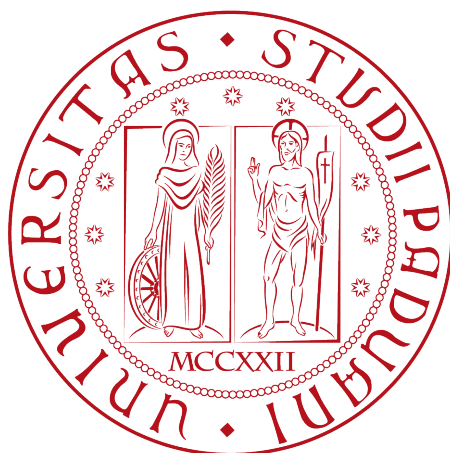


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA TRIENNALE IN INFORMATICA



**Sviluppo di un modulo web per la gestione
delle campagne marketing in ambiente
bancario/fintech**

Tesi di laurea

Relatore

Prof. Ombretta Gaggi

Laureando

Michael Amista'

ANNO ACCADEMICO 2022-2023

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecentoventi ore, dal laureando Michael Amista' presso l'azienda CWBI (Codice Web Banking Innovation).

L'obiettivo dello stage riguarda lo sviluppo di un modulo web per una webapp aziendale preesistente. In particolare ciò che è stato sviluppato è un sistema di gestione delle campagne a supporto dell'area marketing in un contesto bancario/fintech. Vincolo di sviluppo sono le metodologie di analisi, progettazione e sviluppo aziendale, oltre allo stack tecnologico utilizzato dall'azienda. Viene inoltre riportata un'analisi finale che pone a confronto le aspettative iniziali personali con quelli che sono stati i risultati finali e gli obiettivi raggiunti.

Ringraziamenti

Innanzitutto, vorrei ringraziare la Prof. Ombretta Gaggi per la disponibilità e l'aiuto fornitomi nel redarre questo documento, oltre al supporto durante il periodo di stage.

Ringrazio con affetto la mia famiglia per il sostegno, il supporto e gli sforzi fatti durante questi tre anni di studio.

Ringrazio i miei amici per i bellissimi momenti passati insieme durante questi anni universitari.

Desidero infine ringraziare il mio tutor aziendale Roberto Martina per il tempo dedicatomi durante lo stage e per la conoscenza trasmessami con tanta passione e metodo.

Padova, Settembre 2023

Michael Amista'

Indice

1	Introduzione	1
1.1	L'azienda	1
1.2	Aspettative personali	2
1.3	Organizzazione del testo	2
2	Descrizione dello stage	4
2.1	Il problema	4
2.2	Il progetto	4
2.3	Vincoli del progetto	5
2.3.1	Vincoli temporali	5
2.3.2	Vincoli metodologici	6
2.3.3	Vincoli tecnologici	6
3	Analisi dei requisiti	7
3.1	Casi d'uso	7
3.1.1	UC0: Menu	8
3.1.2	UC1: Crea-Modifica progetto	9
3.1.3	UC2: Tabella progetti	11
3.1.4	UC2.2: Visualizzazione lista progetti	13
3.1.5	UC2.3: Visualizzazione progetto in dettaglio	14
3.1.6	UC3: Tabella progetti aperti	16
3.1.7	UC4: Tabella clienti	17
3.1.8	UC4.2: Visualizzazione lista clienti	18
3.1.9	UC4.3: Visualizzazione cliente in dettaglio	19
3.1.10	UC4.3.3: Visualizzazione offerte correlate	20
3.1.11	UC4.3.3.1: Crea-Modifica offerta	21
3.1.12	UC5: Tabella offerte	22
3.1.13	UC5.2: Visualizzazione lista offerte	23
3.1.14	UC5.3: Visualizzazione offerta in dettaglio	24
3.2	Definizione e tracciamento dei requisiti	26
3.2.1	Requisiti funzionali	27
3.2.2	Requisiti di vincolo	30
4	Strumenti utilizzati	31
4.1	Ambiente di lavoro	31
4.2	Framework	32
4.2.1	Spring (core)	32
4.2.2	Hibernate	33

4.2.3	Bootstrap	33
4.2.4	Apache Struts	34
4.3	Linguaggi utilizzati	35
4.3.1	Java	35
4.3.2	HTML5, CSS3 e JavaScript	35
4.4	Librerie a supporto della codifica	36
4.4.1	Apache Commons	36
4.4.2	JSTL	36
4.5	IDE	37
5	Progettazione	38
5.1	Approccio MDA	38
5.2	Architettura MVC	41
6	Prodotto finale	44
6.1	Pagina 1 della webapp	44
6.2	Pagina 2 della webapp	44
6.3	Pagina 3 della webapp	44
6.4	Pagina 4 della webapp	44
6.5	Pagina 5 della webapp	44
6.6	Pagina N della webapp	44
7	Conclusioni	45
7.1	Test e validazione	45
7.2	Resoconto dello stage	45
	Sitografia	47
	Glossario	48

