Gestionale Magazzino

# Indice

1 Indice 2

2 Introduzione 3

2.1 Informazioni sul progetto 3

2.2 Abstract 3

2.3 Scopo 3

3 Analisi 4

3.1 Analisi del dominio 4

3.2 Analisi e specifica dei requisiti 4

3.3 Use case 9

3.4 Pianificazione 9

3.5 Analisi dei mezzi 11

3.5.1 Software 11

3.5.2 Hardware 11

4 Progettazione 11

4.1 Design dell’architettura del sistema 11

4.2 Design dei dati e database 11

4.3 Design delle interfacce 14

4.4 Design procedurale 14

5 Implementazione 15

6 Test 15

6.1 Protocollo di test 15

6.2 Risultati test 17

6.3 Mancanze/limitazioni conosciute 17

7 Consuntivo 17

8 Conclusioni 19

8.1 Sviluppi futuri 19

8.2 Considerazioni personali 19

9 Bibliografia 20

9.1 Bibliografia per articoli di riviste: 20

9.2 Bibliografia per libri 20

9.3 Sitografia 20

10 Glossario 21

11 Indice delle figure 21

12 Allegati 21

# Introduzione

## Informazioni sul progetto

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

* Allievo/i e docente/i coinvolti nel progetto e rispettivi ruoli,
* scuola, sezione, materia/e,
* date di inizio e termine di consegna,
* …

## Abstract

*Managing a medium-scale warehouse alone, without tools that simplify the work, can become a very tedious process and allow room for error. For this project we’ve decided to create a program that allows you to manage all of this automatically and intuitively. With it a person can choose whether to manage the warehouse all alone or distribute the load among several people. Without it, coordinating multiple people and updating a list of available equipment would be nearly impossible and would take a lot of time. Instead, with the use of the Warehouse Management System, these problems are solved and the time spent making sure that all the equipment is present is pretty much non existent.*

## Scopo

### Scopi didattici

Gli obiettivi principali di questo progetto sono farci lavorare usando la metologia Agile e prepararsi per il progetto del quarto anno.

Invece riguardo alla creazione del progetto volevamo migliorare a interagire con le banche dati e integrarci un sistema di codici qr.

Invece di Apache per il webserver abbiamo deciso di usare nodeJS con il modulo Express.

### Scopi operativi

Senza un sistema di gestione del magazzino, gestirne uno di medie/grandi dimensioni diventa molto laborioso e duraturo. Perciò lo scopo del progetto è proprio quello di riuscire semplificare la gestione dei magazzini implementando un gestionale che è in grado di:

* Aggiornare Database tramite la lettura dei codici QR.
* Visualizzare una lista di oggetti e gestirla.
* Implementare un sistema a noleggi con date di scadenza.
* Implementare un sistema a utenti con diversi permessi.

# Analisi

## Analisi del dominio

Un’azienda di cinematografia ha un magazzino che contiene tutto il materiale che viene usato per registrare. Essa per gestirlo conta a mano che il materiale sia presente. Stessa cosa per quando viene portato via per fare dei progetti.

Essendo che la quantità di materiale da gestire è elevata e c’è solamente una persona a gestire il magazzino, queste azioni richiedono troppo tempo e sono troppo soggette a errori.

Per semplificare la gestione del magazzino verranno usati dei codici QR sul materiale per poterli memorizzare e scansionare in maniera veloce.

Per la parte di presa del materiale per i progetti viene introdotto un sistema che ti permette di vedere che cos’è disponibile e che gestisce in maniera automatica la parte di “noleggio”.

Esso verrà utilizzato principalmente dal gestore del magazzino. Esso dovrà leggere il manuale di utilizzo per capire il pieno funzionamento dell’applicativo e ridurre il margine di errore.

## Analisi e specifica dei requisiti

In base alle direttive del cliente, sono state date le seguenti specifiche per il software:

|  |  |
| --- | --- |
| **Req-001** | |
| **Nome** | Presenza database |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Deve essere presente un database dove verranno memorizzati tutti i dati riguardo al noleggio del materiale e le credenziali di accesso al sito |

|  |  |
| --- | --- |
| **Req-002** | |
| **Nome** | Login |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’utente necessita di un’interfaccia per accedere al sito.  Solo i dipendenti dell’azienda potranno fare l’accesso al sito. |
| **Sotto requisiti** | |
| **Req-001** | Si necessita la presenza di un database funzionante |

|  |  |
| --- | --- |
| **Req-003** | |
| **Nome** | Registrazione nuovi articoli |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il gestore del magazzino deve poter registrare dei nuovi prodotti.  Quando viene registrato un nuovo prodotto, viene generato un codice QR |
| **Sotto requisiti** | |
| **Req-002** | L’utente deve aver fatto l’accesso al sito “gestore magazzino” |

