Gestione panetteria

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 7

2.4 Pianificazione 8

2.4.1 Analisi 10

2.4.2 Progettazione 10

2.4.3 Implementazione 10

2.4.4 Testing 11

2.4.5 Documentazione 11

2.5 Analisi dei mezzi 11

2.5.1 Software 11

2.5.2 Hardware 12

3 Progettazione 12

3.1 Design dell’architettura del sistema 12

3.2 Design dei dati e database 12

3.3 Design delle interfacce 12

3.4 Design procedurale 12

4 Implementazione 13

5 Test 13

5.1 Protocollo di test 13

5.2 Risultati test 13

5.3 Mancanze/limitazioni conosciute 14

6 Consuntivo 14

7 Conclusioni 14

7.1 Sviluppi futuri 14

7.2 Considerazioni personali 14

8 Bibliografia 14

8.1 Bibliografia per articoli di riviste: 14

8.2 Bibliografia per libri 14

8.3 Sitografia 14

9 Allegati 15

# Introduzione

## Informazioni sul progetto

* Allievo: Mattia Lazzaroni.   
  Superiore professionale: Massimo Sartori.
* Scuola d’Arti e Mestieri di Trevano, sezione Informatica, classe 4, progetti individuali.
* Data inizio: 24.01.2020  
  Data fine: 06.04.2020

## 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.*

## Scopo

Lo scopo di questo progetto è quello di creare un sito Web che permetta ai clienti di vedere le ordinazioni effettuate e di farne di nuove. Per poter effettuare le ordinazioni gli utenti dovranno registrarsi e confermare la registrazione tramite il link che verrà inviato per email. Ci sarà anche una sezione riservata agli amministratori che potranno gestire la panetteria (i prodotti, le tipologie, i prezzi, i giorni di apertura/ordinazione, le ordinazioni, l’aspetto contabile e la homepage). Ci saranno poi i venditori e i panettieri che in generale potranno visualizzare e gestire la panetteria, quindi i clienti, le ordinazioni effettuate. Egli potranno anche gestire il sistema di sconti del sito. Infine, ci saranno i visitatori, ovvero coloro che entrano nel sito web senza registrarsi e possono vedere le informazioni generali sulla panetteria.

# Analisi

## Analisi del dominio

Spesso, per poter gestire dei negozi vengono utilizzati dei siti web confusionali, complicati da usare e poveri di funzionalità o con delle cose che non funzionano. Il mio prodotto servirà a gestire un’ipotetica panetteria, quindi a visualizzare tutte le ordinazioni che vengono effettuate dai clienti e a gestire le varie disponibilità e i guadagni del sito. Per poter accedere al mio sito, sarà necessario come minimo un dispositivo con accesso ad Internet, un’e-mail, un recapito telefonico e una conoscenza basilare per poter usare un sito Web munito di registrazione e conferma tramite e-mail.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Creazione pagina di registrazione |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Chiunque avrà accesso alla pagina di registrazione |
| **002** | I campi da riempire per poter completare una riservazione sono: nome, cognome, e-mail, password e numero di telefono |
| **003** | L’utente stesso che si registra dovrà confermare la registrazione |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Creazione pagina di login |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Solo gli utenti registrati possono effettuare l’accesso |
| **002** | Per effettuare l’accesso basterà inserire l’e-mail e la password |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Creazione pagina amministratore |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | L’amministratore potrà gestire i prodotti |
| **002** | L’amministratore potrà gestire le tipologie con i prezzi |
| **003** | L’amministratore potrà gestire i giorni di apertura/ordinazione |
| **004** | L’amministratore potrà gestire le ordinazioni |
| **005** | L’amministratore potrà gestire l’aspetto contabile |
| **006** | L’amministratore potrà gestire la homepage |
| **007** | L’amministratore potrà scegliere degli sconti in base al periodo o ai clienti. |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Creazione pagina clienti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | I clienti potranno registrarsi |
| **002** | I clienti potranno loggarsi |
| **003** | I clienti potranno vedere le ordinazioni fatte e il loro stato |
| **004** | I clienti potranno eseguire nuovi ordini |
| **005** | I clienti potranno stampare i dettagli dell’ordinazione appena effettuata |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Creazione pagina visitatori |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | I visitatori vedranno la homepage del sito con le informazioni generali sulla panetteria |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Creazione pagina panettiere |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Il panettiere potrà avere il resoconto delle ordinazioni |
| **002** | Il panettiere potrà effettuare delle ricerche |
| **003** | Il panettiere potrà registrare una vendita nel sistema |
| **004** | Il panettiere avrà accesso a delle statistiche |
| **005** | Il panettiere potrà confermare l’avvenuta consegna e/o pagamento |
| **006** | Il panettiere potrà stampare lo scontrino |