Elenco delle figure

1.1	Logo dell'azienda	1
3.1	UC0: Menu	8
3.2	UC1: Crea-Modifica progetto	9
3.3	UC2: Tabella progetti	11
3.4	UC2.2: Visualizzazione lista progetti	13
3.5	UC2.3: Visualizzazione progetto in dettaglio	14
3.6	UC3: Tabella progetti aperti	16
3.7	UC4: Tabella clienti	17
3.8	UC4.2: Visualizzazione lista clienti	18
3.9	UC4.3: Visualizzazione cliente in dettaglio	19
3.10	UC4.3.3: Visualizzazione offerte correlate	20
3.11	UC4.3.3.1: Crea-Modifica offerta	21
3.12	UC5: Tabella offerte	22
3.13	UC5.2: Visualizzazione lista offerte	23
3.14	UC5.3: Visualizzazione offerta in dettaglio	24
4.1	Logo di Spring	32
4.2	Spring - core container	32
4.3	Logo di Hibernate	33
4.4	Logo di Bootstrap	33
4.5	Logo di Apache Struts	34
4.6	Logo di Java	35
4.7	Logo dello stack tecnologico del front-end	35
4.8	Logo di Apache Commons	36
4.9	Logo di Eclipse	37
5.1	Architettura MVC di Apache Struts	41

Elenco delle tabelle

2.1	Pianificazione delle attività	5
3.1	Requisiti funzionali	30
3.2	Requisiti di vincolo	30

Capitolo 1

Introduzione

Il seguente capitolo ha la funzione di introdurre l'azienda ospitante presso la quale è stato svolto lo stage. Vengono inoltre elencate quelle che sono le aspettative personali principalmente riguardo la crescita tecnica e professionale.

1.1 L'azienda



Figura 1.1: Logo dell'azienda

CWBI (Codice Web Banking Innovation) è un'azienda italiana che opera nel mercato dell'Information Communication Technology e supporta i propri clienti nello studio dei modelli di business, nella definizione dei processi organizzativi e nella progettazione e realizzazione di software con un forte orientamento alle nuove tecnologie.

Fondata a Padova nell'anno 2013, CWBI ha fidelizzato rapporti di collaborazione con aziende nazionali di primaria importanza attraverso la sua struttura interna costituita da professionisti con skills elevate, che negli anni hanno maturato un forte know-how in diversi settori di business quali: Banking, Media and Publishing, Insurance, Industry.

Anni di esperienza permettono all'azienda di affrontare con successo ogni singolo aspetto del ciclo di vita dei progetti nei quali è coinvolta; entusiasmo e visione strategica, accompagnati da un forte orientamento al risultato, sono il motore della sua capacità innovativa.

CWBI offre una vasta gamma di servizi, tra cui:

- sviluppo di applicazioni e portali web-based;

- sviluppo di applicazioni mobile;
- studio di fattibilità e sostenibilità dei modelli di business;
- analisi e definizione dei processi organizzativi;
- studi di navigabilità e usabilità;
- studi di ergonomia del software.

1.2 Aspettative personali

L'obiettivo dell'attività di stage, oltre lo sviluppo del progetto commissionato dall'azienda ospitante, è quello di crescere personalmente e professionalmente tramite un'impronta aziendale caratterizzata da competenza tecnica e professionale.

Di seguito vengono riportati gli obiettivi personali a titolo professionale e formativo da raggiungere:

- apprendimento di Java;
- apprendimento dei framework Spring e Hibernate;
- apprendimento di HTML5/CSS3 e del framework Bootstrap;
- apprendimento JavaScript e della libreria jQuery;
- apprendimento del sistema di controllo di versione aziendale;
- apprendimento dei processi aziendali;
- apprendimento degli strumenti per la gestione di progetto;
- apprendimento dell'ambiente di sviluppo;
- studio di fattibilità del progetto e realizzazione dello stesso le tecnologie aziendali;
- capacità di trovare soluzioni alternative da quelle proposte.

1.3 Organizzazione del testo

Il [secondo capitolo](#) descrive l'attività di stage definendo il problema da affrontare e i vincoli da rispettare lungo lo sviluppo.

