel Tinelli
* Perito 1: Michel Tinelli
* Data di inizio: 04.02.2025
* Data di fine: 04.04.2025
* Luogo di sviluppo: Scuola Arti e Mestieri di Trevano
* Sezione scolastica: Informatica 4° anno

## Abstract

*The GCA (Locker Key Management) project aimed to streamline and enhance the management of locker keys within our institution. By developing a new web-based application, we created a centralized database for tracking keys, enabling efficient search, assignment, and inventory processes. Key features include advanced search capabilities, digital signatures for enhanced security, and a user-friendly interface. The system has significantly improved key management operations, reducing administrative overhead and improving overall user satisfaction.*

## 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).

Lo scopo del progetto è quello di creare un sistema centralizzato per la gestione delle chiavi degli armadietti, si tratta di un miglioramento da un progetto già esistente, che ora gestisce solo le chiavi di docenti, custodi, sistemisti ed esterni.

In aggiunta va migliorata l’usabilità della parte già esistente, secondo alcune specifiche date dagli utilizzatori.

Infine il progetto a me serve per prepararmi e consolidare le mie skill di programmazione per l’esame finale della professione (LPI)

# 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?
* …

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** | Utente Admin |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Modifiche da apportare agli utenti admin |
| **Sotto requisiti** | |
| **001** | Gestione degli utenti da parte degli admin |
| **002** | Possibilità di importare un file Excel per creare più utenti contemporaneamente |
| **003** | Aggiungere campo email per gli utenti |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Consegna chiavi |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Modifiche da apportare alla consegna delle chiavi |
| **Sotto requisiti** | |
| **001** | Chiavi preferite sempre visibili nella UI |
| **002** | Live search per le chiavi |
| **003** | Possibilità di aggiungere una chiave nella stessa UI |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Archivio |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Aggiunta sezione archivio per gli utenti che non hanno chiavi |
| **Sotto requisiti** | |
| **001** | Creare archivio utenti per quelli che non hanno più chiavi |
| **002** | Al posto del bottone cancella, bottone archivio |
| **003** | Solo nell’archivio è possibile cancellare un utente |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Gestione delle chiavi degli armadietti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Le chiavi possono essere assegnate a singoli utenti o essere generiche |
| **Sotto requisiti** | |
| **001** | Possibilità di assegnare una chiave a un utente specifico |
| **002** | Supporto per chiavi generiche utilizzabili da più utenti |
| **003** | Visualizzazione dello storico delle assegnazioni delle chiavi |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Inventario degli armadietti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’inventario deve essere aggiornabile e visualizzabile facilmente |
| **Sotto requisiti** | |
| **001** | Elenco completo degli armadietti disponibili e assegnati |
| **002** | Possibilità di filtrare e cercare armadietti per stato (libero, occupato) |
| **003** | Opzione per aggiornare lo stato di un armadietto |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Miglioramento dell’usabilità e gestione assegnazioni |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | L’interfaccia deve essere intuitiva e semplificare l’assegnazione delle chiavi |
| **Sotto requisiti** | |
| **001** | Implementazione di un’interfaccia utente più chiara con notifiche sugli stati delle chiavi |
| **002** | Possibilità di assegnare più chiavi contemporaneamente |
| **003** | Opzione di riassegnazione veloce delle chiavi scadute o restituite |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Gestione utenti e permessi |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il sistema deve permettere la gestione di utenti con diversi ruoli e permessi |
| **Sotto requisiti** | |
| **001** | Creazione di ruoli amministratore, docente, studente, ecc |
| **002** | Definizione di permessi specifici per ogni ruolo |
| **003** | Accesso con autenticazione e gestione password |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | Gestione sezioni |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il sistema deve essere in grado di gestire le sezioni dei vari docenti |
| **Sotto requisiti** | |
| **001** | Creazione delle sezioni |