## Use case

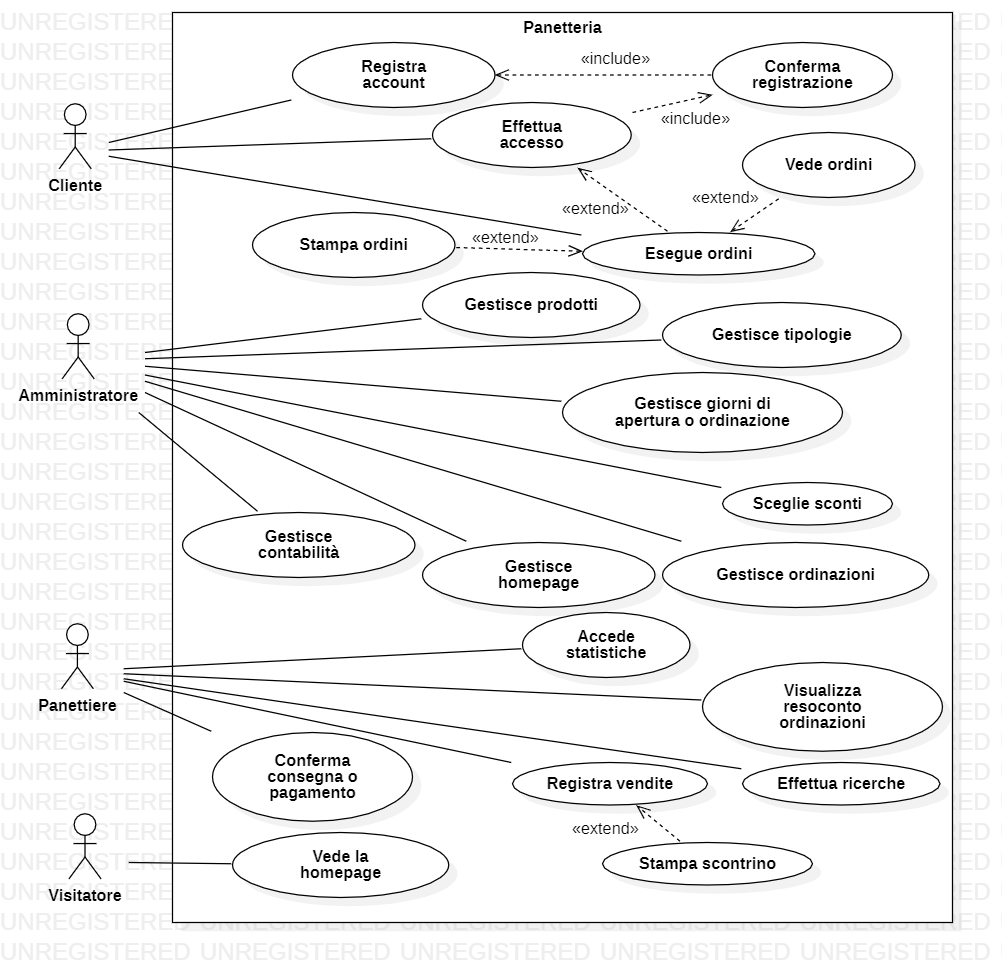
Questo è il diagramma Use Case che ho realizzato per questo progetto:

Figura , Use Case

Nel diagramma è possibile notare ben cinque attori: cliente, amministratore, panettiere, venditore e visitatore.