Il [terzo capitolo](#) approfondisce l'analisi dei requisiti effettuata, elencando i casi d'uso raccolti e definendo i rispettivi requisiti.

Il [quarto capitolo](#) approfondisce gli strumenti e le tecnologie utilizzate nello sviluppo del prodotto commissionato e che si pongono come vincolo di progetto.

Il [quinto capitolo](#) approfondisce la progettazione, caratterizzata dall'architettura del sistema e dai design pattern utilizzati.

Il [sesto capitolo](#) illustra il prodotto finale in tutte le sue componenti.

Nel [settimo capitolo](#) si possono trovare le conclusioni sul lavoro svolto e formate dai test effettuati e da un resoconto finale che pone a confronto le aspettative iniziali con quelli che sono stati i traguardi raggiunti al termine dell'esperienza di stage.

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: parola^[g];
- per la bibliografia e sitografia riguardo un certo argomento viene utilizzata la seguente nomenclatura: argomento^[b].

Capitolo 2

Descrizione dello stage

Il seguente capitolo ha la funzione di introdurre l'attività di stage definendo quella che è la richiesta aziendale e i vincoli di progetto da rispettare nello sviluppo dell'applicativo commissionato.

2.1 Il problema

L'azienda CWBI realizza prodotti software su commissione diretta dei clienti, oltre ad applicativi pensati per raggiungere nuovi acquirenti. L'azienda necessita di una metodologia efficiente per tenere traccia dei contatti con i clienti e delle offerte formulate per uno o più progetti di interesse. Attualmente la soluzione aziendale si basa su una cartella in rete dove sono contenuti tutti i documenti e i contatti con i clienti negli ultimi 10 anni di attività.

Ciò che l'azienda desidera è un applicativo web che consenta di creare, organizzare e tenere traccia di queste informazioni in modo da creare un ambiente di lavoro/tracciamento più efficiente ed efficace.

2.2 Il progetto

L'obiettivo dello stage riguarda quindi lo sviluppo di un modulo web per la webapp aziendale "CW GEST" preesistente. CW GEST è una webapp utilizzata dai dipendenti dell'area commerciale di CWBI e composta da diversi moduli, ciascuno dei quali assolve differenti compiti riguardo la gestione interna aziendale. In particolare ciò che deve essere integrato in CW GEST è un sistema di gestione delle campagne a supporto dell'area marketing nel contesto aziendale, ovvero l'ambiente bancario/fintech.

In particolare, il flusso principale risulta essere il seguente:

1. contatto con il cliente e raccolta preliminare delle informazioni (contatti, idee, budget, ecc.);
2. definizione del progetto da realizzare;
3. creazione dell'offerta;
4. proposta dell'offerta formulata e eventuale contrattazione;
5. approvazione o rifiuto dell'offerta.

La webapp sarà utilizzata dai dipendenti dell'area commerciale di CWBI e dovrà consentire di aggiungere, visualizzare e aggiornare la lista completa dei propri clienti, con annessi i progetti richiesti e le offerte formulate per essi.

2.3 Vincoli del progetto

2.3.1 Vincoli temporali

Come citato nel regolamento delle attività di stage il tempo limite delle attività si colloca tra le 300 e le 320 ore; motivo per il quale questo risulta essere il tempo limite nel realizzare il prodotto commissionato. È stato quindi redatto un piano di lavoro, tenente conto del limite massimo consentito, che organizza le attività in 8 settimane individualmente composte da 40 ore lavorative.

Si riporta di seguito la pianificazione delle attività:

Durata	Tipo di attività
5gg (40h)	Formazione iniziale e introduzione tecnologie utilizzata JAVA/-JEE.
5gg (40h)	Formazione soluzione “baseapp” con apprendimento “framework” di lavoro.
8,5gg (68h)	Analisi e raccolta requisiti.
15gg (122h)	Sviluppo soluzione (Realizzazione soluzione software in java back-end e sviluppo front- end).
2gg (15h)	Test e supporto UAT.
2gg (15h)	Documentazione progetto.
37,5gg (300h)	TOTALE

Tabella 2.1: Pianificazione delle attività

2.3.2 Vincoli metodologici

Per il versionamento del codice l'azienda ha imposto l'utilizzo di TortoiseSVN, un servizio di repository da loro utilizzato.

2.3.3 Vincoli tecnologici

Al fine di raggiungere gli obiettivi prefissati l'azienda ha imposto i seguenti vincoli tecnologici:

- utilizzo di Java e dei framework Spring MVC e Hibernate per lo sviluppo lato server;
- utilizzo del framework Bootstrap, JSTL e della libreria jQuery per lo sviluppo lato client;
- utilizzo di HTML5, CSS3 e JavaScript come supporto allo sviluppo lato client.

