Esempio di documentazione

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 4

2 Analisi 5

2.1 Analisi del dominio 5

2.2 Analisi e specifica dei requisiti 5

2.3 Use case 11

2.4 Pianificazione 11

2.5 Analisi dei mezzi 14

2.5.1 Software 14

2.5.2 Hardware 14

3 Progettazione 14

3.1 Design dell’architettura del sistema 14

3.2 Design dei dati e database 14

3.3 Design delle interfacce 17

3.4 Design procedurale 17

4 Implementazione 18

5 Test 18

5.1 Protocollo di test 18

5.2 Risultati test 20

5.3 Mancanze/limitazioni conosciute 20

6 Consuntivo 20

7 Conclusioni 22

7.1 Sviluppi futuri 22

7.2 Considerazioni personali 22

8 Bibliografia 23

8.1 Bibliografia per articoli di riviste: 23

8.2 Bibliografia per libri 23

8.3 Sitografia 23

9 Glossario 24

10 Indice delle figure 24

11 Allegati 24

# Introduzione

## Informazioni sul progetto

In questo capitolo raccogliere le informazioni relative al progetto, ad esempio:

* Gabriel Gonçalves
* Scuola arti e mestieri Trevano, I3AA, M306
* 05.09.2025-

## Abstract

È 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.*

Questo progetto è un simulatore di investimenti realizzato come sito web in HTML, CSS e JavaScript. L’obiettivo è permettere all’utente di capire come funzionano i mercati senza usare soldi veri. Per farlo, il programma utilizza file CSV con dati storici di prezzi (Ask e Bid), che vengono mostrati in un grafico aggiornato progressivamente per simulare l’andamento in tempo reale. L’utente parte con un saldo iniziale e può comprare o vendere durante la simulazione.

## 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 realizzare un simulatore di investimenti che permetta di comprendere in modo pratico e intuitivo il funzionamento dei mercati finanziari. L’applicazione ha finalità didattiche: consente di osservare l’andamento dei prezzi attraverso dati storici e di esercitarsi con operazioni di acquisto e vendita senza rischiare denaro reale.

# 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?
* …

Online si trovano molte applicazioni per fare trading, ma spesso sono complicate e devi spendere soldi, con la mia applicazione si potrà fare pratica tramite simulazione utilizzando dati vecchi, l’interfaccia sarà intuitiva, ci saranno più mercati, gli utenti saranno i neofiti in questo ambito, che impareranno le basi gratuitamente.

