

Gestione quaderno dei compiti e valutazioni

Titolo del progetto: Gestione quaderno dei compiti e valutazioni
Alunno/a: Lucas Previtali
Classe: I4AA
Anno scolastico: 2019/2020
Docente responsabile: Ugo Bernasconi

1	Introduzione.....	3
1.1	Informazioni sul progetto.....	3
1.2	Abstract.....	3
1.3	Scopo.....	3
	Analisi.....	4
1.4	Analisi del dominio.....	4
1.5	Analisi e specifica dei requisiti.....	4
1.6	Use case.....	8
1.7	Pianificazione.....	9
1.8	Analisi dei mezzi.....	9
1.8.1	Software.....	9
1.8.2	Hardware.....	9
1.8.3	Analisi dei costi.....	10
2	Progettazione.....	10
2.1	Design dell'architettura del sistema.....	10
2.2	Design dei dati e database.....	10
2.3	Design delle interfacce.....	10
2.4	Design procedurale.....	10
3	Implementazione.....	11
4	Test.....	11
4.1	Protocollo di test.....	11
4.2	Risultati test.....	13
4.3	Mancanze/limitazioni conosciute.....	13
5	Consuntivo.....	13
6	Conclusioni.....	14
6.1	Sviluppi futuri.....	14
6.2	Considerazioni personali.....	14
7	Bibliografia.....	14
7.1	Bibliografia per articoli di riviste.....	14
7.2	Bibliografia per libri.....	14
7.3	Sitografia.....	14
8	Allegati.....	14

1 Introduzione

1.1 Informazioni sul progetto

Progetto svolto da: Lucas Previtali

Mandante del progetto: Ugo Bernasconi

Docente Responsabile: Ugo Bernasconi

Scuola: Arti e Mestieri Trevano

Sezione: Informatica

Classe: I4AA

Data d'inizio: 03.09.2019

Termine della consegna: 20.12.2019

1.2 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.

1.3 Scopo

Questo progetto nasce per semplificare il lavoro della creazione di un quaderno dei compiti e per velocizzare il processo dell'assegnazione dei punti per la valutazione. Lo scopo è quello di creare un applicativo che si occupi di far visualizzare i criteri di valutazione per un progetto e che dia la possibilità di scegliere quelli adatti a dipendenza del progetto in questione. Oltre a questo per rendere il più completo possibile il quaderno dei compiti deve essere possibile inserire le varie informazioni sul candidato al quale il progetto è assegnato e altre informazioni generiche. Per quanto riguarda la parte relativa alle valutazioni, l'applicativo dovrebbe riuscire ad interfacciarsi con Word per riuscire a prendere i requisiti scelti. Deve poi mostrarli e deve essere possibile inserire i punti per ogni requisito (una volta terminato il progetto). Inoltre deve essere possibile anche inserire le motivazioni complete per ogni valutazione data.

Analisi

1.4 Analisi del dominio

L'applicativo sarà usato principalmente dai docenti che creano i quaderni dei compiti. Non necessariamente questi docenti saranno informatici, quindi l'applicazione deve essere usabile in modo intuitivo e veloce da chiunque, anche da chi non ha competenze informatiche più che basilari. Siccome lo scopo del progetto è quello di rendere più semplice tutto il processo di creazione di un quaderno dei compiti e della successiva assegnazione dei punti per ogni requisito, il programma essere veloce e facile da usare.

1.5 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.

ID: REQ-001	
Nome	Ambiente di sviluppo adeguato e funzionale
Priorità	1
Versione	1.0
Note	-
Sotto requisiti	
001	Installare la versione migliore di Visual Studio (che sia recente ma allo stesso tempo stabile). Per questo progetto ho scelto Visual Studio 2017 Versione 15.9.6.
002	Installare tutti i componenti aggiuntivi necessari.

ID: REQ-002	
Nome	View per l'inserimento di informazioni generiche
Priorità	1
Versione	1.0
Note	-
Sotto requisiti	
001	Sviluppare una view che permetta l'inserimento di tutte le informazioni generiche relative al progetto (dati candidato, dati formatore, date, ecc.).