Non sono stati imposti limiti sull'utilizzo di eventuali tecnologie aggiuntive.

Capitolo 3

Analisi dei requisiti

Il seguente capitolo assume la funzione di illustrare i casi d'uso e i requisiti raccolti nella fase di analisi del progetto commissionato.

3.1 Casi d'uso

Per poter capire e studiare a fondo tutte le funzionalità che devono essere messe a disposizione dell'utente che utilizza l'applicativo da sviluppare, sono stati realizzati i relativi diagrammi dei casi d'uso di tipo [UML](#)^[g]. Tali diagrammi sono risultati fondamentali per individuare correttamente tutti i requisiti del sistema in questione.

Ciascun caso d'uso è costituito da:

- attore primario;
- precondizione;
- postcondizione;
- scenario principale;
- generalizzazioni (qualora fossero presenti);
- estensioni (qualora fossero presenti).

I casi d'uso identificati dalla sigla "UCE" rappresentano un caso d'uso d'errore. L'unico attore coinvolto e identificato come *Utente autenticato* corrisponde ad un qualunque dipendente dell'area commerciale di CWBI, al quale sono stati assegnati i permessi necessari a svolgere le operazioni di seguito riportate.

3.1.1 UC0: Menu

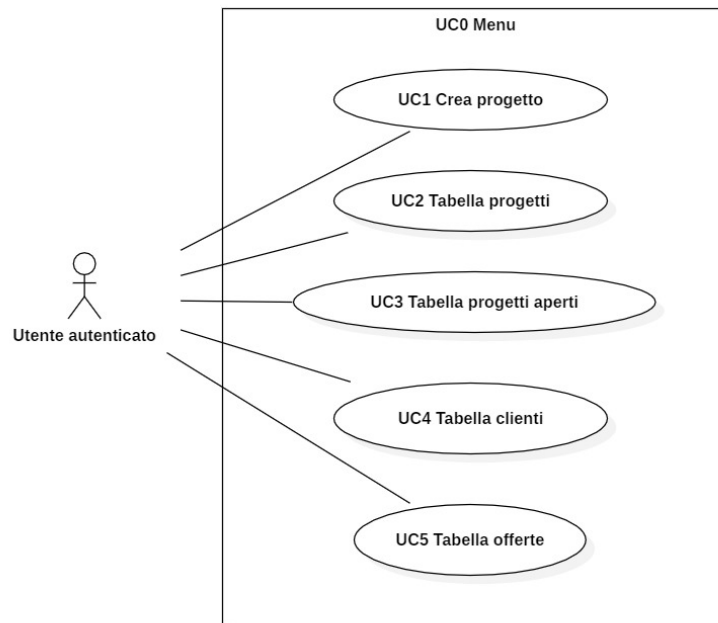


Figura 3.1: UC0: Menu

- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha accesso al modulo *Campaign* della webapp CW GEST.
- **Postcondizione:** Il sistema è pronto all'uso.
- **Scenario principale:**
 1. L'attore crea un nuovo progetto [UC1].
 2. L'attore accede alla tabella dei progetti [UC2].
 3. L'attore accede alla tabella dei progetti aperti [UC3].
 4. L'attore accede alla tabella dei clienti [UC4].
 5. L'attore accede alla tabella delle offerte [UC5].