Il cliente per poter gestire gli ordini (eseguirli, stamparli e vederli) deve prima aver effettuato l’accesso. Prima di poter però effettuare l’accesso, è necessario che venga registrato un account e che venga poi confermato dal cliente stesso tramite un’e-mail. Prima di poter vedere o stampare un ordine deve ovviamente eseguirlo.

L’amministratore, invece, si occupa semplicemente della gestione del sito. Può quindi gestire i prodotti offerti dalla panetteria, le tipologie dei pani, i giorni di apertura o di ordinazione, le ordinazioni effettuate dai clienti, la homepage del sito e la parte di contabilità del sito. In più, egli può anche scegliere eventualmente di aggiungere degli sconti nel sito per un certo periodo.

Per quanto riguarda il panettiere, egli ha la possibilità di accedere alle statistiche della panetteria, visualizzare il resoconto delle ordinazioni, effettuare delle ricerche, confermare l’avvenuta consegna e/o pagamento e registrare le vendite fatte dalla panetteria; Vendite di cui può poi eventualmente stampare lo scontrino.

Infine, c’è il visitatore che può solamente vedere la homepage con le informazioni principali della panetteria.

## Pianificazione

Per la realizzazione di questo progetto ho ovviamente dovuto creare un Gantt preventivo in cui ho provato a stimare tutti i vari tempi per ogni attività.