ID: REQ-003	
Nome	View per la descrizione
Priorità	1
Versione	1.0
Note	-
Sotto requisiti	
001	Sviluppare una semplice view per l'inserimento della descrizione del progetto

ID: REQ-004	
Nome	View per la scelta dei requisiti
Priorità	1
Versione	1.0
Note	-
Sotto requisiti	
001	Creare una view che permetta di scegliere i 7 requisiti A14-A20 da assegnare al candidato
002	Far visualizzare sia il codice che il titolo (eventualmente descrizione)

ID: REQ-005	
Nome	Interfacciamento tra documento word e view per la scelta dei requisiti
Priorità	1

Versione	1.0
Note	-
Sotto requisiti	
001	Fare in modo che la view si interfacci al documento word nel quale sono contenuti tutti i requisiti estesi
002	Scrivere il titolo del requisito in base al codice dato

ID: REQ-006	
Nome	View per l'assegnazione dei punteggi.
Priorità	1
Versione	1.0
Note	-
Sotto requisiti	
001	Creare una view che permetta l'assegnazione del punteggio per ogni requisito scelto in precedenza.
002	Visualizzare i requisiti in maniera estesa

ID: REQ-007	
Nome	Motivazioni dei voti.
Priorità	2
Versione	1.0
Note	-
Sotto requisiti	
001	Prevedere un sistema di copia-incolla per rendere più veloce la parte della motivazione del punteggio assegnato.

ID: REQ-008	
Nome	Modifica di un qdc esistente
Priorità	3
Versione	1.0
Note	-

Sotto requisiti	
001	Fare in modo che sia possibile modificare un qdc già esistente. (Modifica delle info generiche, della descrizione e eventualmente dei requisiti scelti).

ID: REQ-009	
Nome	Sviluppo della banca dati
Priorità	1
Versione	1.0
Note	-
Sotto requisiti	
001	Ideare una banca dati adeguata ai requisiti
002	Generare e aggiornare la banca dati direttamente dal codice tramite SQLServer

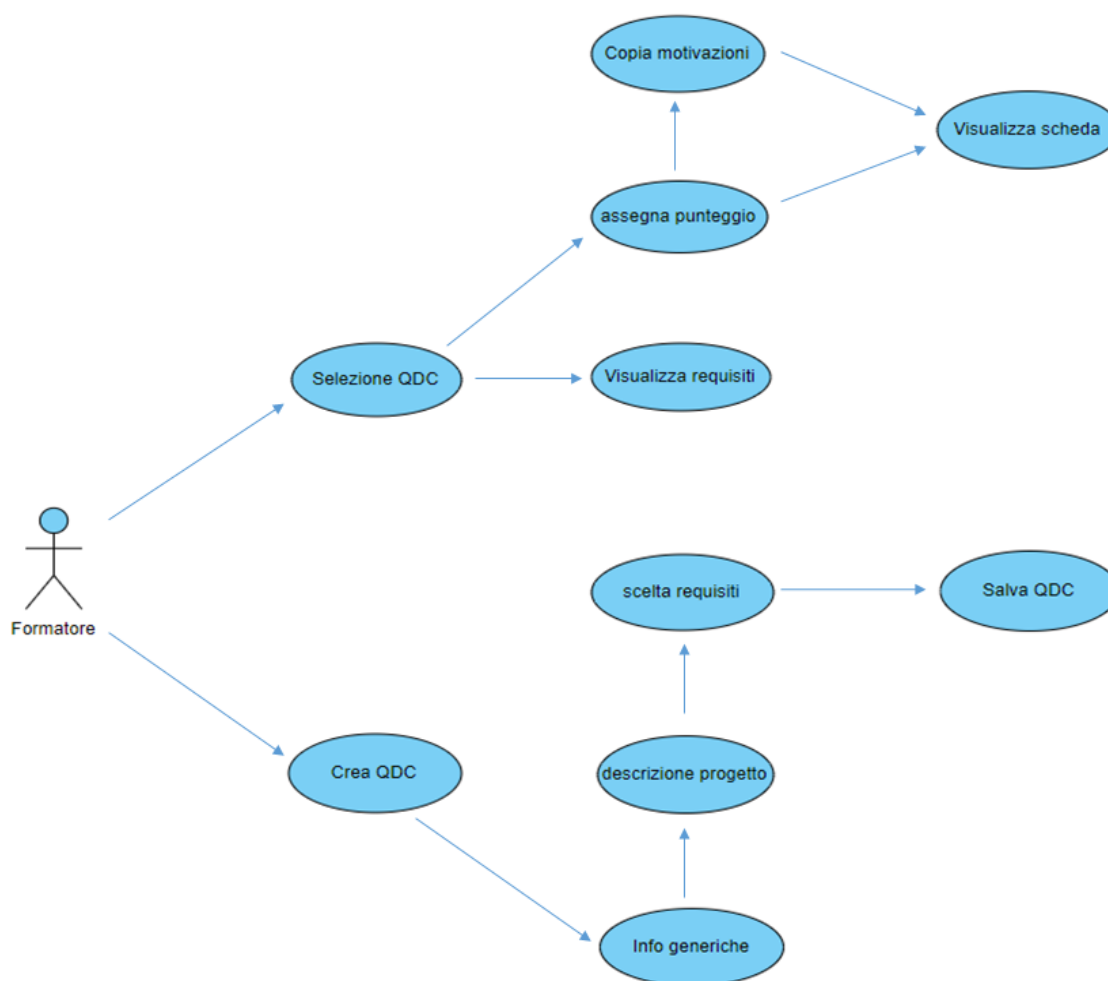
ID: REQ-010	
Nome	Sviluppo dell'applicativo con modello MVC
Priorità	1
Versione	1.0
Note	-
Sotto requisiti	
001	Sviluppare l'intero applicativo usando il modello MVC.

