Piattaforma per la gestione della comunicazione casa / scuola per una scuola media

1 Introduzione 2

1.1 Informazioni sul progetto 2

1.2 Abstract 2

1.3 Scopo 2

2 Analisi 3

2.1 Analisi del dominio 3

2.2 Analisi e specifica dei requisiti 3

2.3 Use case 5

2.4 Pianificazione 6

2.5 Analisi dei mezzi 7

2.5.1 Software 7

2.5.2 Hardware 7

3 Progettazione 8

3.1 Design dell’architettura del sistema 8

3.2 Design dei dati e database 8

3.3 Design delle interfacce 8

3.4 Design procedurale 8

4 Implementazione 9

5 Test 10

5.1 Protocollo di test 10

5.2 Risultati test 10

5.3 Mancanze/limitazioni conosciute 10

6 Consuntivo 11

7 Conclusioni 12

7.1 Sviluppi futuri 12

7.2 Considerazioni personali 12

8 Bibliografia 12

8.1 Bibliografia per articoli di riviste: 12

8.2 Bibliografia per libri 12

8.3 Sitografia 12

9 Allegati 13

# Introduzione

## Informazioni sul progetto

* Candidato: Gabriele Alessi, gabriele.alessi@samtrevano.ch
* Superiore professionale: Luca Peduzzi, luca.peduzzi@edu.ti.ch
* Scuola d’Arti e Mestieri di Trevano, Sezione Informatica, Classe I4AA, Progetti individuali
* Numero di ore: 154
* Orario di lavoro:
  + Lunedì: 8.20 – 11.35, 15.00 – 16.30
  + Mercoledì, Giovedì, Venerdì: 13.15 – 16.30
* Data inizio: 23.01.2020
* Data fine: 06.04.2020
* Data presentazione: aprile 2020

## 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 consiste nello sviluppare un’applicazione web che permetta a docenti e genitori di gestire e consultare l’andamento scolastico degli allievi di una scuola media. Ciò viene fatto per facilitare la comunicazione casa/scuola e per permettere al docente di inserire delle note a un allievo e caricando le scansioni delle verifiche fatte in classe. In questo modo i genitori hanno la possibilità di esaminare in maniera semplice e veloce la situazione scolastica dei propri figli analizzando le loro schede con le eventuali note, il calendario con i compiti e le verifiche pianificate e visionare i test caricati sulla piattaforma.

Ovviamente anche gli stessi studenti avranno accesso al sistema, in modo da poter consultare principalmente il calendario.

# Analisi

## Analisi del dominio

I genitori e i docenti sono d’accordo sul fatto di fare qualcosa riguardo alla comunicazione casa/scuola visto che le informazioni che passano da dei studenti adolescenti non sono sempre così sicure e dirette.

Il prodotto deve funzionare nel contesto di una scuola media e per ora i genitori non hanno la possibilità di consultare l’andamento scolastico dei propri figli in maniera semplice, sicura e veloce. Dunque gli utenti principali di questo sistema sono docenti, genitori e studenti di una scuola media.

Per poter utilizzare il prodotto è necessario ovviamente un dispositivo con accesso a internet, dopodiché sarà possibile usufruire del servizio e sarà sviluppato in modo che sia intuitivo utilizzabile da chiunque.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| ID: REQ-001 | |
| Nome | Creare un applicativo web per la gestione della comunicazione casa/scuola |
| Priorità | 1 |
| Versione | 1.0 |
| Note | Questo è il requisito principale e i sotto requisiti rappresentano le varie funzionalità del prodotto richieste del committente. |
| Sotto requisiti | |
| 001 | Amministrazione (REQ-002) |
| 002 | Sistema di login (REQ-003) |
| 003 | Schede degli studenti (REQ-004) |
| 004 | Calendario (REQ-005) |
| 005 | Pagina del docente (REQ-006) |
| 006 | Caricamento verifica (REQ-007) |

|  |  |
| --- | --- |
| ID: REQ-002 | |
| Nome | Amministrazione |
| Priorità | 2 |
| Versione | 1.0 |
| Note | È presente un utente amministratore che può accedere al sito e ai dati in modo completo. |
| Sotto requisiti | |
| 001 | L’amministratore può creare, cancellare e modificare i vari campi direttamente da una pagina. |
| 002 | L’amministratore deve poter creare nuovi utenti specificando lo username (nome.cognome). |
| 003 | C’è una pagina di amministrazione dove l’admin può visualizzare e creare utenti, classi e materie. |