**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.

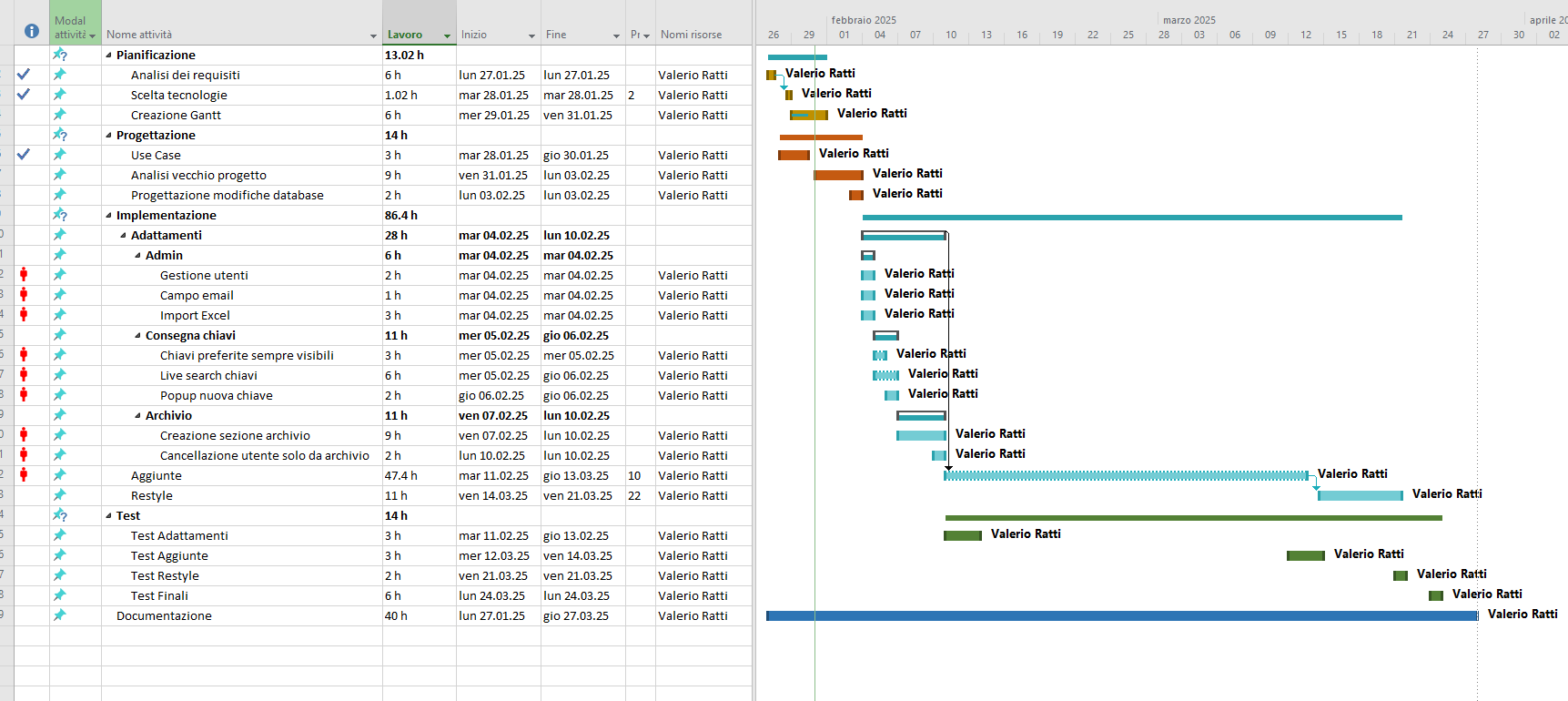
**Sotto requisiti**: elementi che compongono il requisito.

## Use case

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:

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

Se si usano altri metodi di pianificazione (p.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

PhpStorm 2024.1.1

DataGrip 2024.3.3

Chrome

Visual Studio Code

MySQL Workbench

Vagrant

Microsoft Project

Postman?

### Hardware

PC fornito dalla scuola

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.