1.6 Use case

L'applicativo è pensato per essere usato esclusivamente dai docenti formatori che si occupano di preparare un quaderno dei compiti, per questo motivo nello use case è presente un solo attore, ovvero il *Formatore*. Il formatore quando apre l'applicazione ha la possibilità di scegliere se creare un Quaderno dei compiti nuovo (operazione di default) oppure se selezionarne uno già esistente.

Nel caso in cui scelga la prima opzione si troverà una maschera nella quale potrà inserire le informazioni generiche riguardo al progetto (informazioni formatore, informazioni candidato, titolo, date e altro). Una volta inserite queste informazioni potrà scrivere la descrizione del progetto (o copiarla). Questo avviene in un'altra maschera. Dopodiché si arriva alla parte più importante, cioè quella della scelta dei requisiti. In questa maschera il formatore potrà andare a selezionare i sette requisiti A14 – A20 da assegnare al candidato oppure potrà semplicemente inserire il numero ID del requisito e il programma si occuperà di trovare nel file word contenente tutti i requisiti in versione estesa il requisito giusto. A questo punto sarà possibile salvare il quaderno dei compiti appena creato.

Nel caso in cui invece venga scelta la seconda opzione, il docente dovrà selezionare uno dei qdc creati in precedenza. Al quaderno dei compiti selezionato potranno essere aggiunte le valutazioni per ognuno dei sette requisiti scelti e potranno essere inserite o copiate le motivazioni per il voto assegnato. Infine potrà essere visualizzata la scheda.



1 Use Case

1.7 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:

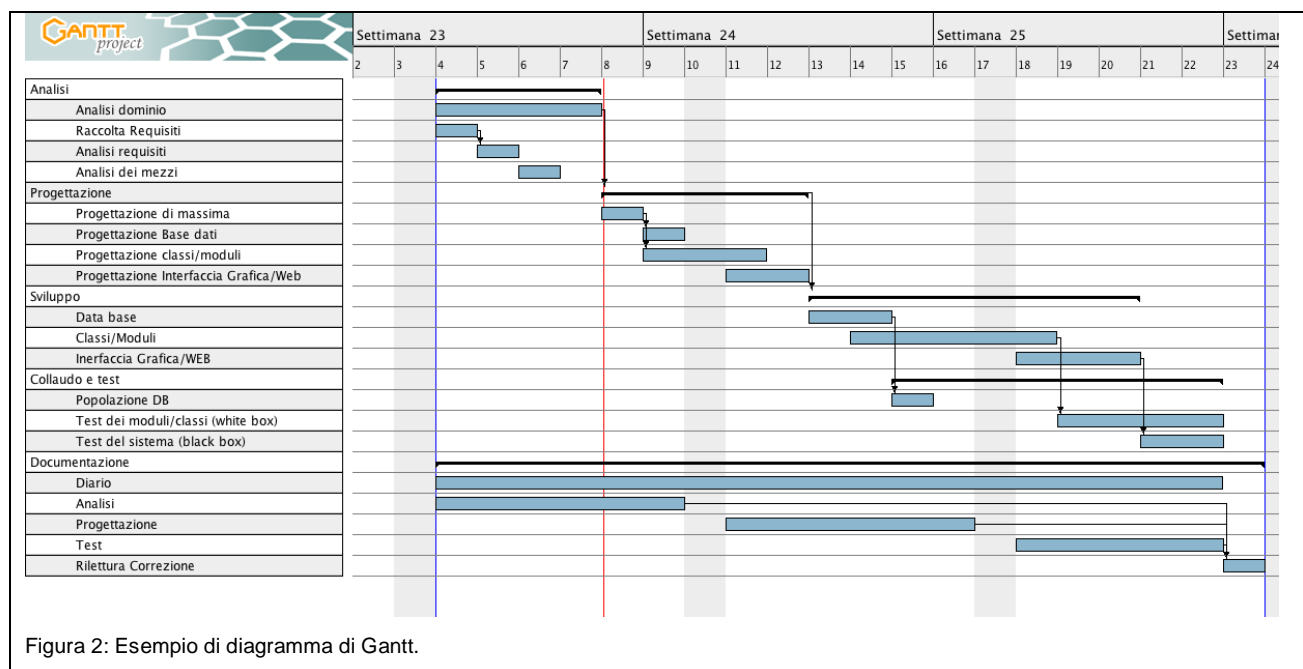


Figura 2: Esempio di diagramma di Gantt.

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

1.8 Analisi dei mezzi

Per lo svolgimento del progetto non servono particolari software o componenti hardware al di fuori da ciò che riguarda Visual Studio (con le varie librerie e plugins elencati nei punti successivi).

1.8.1 Software

Software	Versione	Note
MySQL	5.6.16	? (usato per interagire con il database)
Windows	10	Sistema operativo usato per lo svolgimento dell'intero progetto
Visual Studio 2017	15.9.6	IDE usato per lo sviluppo dell'applicativo (linguaggio c#, progettazione secondo standard MVVM)
NuGet	4.1	?

1.8.2 Hardware

Hardware	Modello	Note
HP	7-x185nz	Computer personale usato per lo svolgimento del progetto
?	?	Computer fisso della scuola usato per lo svolgimento del progetto.

1.8.3 Analisi dei costi

Considero come costo per persona 60 franchi all'ora.

Costo per ora	Ore	Impiegati	Totale
60 CHF	174	1	10440 CHF

2 Progettazione

Questo capitolo descrive esaurientemente 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.

2.1 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

2.2 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.

2.3 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.

2.4 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.

3 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.

4 Test

4.1 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:	TC-001	Nome:	Funzionamento ambiente di lavoro
Riferimento:	RQ_001 / RQ_010		
Descrizione:			
Prerequisiti:	Visual Studio installato con tutti i plugins necessari		
Procedura:	<ol style="list-style-type: none"> 1. Aprire Visual Studio e creare un nuovo progetto 2. Eseguire il programma MenuSAMT.vsix 3. Verificare il funzionamento del modello MVC 4. Provare a salvare un programma 5. Chiudere Visual Studio, riaprirlo e verificare che il modello MVC funzioni ancora 		
Risultati attesi:	Dopo aver fatto partire il programma MenuSAMT il template MVC viene creato. Se si chiude VS e si riapre in seguito il template rimane funzionante.		

Test Case:	TC-002	Nome:	Funzionamento database (SQLServer)
Riferimento:	RQ_009		
Descrizione:			
Prerequisiti:	Database creato correttamente e accessibile		
Procedura:	<ol style="list-style-type: none"> 1. Creare le tabelle del database partendo dallo schema ER 2. Inserire manualmente qualche dato per il test 3. Provare a eseguire qualche semplice query di select, insert e update per verificare l'effettivo funzionamento del database 		
Risultati attesi:	Gli insert e gli update e tutti gli altri comandi principali devono funzionare correttamente. Il database si aggiorna ed è accessibile dal codice.		