3.1.2 UC1: Crea-Modifica progetto

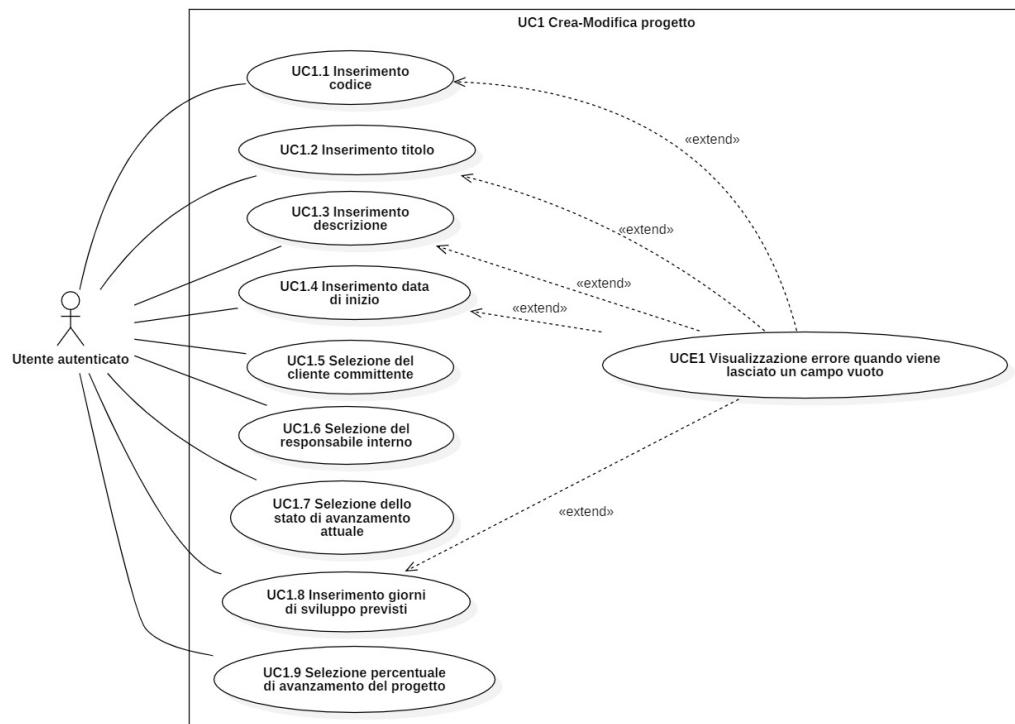


Figura 3.2: UC1: Crea-Modifica progetto

- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha selezionato:
 - l'opzione *nuovo progetto* dal menu;
 - l'opzione *modifica* dal dettaglio di un progetto
- **Postcondizione:** L'attore ha creato/modificato un progetto.
- **Scenario principale:**
 1. L'attore inserisce/modifica il codice del progetto [UC1.1].
 2. L'attore inserisce/modifica il titolo del progetto [UC1.2].
 3. L'attore inserisce/modifica la descrizione del progetto [UC1.3].
 4. L'attore inserisce/modifica la data di inizio del progetto [UC1.4].
 5. L'attore seleziona/modifica il cliente committente [UC1.5].
 6. L'attore seleziona/modifica il responsabile interno assegnato al progetto [UC1.6].
 7. L'attore seleziona/modifica lo stato di avanzamento attuale delle attività [UC1.7].

8. L'attore inserisce/modifica i giorni di sviluppo previsti [UC1.8].
9. L'attore seleziona/modifica la percentuale di avanzamento delle attività di progetto [UC1.9].

- **Estensioni:**

- L'attore visualizza un messaggio di errore quando viene lasciato un campo vuoto [UCE1].

3.1.3 UC2: Tabella progetti



Figura 3.3: UC2: Tabella progetti

- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha selezionato l'opzione *tabella progetti* dal menu.
- **Postcondizione:** L'attore ha accesso alla tabella dei progetti.
- **Scenario principale:**
 1. L'attore cerca uno o più progetti tramite la funzionalità di ricerca [UC2.1].
 2. L'attore visualizza la lista completa dei progetti presenti nel sistema [UC2.2].
 3. L'attore visualizza un progetto nel dettaglio [UC2.3].
- **Generalizzazioni:**
 1. L'attore ricerca un progetto per ID [UC2.1.1].

2. L'attore ricerca un progetto per uno specifico cliente associato [UC2.1.2].
3. L'attore ricerca un progetto per responsabile interno [UC2.1.3].
4. L'attore ricerca un progetto per codice [UC2.1.4].
5. L'attore ricerca un progetto per titolo [UC2.1.5].
6. L'attore ricerca un progetto per stato di avanzamento [UC2.1.6].
7. L'attore ricerca un progetto per data di creazione [UC2.1.7].