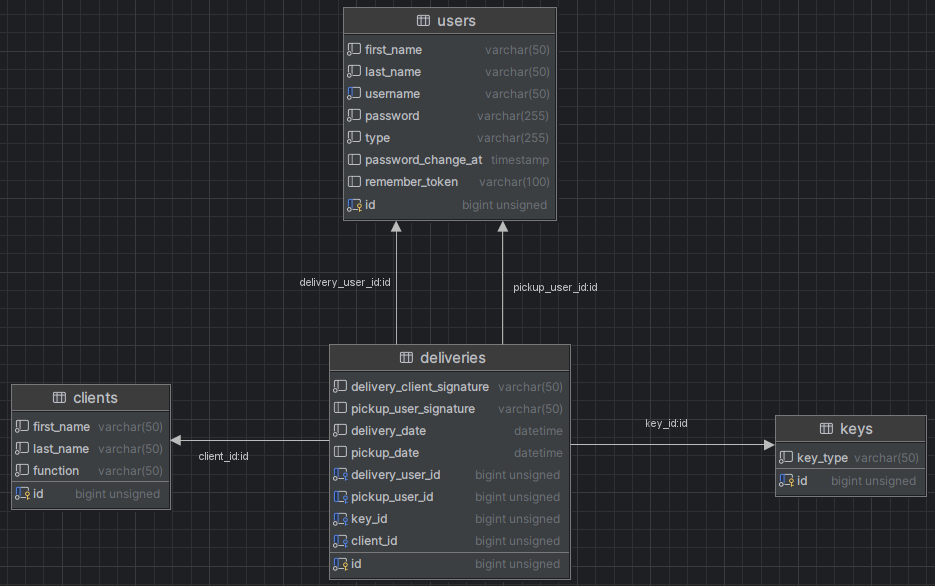
## Design dell’architettura del sistema

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

Design iniziale



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

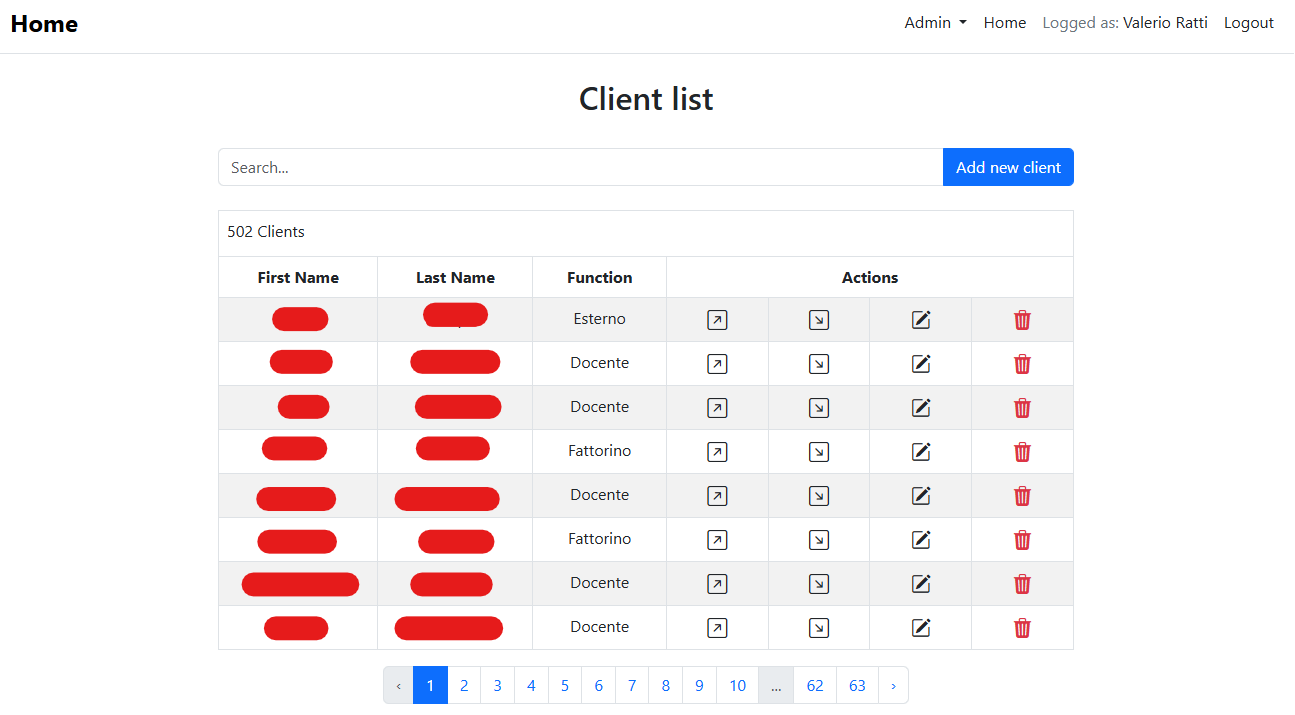
Schema E-R, schema logico e descrizione.