## 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.

Stile 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-1 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Login/Sign in | | | | |
| **Note** | L’utente potrà registrarsi con un nuovo account o fare il login con un account già creato. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Il sistema dovrà salvare i dati e ricordarli in modo che gli account non si perdano. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-2 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | File CSV interni | | | | |
| **Note** | I file CSV con i dati storici sono già inclusi nel progetto, non è previsto l’upload da parte dell’utente. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | i dati storici sono già forniti all’interno del progetto, senza bisogno di caricamento esterno. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-3 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Lettura CSV | | | | |
| **Note** | Dipende dal requisito Req-1, perché è necessario disporre dei file CSV per estrarre i valori Ask e Bid. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | il sistema deve estrarre correttamente i valori Ask e Bid dalle righe valide. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-4 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Creazione grafico dinamico | | | | |
| **Note** | Si utilizza una libreria di grafici (Chart.js) per visualizzare in modo chiaro i dati Ask e Bid. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Il sistema deve rappresentare i valori Ask in blu e i valori Bid in rosso. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-5 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Aggiornamento progressivo | | | | |
| **Note** | Simula un flusso in tempo reale aggiornando il grafico ogni 10 secondi. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Ogni intervallo di tempo deve essere aggiunto un nuovo punto al grafico. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-6 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Limitazione punti grafico | | | | |
| **Note** | Serve per rendere il grafico leggibile, mantenendo visibili solo gli ultimi 30 punti. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Dopo 30 punti, i valori più vecchi devono essere rimossi automaticamente. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-7 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Saldo iniziale | | | | |
| **Note** | L’utente inizia la simulazione con un saldo fisso di 1000 chf in caso si fosse appena registrato, invece in caso dovesse avere già un account farà l’accesso e avrà lo stesso saldo dell’ultima volta. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Il saldo iniziale deve essere impostato automaticamente al caricamento della pagina. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-8 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Funzione compra | | | | |
| **Note** | Dipende dal requisito Req-2 perché serve il valore Ask corrente. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Il sistema deve permettere l’acquisto al prezzo Ask del punto corrente. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-9 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Funzione vendi | | | | |
| **Note** | Dipende dal requisito Req-2 perché serve il valore Bid corrente. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Il sistema deve permettere la vendita al prezzo Bid del punto corrente. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-10 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Gestione portafoglio | | | | |
| **Note** | Dipende dai requisiti Req-6, Req-7 e Req-8 per aggiornare saldo e asset posseduti. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Dopo ogni operazione di acquisto o vendita devono essere aggiornati saldo e quantità possedute. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-11 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Controllo errori | | | | |
| **Note** | Necessario per evitare operazioni impossibili, come comprare senza saldo o vendere senza asset. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Il sistema deve bloccare l’operazione e avvisare l’utente se non ci sono fondi o asset sufficienti. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-12 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Interfaccia grafica | | | | |
| **Note** | L’interfaccia deve essere semplice e chiara, pensata per scopi didattici. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Il grafico deve essere visualizzato nella parte superiore, mentre il saldo e i pulsanti devono stare sotto. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-13 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Compatibilità browser | | | | |
| **Note** | Il sistema deve funzionare senza problemi sui browser più comuni (Chrome, Firefox, Edge). | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Il sito deve essere eseguibile senza installazioni aggiuntive su browser moderni. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-14 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Prestazioni minime | | | | |
| **Note** | Il sistema deve gestire CSV fino a qualche migliaio di righe senza rallentamenti evidenti. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Il grafico deve aggiornarsi senza blocchi anche con file di grandi dimensioni. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-15 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Sicurezza | | | | |
| **Note** | Non gestendo dati reali o sensibili, non è necessaria un’autenticazione o crittografia. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Il sistema deve funzionare interamente lato client senza trasmettere dati all’esterno. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-16 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Salvataggio dati | | | | |
| **Note** | I dati dovranno essere salvati in un file apposito per ricordarsi account e saldo | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Continuo aggiornamento del file di salvataggio dati. | | | | |

**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.

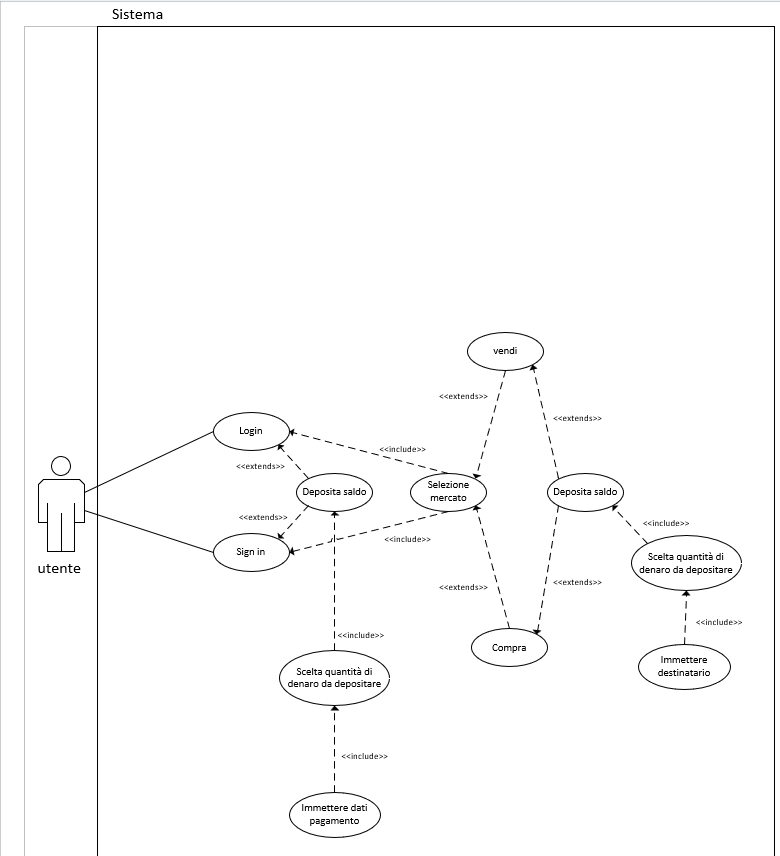
Sulla documentazione apparirà solamente l’ultima versione, mentre le vecchie dovranno essere inserite nei diari.