3.1.4 UC2.2: Visualizzazione lista progetti

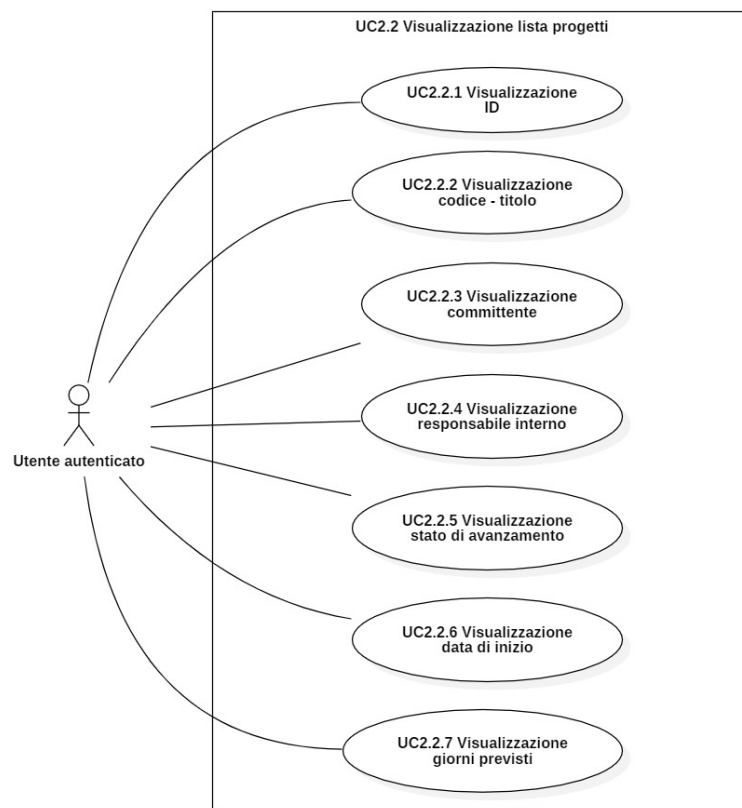


Figura 3.4: UC2.2: Visualizzazione lista progetti

- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha accesso alla tabella dei progetti.
- **Postcondizione:** L'attore ha visualizzato la lista dei progetti.
- **Scenario principale:**
 1. L'attore visualizza l'ID del progetto [UC2.2.1].
 2. L'attore visualizza la stringa "codice - titolo" del progetto [UC2.2.2].
 3. L'attore visualizza il committente del progetto [UC2.2.3].
 4. L'attore visualizza il responsabile interno del progetto [UC2.2.4].
 5. L'attore visualizza lo stato di avanzamento del progetto [UC2.2.5].
 6. L'attore visualizza la data di inizio del progetto [UC2.2.6].
 7. L'attore visualizza i giorni di sviluppo previsti del progetto [UC2.2.7].