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

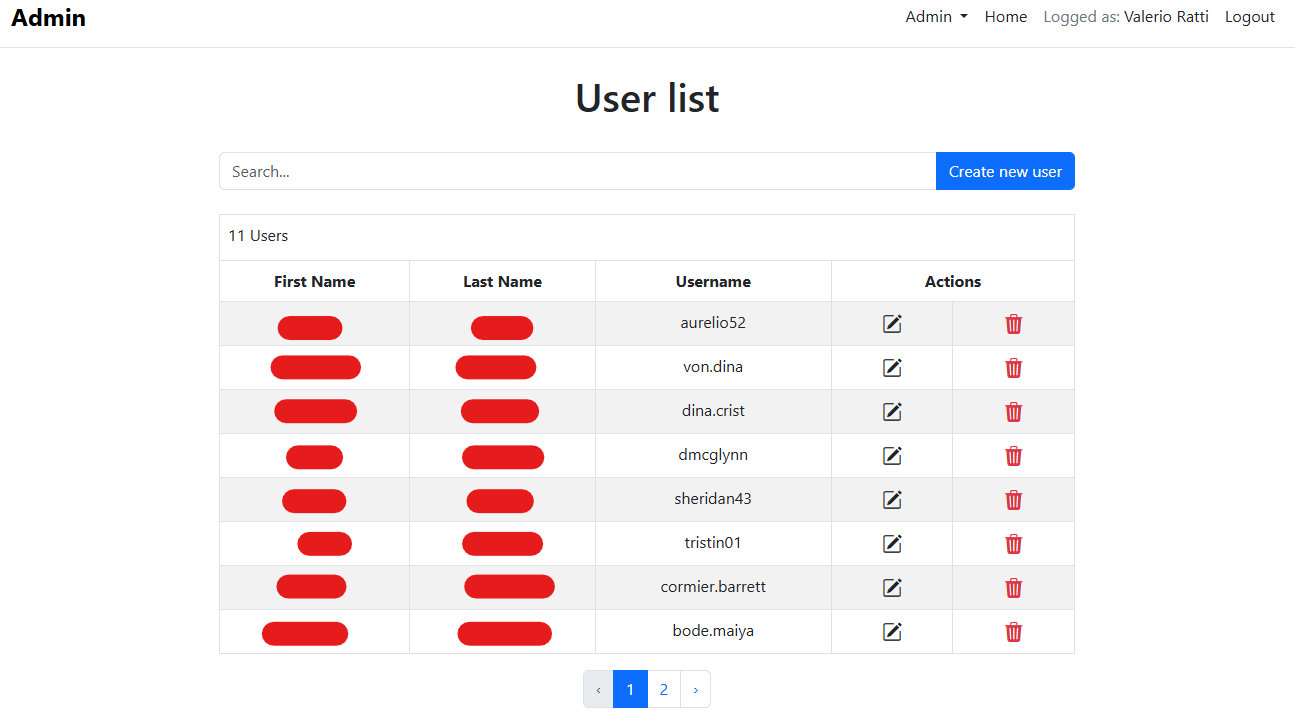
## Design delle interfacce

### Design iniziale

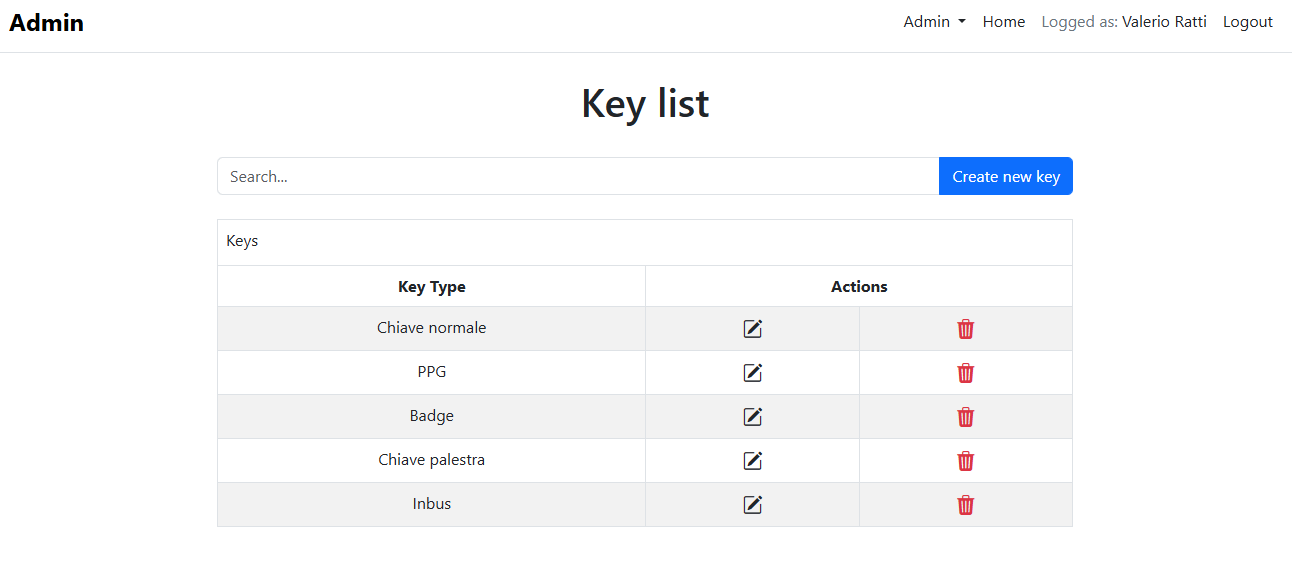
**Client list**

****

**User list**

****

**Key list**

****

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
* …