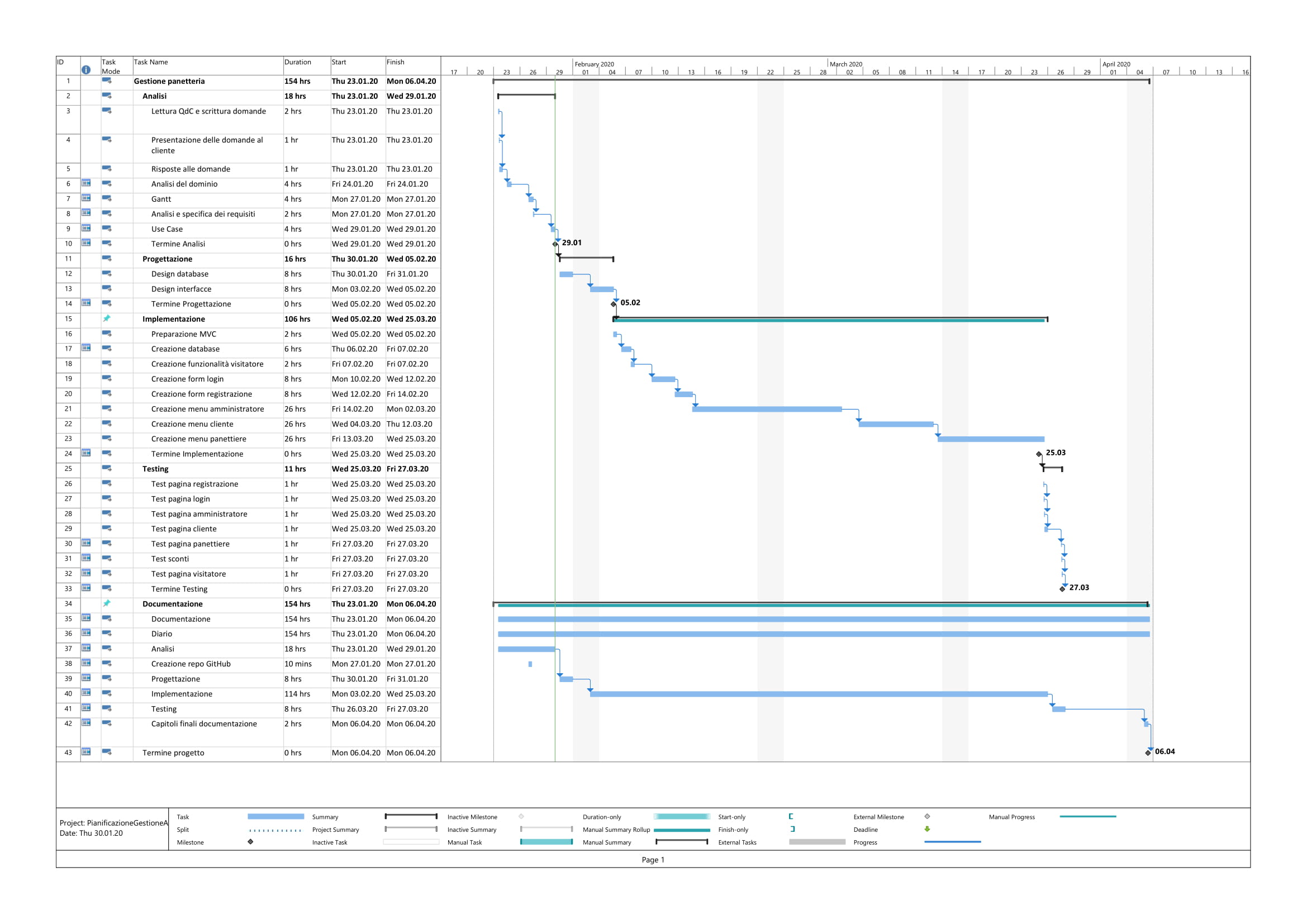


Figura 2, Gantt preventivo

Questo Gantt si divide in 5 fasi principali: Analisi, Progettazione, Implementazione, Testing e Documentazione.

### Analisi

Figura , Analisi del Gantt preventivo

Per la parte di analisi, inizialmente ho pensato di spendere del tempo per leggere il quaderno dei compiti e per pensare alle domande da fare al cliente. Ho poi posto le domande al cliente e mi sono annotato le risposte. Dopodiché, ho previsto 4 ore per l’analisi del dominio e altre 4 per la stesura del Gantt preventivo. Infine, ho lasciato 2 ore per la scrittura dei requisiti e 4 per gli use case.

### Progettazione

Figura , Progettazione del Gantt preventivo

Per quanto riguarda la progettazione, che dura solamente 16 ore, vado inizialmente a creare il design del Database e poi procedo semplicemente con i vari mockup delle pagine del sito.

### Implementazione

Figura , Implementazione del Gantt preventivo

L’implementazione, che durerà all’incirca due mesi, comincerà con la preparazione dell’ambiente di lavoro e della struttura MVC. Dopo di questo partirò subito con la creazione del database, che sarà poi fondamentale per poter realizzare il form di login e di registrazione. Dopo aver creato il meccanismo di accesso e registrazione al sito, creerò la pagina che vedranno i visitatori con le informazioni della panetteria. Infine, ho pensato di assegnare 26 ore per ogni tipo di utente che necessiterà di funzionalità. Questi tipi di utenti sono: amministratori, clienti, panettieri.

### Testing

Figura , Testing del Gantt preventivo

La fase di testing semplicemente servirà per testare tutto quello che avrò fatto nella parte di implementazione, quindi i due form di login e tutte le pagine per i vari utenti, compresi i visitatori. Ho poi anche aggiunto un test per verificare il funzionamento degli sconti nel sito.

### Documentazione

Figura , Documentazione del Gantt preventivo.

Infine, come ultima attività, possiamo vedere la documentazione, che dura tutto il progetto e contiene i diari giornalieri e le 4 fasi principali (analisi, progettazione, implementazione e testing). Ho anche inserito la creazione della repository GitHub. All’ultima riga è presente la milestone che indica la fine del progetto.

## Analisi dei mezzi

### Software

I software utilizzati per la realizzazione di questo progetto sono:

* Microsoft Word 2016
* Microsoft Power Point 2016
* Microsoft Project Professional 2016
* Google Chrome 79.0.3945.130
* GitHub Desktop 2.1.3
* Visual Studio Code 1.34.0
* MySQL Workbench 8.0.17
* StarUML 3.2.2
* Draw.io 12.4.2 (<https://draw.io>)
* MockFlow 2019 (<http://wireframepro.mockflow.com/>)

### Hardware

* Computer portatile:
  + Modello: Acer Aspire E 15
  + Processore: Intel Core i7 7500u
  + RAM: 16GB LPDDR4 2133 MHz
  + GPU: NVIDIA 940MX
  + SSD: 256 GB

# 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
* …