3.1.5 UC2.3: Visualizzazione progetto in dettaglio

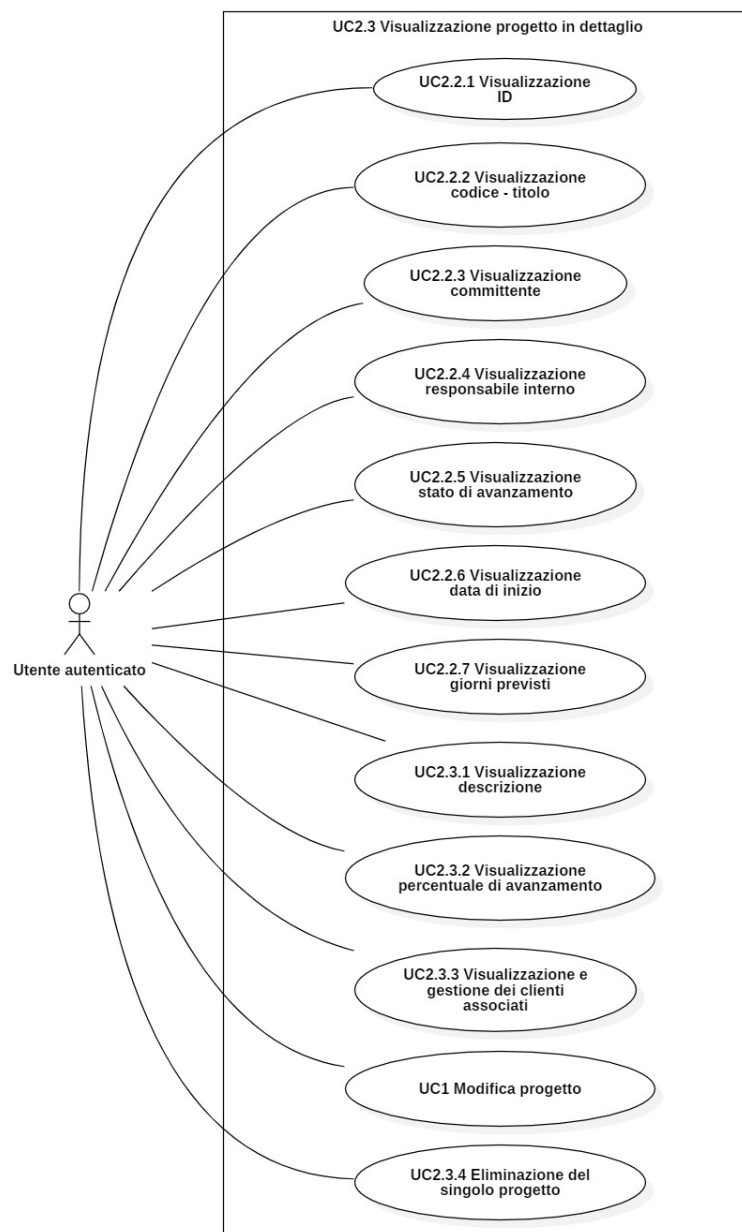


Figura 3.5: UC2.3: Visualizzazione progetto in dettaglio

- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha accesso alla tabella dei progetti.
- **Postcondizione:** L'attore ha visualizzato il dettaglio di un progetto.

- **Scenario principale:**

1. L'attore visualizza l'ID del progetto [UC2.2.1].
2. L'attore visualizza la stringa "codice - titolo" del progetto [UC2.2.2].
3. L'attore visualizza il committente del progetto [UC2.2.3].
4. L'attore visualizza il responsabile interno del progetto [UC2.2.4].
5. L'attore visualizza lo stato di avanzamento del progetto [UC2.2.5].
6. L'attore visualizza la data di inizio del progetto [UC2.2.6].
7. L'attore visualizza i giorni di sviluppo previsti del progetto [UC2.2.7].
8. L'attore visualizza la descrizione del progetto [UC2.3.1].
9. L'attore visualizza la percentuale di avanzamento del progetto [UC2.3.2].
10. L'attore visualizza e gestisce i clienti associati al progetto [UC2.3.3].
11. L'attore può modificare il progetto [UC1].
12. L'attore può eliminare il progetto [UC2.3.4].

3.1.6 UC3: Tabella progetti aperti

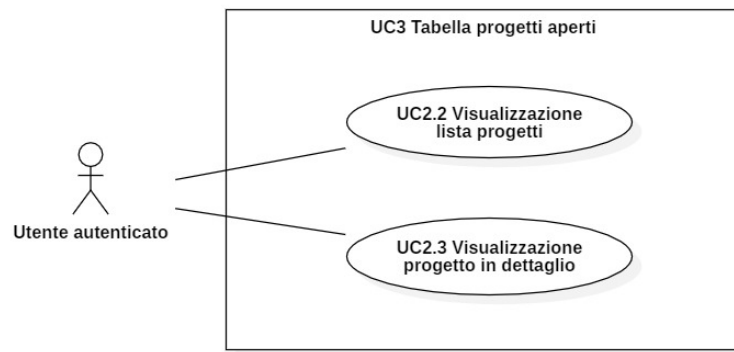


Figura 3.6: UC3: Tabella progetti aperti

- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha selezionato l'opzione *tabella progetti aperti* dal menu.
- **Postcondizione:** L'attore ha accesso alla tabella dei progetti aperti.
- **Scenario principale:**
 1. L'attore visualizza la lista dei progetti aperti presenti nel sistema [UC2.2].
 2. L'attore visualizza un progetto aperto nel dettaglio [UC2.3].