**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:

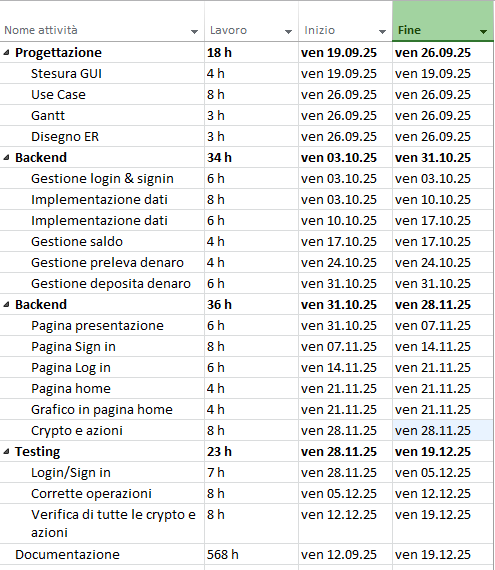


Figura 1 Esempio di diagramma di Gantt

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

Stile orizzontale

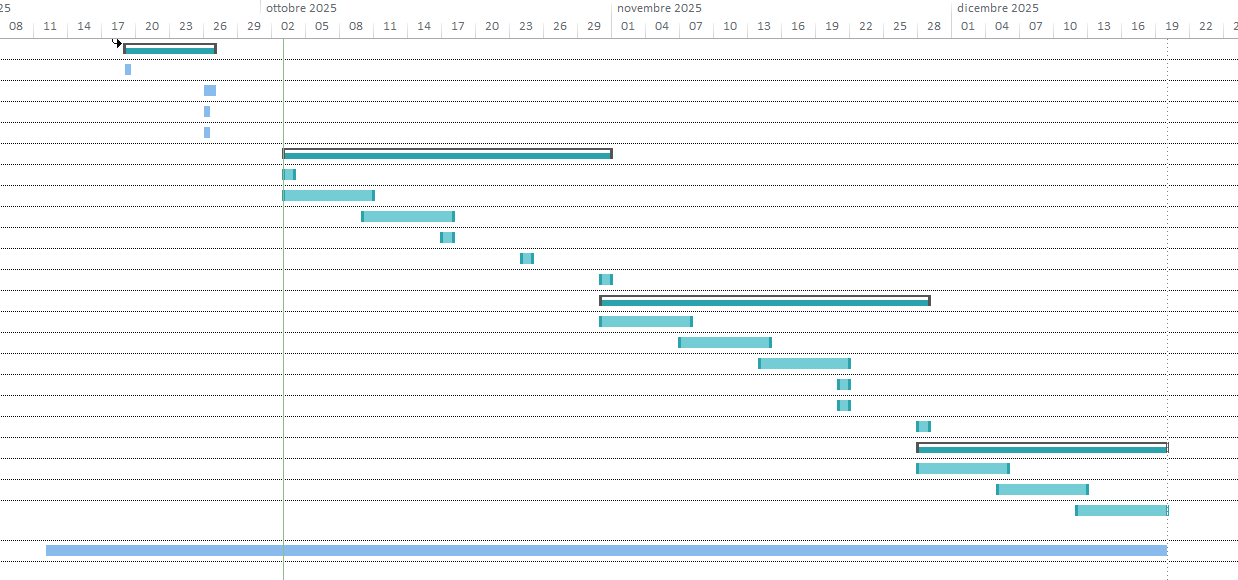


Figura 2 Esempio di diagramma di Gantt

## Analisi dei mezzi

Computer scolastico, con installato tutto il necessario: Project Professional per il diagramma di Gantt, xampp per provare il sito, Visual Studio Code per sviluppare il sito, MySQL per il database.

### Software

Visual Studio Code, per programmare in html, css, javascript e php, inoltre MySQL Workbench 8.0 per sql.

