WatchTower

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 5

2.4 Pianificazione 5

2.5 Analisi dei mezzi 6

2.5.1 Software 6

2.5.2 Hardware 6

3 Progettazione 6

3.1 Design dell’architettura del sistema 6

3.2 Design dei dati e database 6

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: WatchTower
* Allievi: Tom Schillerwein I4BC
* Docente responsabile: Pascal Poncini
* Scuola: Arti e Mestieri Trevano, sezione Informatica
* Data di inizio e fine: 03.02.2025 – 03.04.2025
* Data di presentazione: 07 – 11.04.2025

## Abstract

The aim of this project is to implement a physical monitoring system for the server room of the black network of the CPT in Trevano. This monitoring includes temperature, humidity, access control and others. In addition, it is necessary to develop a web application that allows the visualisation of the data collected by the sensors and a system of notifications and alerts for the network system administrators.

## Scopo

Lo scopo di questo progetto è implementare un sistema di monitoraggio fisico della server room della rete nera della CPT di Trevano. Questo monitoraggio include la temperatura, l’umidità, il controllo degli accessi e altro. Inoltre è necessario sviluppare un’applicazione web che permette la visualizzazione dei dati raccolti dai sensori e un sistema di notifiche e allerte per i sistemisti della rete.

# Analisi

## Analisi del dominio

Questo progetto permette di semplificare il monitoraggio fisico della sala server tramite sensori che trasmettono le informazioni raccolte ad una dashboard a cui i sistemisti hanno accesso. Se i valori rilevati superano un certo livello definito, i sistemisti ricevono un avviso. Al momento i server nella sala vengono monitorati, ma non la sala fisica. Viene unicamente richiesto aprire la porta con un badge con i permessi necessari. Esistono già soluzioni simili, come Monnit, che però sono costose e meno specifiche per la nostra situazione.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Installazione sensori |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Montare i sensori, cavi e controller nella sala server. |
| **Sotto requisiti** | |
| **001** | Avere tutti i componenti hardware necessari per il progetto. |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Raccolta dati dai sensori |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Necessita che i sensori siano correttamente configurati e comunicano con il controller. |
| **Sotto requisiti** | |
| **001** | Il sistema deve raccogliere i dati di temperatura e umidità dal sensore ENV III. |
| **002** | Deve registrare gli accessi tramite RFID. |
| **003** | Deve rilevare movimento attraverso il sensore PIR. |
| **004** | Deve acquisire input dalla mini tastiera per capire il tipo di accesso. |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Creazione database |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessitano i permessi di root |
| **Sotto requisiti** | |
| **001** | Il database deve supportare la memorizzazione dei dati rilevati dai sensori. |
| **002** | Definire un tempo di mantenimento dei dati. |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Creazione dashboard |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Sviluppare un'interfaccia utente chiara e responsive. |
| **Sotto requisiti** | |
| **001** | Si necessita una maschera di login |
| **002** | Bisogna visualizzare i dati dei sensori in tempo reale. |
| **003** | Grafici con storico dei dati. |
| **004** | Possibilità di configurare notifiche e allerte per valori specifici. |
| **005** | Deve includere un log degli eventi principali. |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Notifiche e allerte |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessitano i permessi di root / Dipende dal requisito REQ-001 (Creazione DB) |
| **Sotto requisiti** | |
| **001** | Invio email in base a eventi configurati dai sistemisti. |

## Use case

|  |
| --- |
| 1 Use Case |

## Pianificazione

|  |
| --- |
| 2 Diagramma di Gantt preventivo |

## Analisi dei mezzi

Per questo progetto utilizzo un PC scolastico per lo sviluppo, un controller M5Stack Core2 e X sensori.

### Software

* Visual Studio Code 1.93, come editor di testo
* Google Chrome 131.0, come browser per visualizzare il sito durante lo sviluppo
* Mozilla Firefox 126.0, per testare la visualizzazione del sito
* Onlinegantt.com, per il Gantt
* Microsoft Visio, per fare lo Use Case
* Microsoft Word e Obsidian, per redigere la documentazione, il QdC e il diario
* GitHub, come Repository online

### Hardware

Computer utilizzato per le ricerche e lo sviluppo:

* Processore: Intel(R) Core(TM) i7-9700 CPU @ 3.00GHz
* RAM: 32GB
* Scheda Video: NVIDIA GeForce RTX 2060
* SSD: 512GB
* Monitor 1920x1080 60Hz
* Monitor 1600x900 60Hz

Componenti M5Stack:

* M5Stack Core2
* Sensori

# 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
* …