Test Case:	TC-003	Nome:	Inserimento dei dati generici
Riferimento:	RQ_002		
Descrizione:			
Prerequisiti:	View per l'inserimento dei dati generici già creata. Interfaccia con il database funzionante (TS_002)		
Procedura:	<ol style="list-style-type: none"> 1. Inserire più volte dei dati nei vari campi della view per i dati generici 2. Controllare che tali dati vengano salvati sul database 		
Risultati attesi:	Quando vengono inseriti dei dati nei vari campi, dopo che viene premuto il pulsante "Salva e continua", questi vengono salvati nel database nelle giuste colonne.		

Test Case:	TC-004	Nome:	Inserimento di descrizioni
Riferimento:	RQ_003		
Descrizione:			
Prerequisiti:	View per l'inserimento delle descrizioni già creata. Interfaccia con il database funzionante (TS_002)		
Procedura:	<ol style="list-style-type: none"> 1. Inserire più volte dei dati nei vari campi della view per le descrizioni 2. Controllare che tali dati vengano salvati sul database 		
Risultati attesi:	Quando viene inserita una descrizione dopo che viene premuto il pulsante "Salva e continua", questi vengono salvati nel database nella colonna giusta.		

Test Case:	TC-005	Nome:	Scelta dei requisiti
Riferimento:	RQ_004 / RQ_005		
Descrizione:			
Prerequisiti:	View per la scelta dei requisiti già creata. Interfaccia con il database funzionante (TS_002).		
Procedura:	<ol style="list-style-type: none"> 1. Provare a scegliere sette requisiti dalla select 2. Salvare il qdc e verificare che i requisiti vengano salvati correttamente sul database, compresa l'assegnazione (tabella assegna). 3. Provare a scegliere i sette requisiti inserendo solo l'id 4. Verificare che venga scelto il requisito corrispettivo all'id partendo dal file word contenente tutti i requisiti per esteso. 		
Risultati attesi:	Gli insert e gli update e tutti gli altri comandi principali devono funzionare correttamente. Il database si aggiorna ed è accessibile dal codice.		

Test Case:	TC-006	Nome:	Assegnazione di punteggi
Riferimento:	RQ_006		
Descrizione:			
Prerequisiti:	View per assegnazione dei punteggi già creata. Interfaccia con il database funzionante (TS_002)		
Procedura:	<ol style="list-style-type: none"> 1. Inserire più volte dei punteggi per ogni requisito A14-A20 		

	2. Controllare che i punteggi vengano scritti correttamente sul database
Risultati attesi:	Quando vengono inseriti dei punteggi, dopo che viene premuto il pulsante “Salva e continua”, questi vengono salvati nel database nelle giuste colonne.

Test Case:	TC-007	Nome:	Copia-incolla motivazioni)
Riferimento:	RQ_007		
Descrizione:			
Prerequisiti:	View per assegnazione dei punteggi già creata. Interfaccia con il database funzionante (TS_002)		
Procedura:	1. Dopo aver inserito dei punteggi, provare a cliccare su “copia” e verificare che vengano copiate le motivazioni direttamente dal file word con i requisiti estesi		
Risultati attesi:	Quando viene premuto “copia”, vengono copiate le motivazioni per quel requisito direttamente dal file apposito.		

Test Case:	TC-008	Nome:	Controllo dati
Riferimento:	-		
Descrizione:			
Prerequisiti:	Varie view già create e funzionanti.		
Procedura:	1. Inserire in tutte le view create dei dati validi 2. Controllare che questi dati vengono salvati correttamente sul database 3. Inserire nelle varie view qualche dato non valido 4. Controllare che nel database non venga scritto nulla e che venga richiesto di inserire dei dati validi		
Risultati attesi:	Se vengono inseriti dei dati validi, questi vengono salvati sul database. Se in una view viene inserito un dato non valido, nessun dato di quella view viene salvato sul database e viene richiesto di inserire nuovamente i campi contenenti i dati non validi.		

4.2 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.

4.3 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.

5 Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap 1.7) (ad esempio Gantt consuntivo).

6 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

6.1 Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

6.2 Considerazioni personali

Cosa ho imparato in questo progetto? ecc

7 Bibliografia

7.1 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,

7.2 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.

7.3 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.

8 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
- ...