|  |  |
| --- | --- |
| ID: REQ-003 | |
| Nome | Sistema di login |
| Priorità | 1 |
| Versione | 1.0 |
| Note | Il sistema di login permette agli utenti di accedere ai rispettivi contenuti. |
| Sotto requisiti | |
| 001 | Quando l’admin inserisce un nuovo utente, questo riceve una mail con una password provvisoria e durante il login ne viene fatta inserire una nuova. |
| 002 | Quando si dimentica la password si può richiederne una nuova. |
| 003 | I tipi di utenti (docenti, genitori, studenti) possono accedere solo ai rispettivi contenuti. |

|  |  |
| --- | --- |
| ID: REQ-004 | |
| Nome | Schede degli studenti |
| Priorità | 1 |
| Versione | 1.0 |
| Note | Ogni studente è rappresentato da una scheda che mostra delle informazioni. |
| Sotto requisiti | |
| 001 | Il docente di classe deve poter aggiungere delle note sulle schede degli studenti e caricare le scansioni delle verifiche in PDF. |
| 002 | Dalla scheda devono essere visibili informazioni come nome, cognome, classe e da quest’ultima è possibile raggiungere il calendario. |

|  |  |
| --- | --- |
| ID: REQ-005 | |
| Nome | Calendario |
| Priorità | 1 |
| Versione | 1.0 |
| Note | Il calendario definisce le pianificazioni di compiti e test di una classe. |
| Sotto requisiti | |
| 001 | I docenti devono poter aggiungere e gestire le verifiche e le scadenze di una classe pianificate. |
| 002 | Gli studenti possono visualizzare il calendario della propria classe. |

|  |  |
| --- | --- |
| ID: REQ-006 | |
| Nome | Pagina del docente |
| Priorità | 1 |
| Versione | 1.0 |
| Note | Il calendario definisce le pianificazioni di compiti e test di una classe. |
| Sotto requisiti | |
| 001 | I docenti devono poter aggiungere e gestire le verifiche e le scadenze di una classe pianificate. |
| 002 | Gli studenti possono visualizzare il calendario della propria classe. |

|  |  |
| --- | --- |
| ID: REQ-007 | |
| Nome | Caricamento verifica |
| Priorità | 1 |
| Versione | 1.0 |
| Note | Il calendario definisce le pianificazioni di compiti e test di una classe. |
| Sotto requisiti | |
| 001 | I docenti devono poter aggiungere e gestire le verifiche e le scadenze di una classe pianificate. |
| 002 | Gli studenti possono visualizzare il calendario della propria classe. |

## Use case

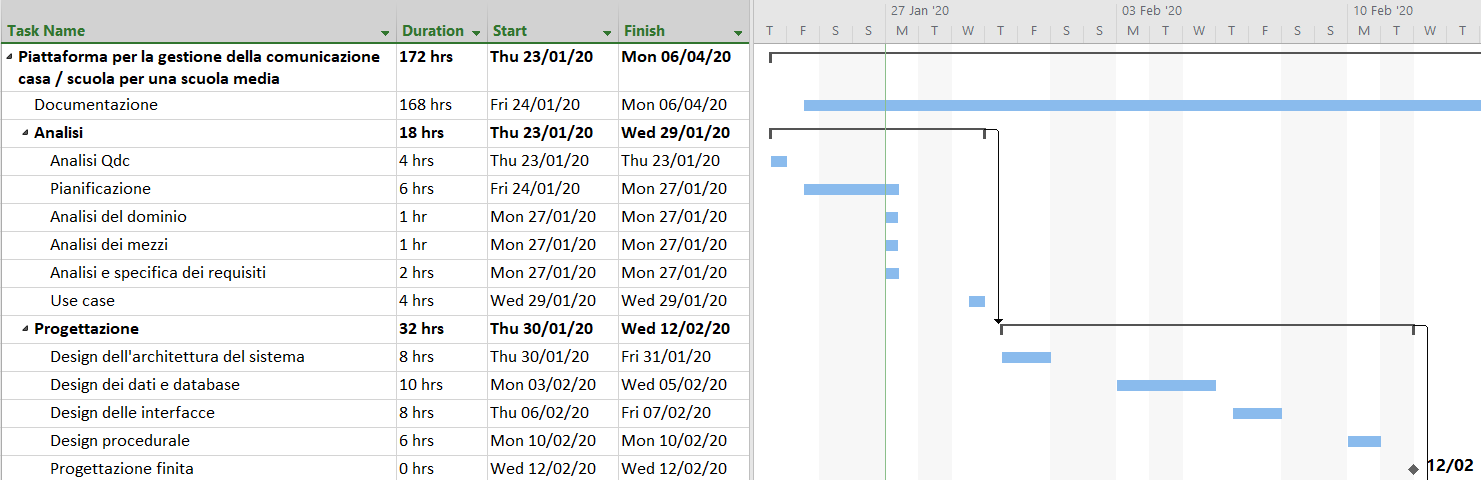
I casi d’uso rappresentano l’interazione tra i vari attori e le funzionalità del prodotto.