|  |  |
| --- | --- |
| **Req-004** | |
| **Nome** | Registrazione nuove categorie |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il gestore del magazzino deve poter registrare delle nuove categorie di prodotti |
| **Sotto requisiti** | |
| **Req-002** | L’utente deve aver fatto l’accesso al sito come “gestore magazzino” |

|  |  |
| --- | --- |
| **Req-005** | |
| **Nome** | Visualizzazione articoli |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’utente deve poter visualizzare tutti gli articoli dell’azienda |
| **Sotto requisiti** | |
| **Req-002** | L’utente deve aver fatto l’accesso al sito |
| **Req-003** | Devono essere presenti degli articoli (già registrati) |

|  |  |
| --- | --- |
| **Req-006** | |
| **Nome** | Visualizzazione informazioni singolo articolo |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’utente deve poter visualizzare le informazioni di un determinato articolo |
| **Sotto requisiti** | |
| **Req-002** | L’utente deve aver fatto l’accesso al sito |
| **Req-003** | Devono essere presenti degli articoli |

|  |  |
| --- | --- |
| **Req-007** | |
| **Nome** | Gestione prodotto tramite QR |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’utente deve poter scannerizzare il codice QR per visualizzare la scheda del prodotto |
| **Sotto requisiti** | |
| **Req-002** | L’utente deve aver fatto l’accesso al sito |
| **Req-003** | Devono essere presenti degli articoli |

|  |  |
| --- | --- |
| **Req-008** | |
| **Nome** | Gestione noleggi |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’utente deve poter gestire i noleggi degli articoli |
| **Sotto requisiti** | |
| **Req-002** | L’utente deve aver fatto l’accesso al sito |
| **Req-003** | Devono essere presenti degli articoli |

|  |  |
| --- | --- |
| **Req-009** | |
| **Nome** | Stampa codice QR |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’utente deve poter stampare tramite etichettatrice i codici QR che fanno riferimento ad un articolo |
| **Sotto requisiti** | |
| **Req-002** | L’utente deve aver fatto l’accesso al sito |
| **Req-003** | Devono essere presenti degli articoli |
| **Req-006** | Visualizzazione QR dalle informazioni del prodotto |

|  |  |
| --- | --- |
| **Req-010** | |
| **Nome** | Sistema di allerta per restituzione noleggio |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Deve esserci un sistema di notifica tramite email che deve avvisare l’utente quando sta per scadere il noleggio.  Se il noleggio è scaduto, deve rinviare l’email all’utente e deve allertare il gestore del magazzino. |

|  |  |
| --- | --- |
| **Req-011** | |
| **Nome** | Gestione inventario |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il gestore del magazzino deve poter gestire l’inventario controllando che ci sia tutto il materiale |
| **Sotto requisiti** | |
| **Req-002** | L’utente deve aver fatto l’accesso al sito come “gestore magazzino” |
| **Req-003** | Devono essere presenti degli articoli |

|  |  |
| --- | --- |
| **Req-012** | |
| **Nome** | Visualizzazione lista utenti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’amministratore deve poter visualizzare la lista di tutti gli utenti presenti nell’applicativo |
| **Sotto requisiti** | |
| **Req-002** | L’utente deve aver fatto l’accesso al sito come “amministratore” |

|  |  |
| --- | --- |
| **Req-013** | |
| **Nome** | Gestione utenti sito |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’amministratore deve poter gestire gli utenti del sito, quindi deve poterli creare ed eliminare |
| **Sotto requisiti** | |
| **Req-002** | L’utente deve aver fatto l’accesso al sito come “amministratore” |

**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

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.

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

## Installazione MySQL

Sul server abbiamo deciso di installare il database MySQL per gestire tutti i dati dell’applicativo.

La versione installata è la 8.0.35.

I comandi eseguiti per l’installazione sono stati i seguenti:

Figura 5 Comandi installazione MySQL

sudo apt update

sudo apt install mysql-server

sudo mysql

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'tmp\_pass';

exit

sudo mysql\_secure\_installation

n

y

y

y

y

## Installazione Nodejs

Per il backend del nostro applicativo WEB è stato utilizzato Nodejs.

La versione installata è la LTS, ovvero la 20.11.0.

I seguenti comandi servono per installare curl, Nodejs e NPM:

Figura 6 Comandi installazione Nodejs

sudo apt install curl

curl -sL [https://deb.nodesource.com/setup\_20.x -o nodesource\_setup.sh](https://deb.nodesource.com/setup_20.x%20-o%20nodesource_setup.sh)

sudo bash ./nodesource\_setup.sh

sudo apt-get install nodejs -y

## 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 7 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
* …