### 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.

Diagramma Chen

Figura 3 Diagramma ER Chen

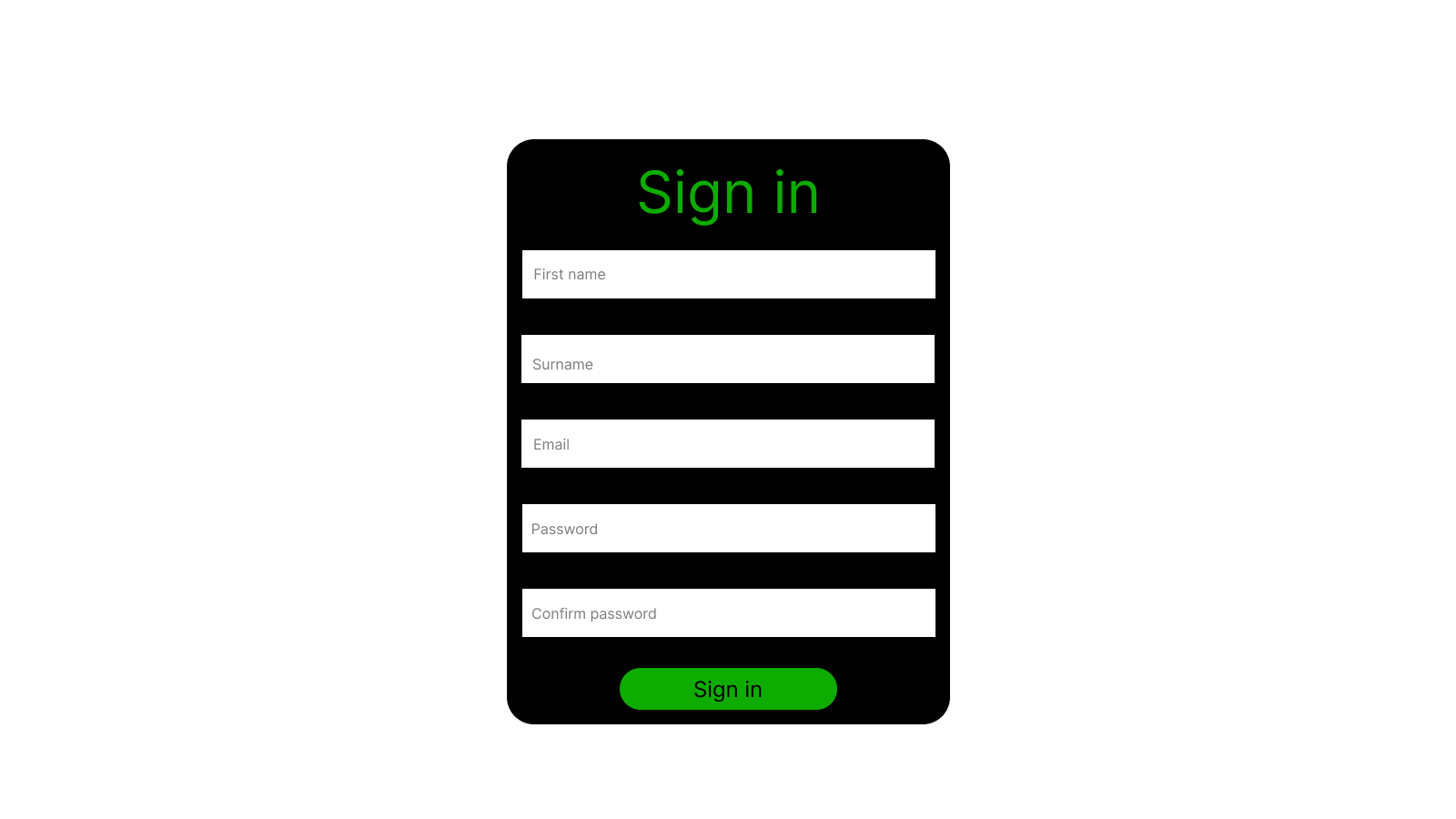
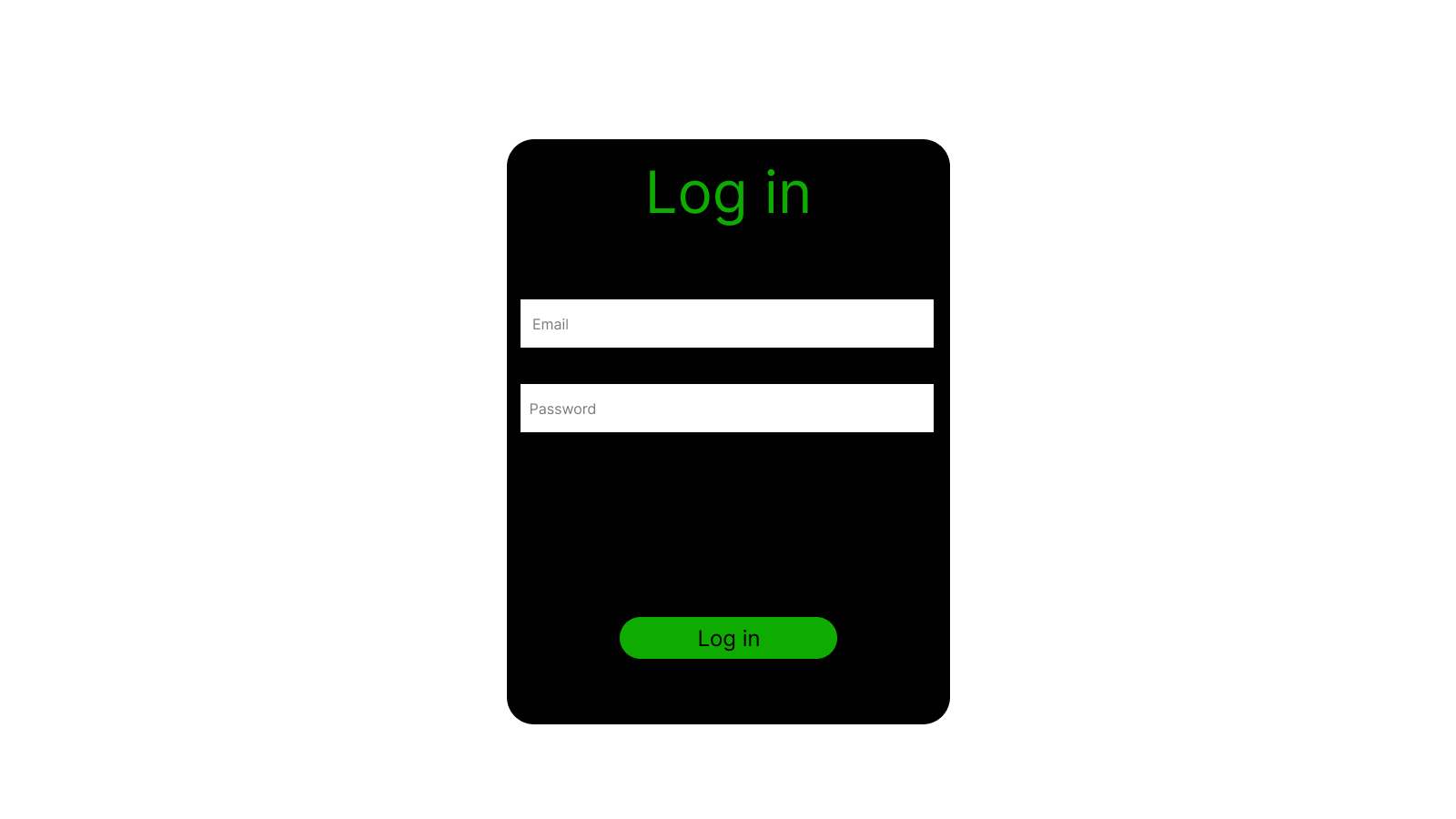
Diagramma Barker/Bachmann



Figura 4 Diagramma ER Barker

## 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.

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Risultato ottenuto** | **Stato** |
| TC-001 | Testo o immagine a conferma del test | Passato |
| TC-002 | Testo o immagine a conferma del test | Fallito |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 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).

Stile orizzontale



CONSUNTIVO

Figura 5 Esempio di diagramma di 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.

# Glossario

|  |  |
| --- | --- |
| **Termine** | **Significato** |
| ASCII | American Standard Code for Information Interchange, codice per la codifica di caratteri. |
|  |  |
|  |  |
|  |  |
|  |  |
| ZFS | Zettabyte File System, filesystem opernsource a 128 bit con funzionalità avanzate. |

# Indice delle figure

[Figura 1 Esempio di diagramma di Gantt 7](#_Toc124775984)

[Figura 2 Esempio di diagramma di Gantt 8](#_Toc124775985)

[Figura 3 Diagramma ER Chen 10](#_Toc124775986)

[Figura 4 Diagramma ER Barker 11](#_Toc124775987)

[Figura 5 Esempio di diagramma di Gantt consuntivo 16](#_Toc124775988)

# 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
* …