## Pianificazione

La pianificazione del progetto è stata effettuata mediante la realizzazione di un diagramma di Gantt.

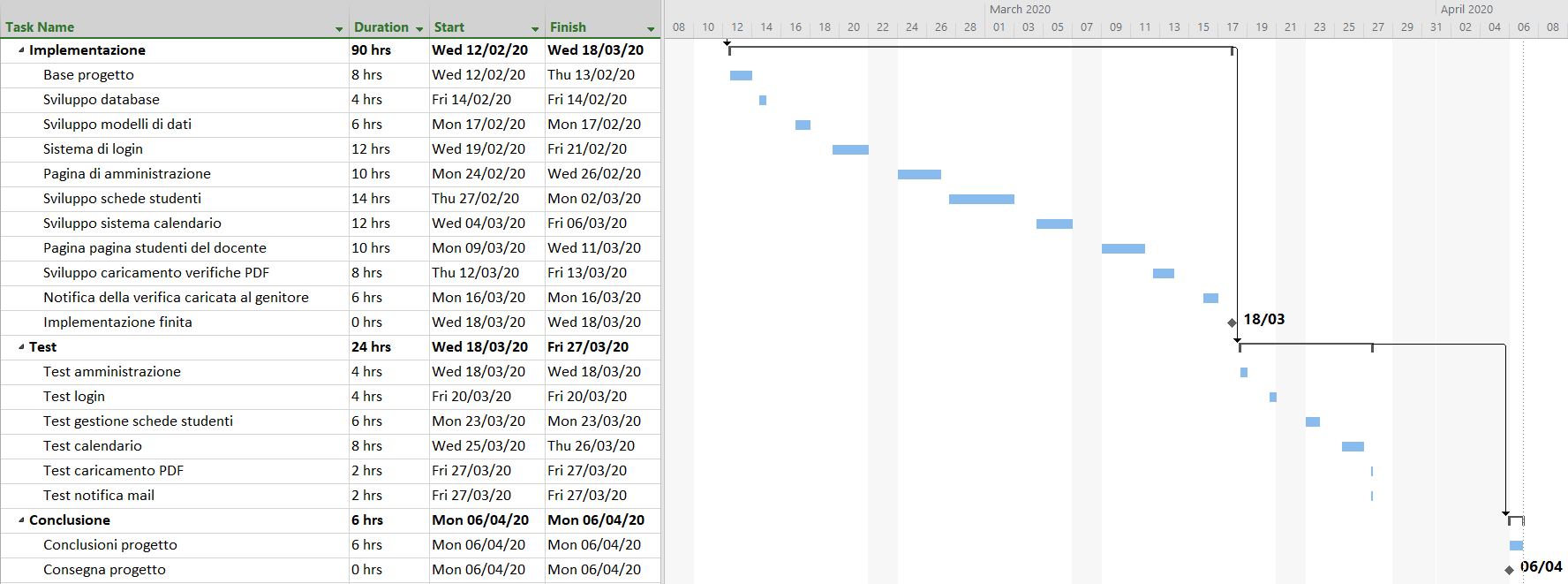
### Analisi e progettazione

La prima parte della pianificazione prende in considerazione l’analisi del progetto e successivamente la progettazione. Per quanto riguarda la documentazione si può dire che essa è inclusa nelle tempistiche di ogni attività in modo da non complicare il diagramma.



### Implementazione, test e conclusioni

La seconda parte del diagramma comprende la parte di implementazione del progetto e quindi i relativi test. Infine c’è uno spazio per le varie conclusioni generali del lavoro.



## Analisi dei mezzi

Qui vengono elencati i software usati e viene descritto il Personal Computer utilizzato per lo sviluppo del

progetto.

### Software

I software utilizzati per la realizzazione di questo progetto sono:

* Microsoft Word 2016   
  Realizzazione della documentazione.
* GitHub Desktop 2.2.4   
  Gestione del sistema di versioning del progetto (GitHub).
* Visual Studio Code 1.41   
  Realizzazione dei diari giornalieri e gestione dei vari documenti di testo.
* Microsoft Project 2016   
  Realizzazione del diagramma di Gantt.
* Microsoft Visio 2016   
  Realizzazione dei vari schemi e diagrammi.
* PhpStorm 2019.3   
  Sviluppo dell’applicazione web.
* XAMPP 7.4.1
  + Apache 2.4
  + MariaDB 10.4
  + PHP 7.4

Gestione Web Server locale e database.

### Hardware

Personal Computer:

* HP Envy Notebook
* Intel Core™ i7-6500U @ 2.50GHz
* 16GB RAM
* Intel HD Graphics 520
* Windows 10 Home 64bit, Versione 1903, Build 18362.592

# 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
* …