3.1.7 UC4: Tabella clienti

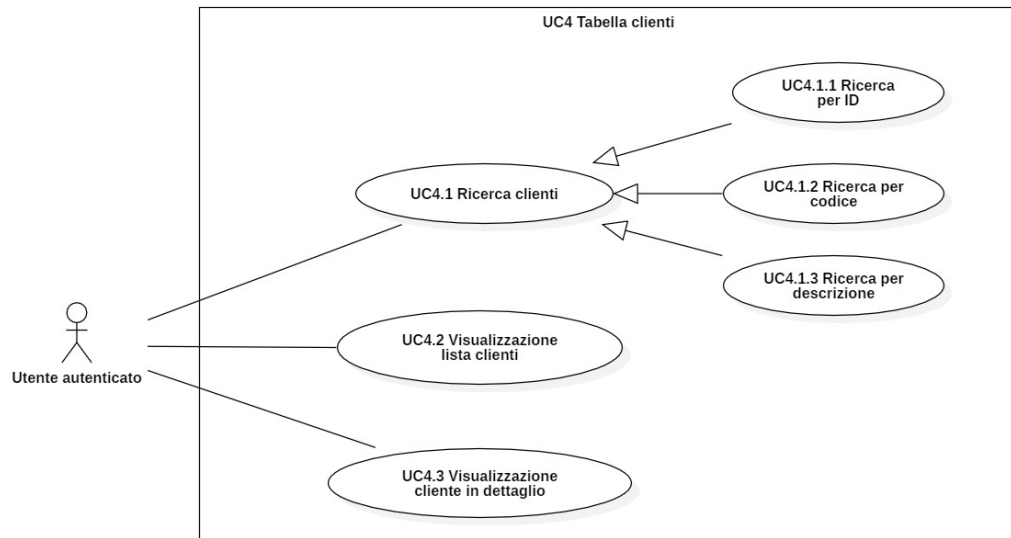


Figura 3.7: UC4: Tabella clienti

- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha selezionato l'opzione *tabella clienti* dal menu.
- **Postcondizione:** L'attore ha accesso alla tabella dei clienti.
- **Scenario principale:**
 1. L'attore cerca uno o più clienti tramite la funzionalità di ricerca [UC4.1].
 2. L'attore visualizza la lista completa dei clienti presenti nel sistema [UC4.2].
 3. L'attore visualizza un cliente nel dettaglio [UC4.3].
- **Generalizzazioni:**
 1. L'attore ricerca un cliente per ID [UC4.1.1].
 2. L'attore ricerca un cliente per codice (nominativo aziendale) [UC4.1.2].
 3. L'attore ricerca un cliente per descrizione [UC4.1.3].

3.1.8 UC4.2: Visualizzazione lista clienti

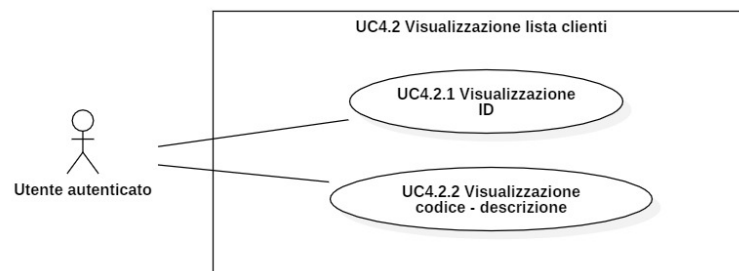


Figura 3.8: UC4.2: Visualizzazione lista clienti

- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha accesso alla tabella dei clienti.
- **Postcondizione:** L'attore ha visualizzato la lista dei clienti.
- **Scenario principale:**
 1. L'attore visualizza l'ID del cliente [UC4.2.1].
 2. L'attore visualizza la stringa "codice - descrizione" del cliente [UC4.2.2].

3.1.9 UC4.3: Visualizzazione cliente in dettaglio

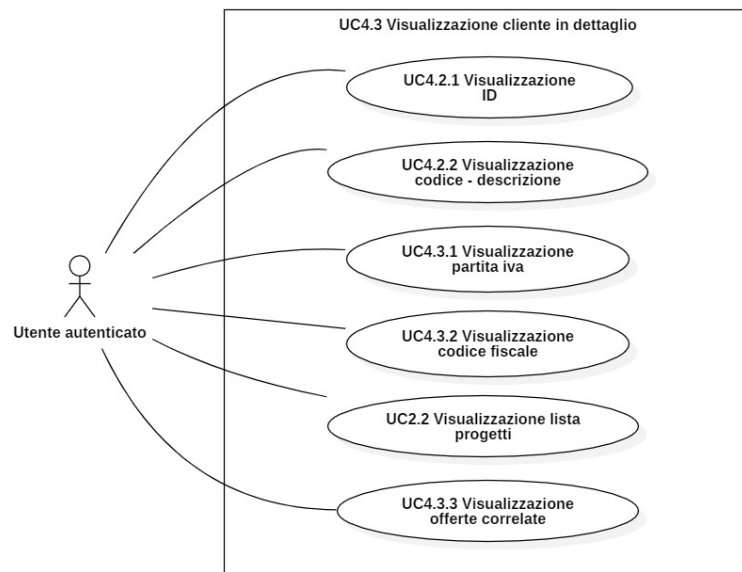
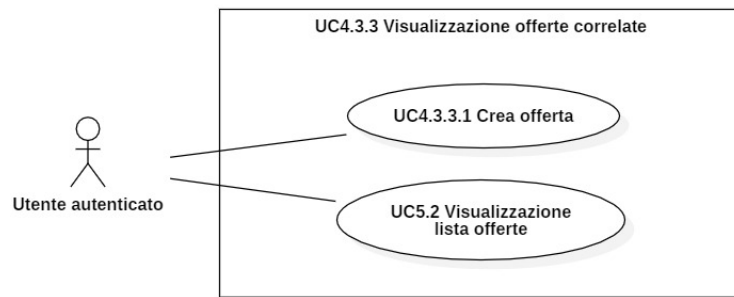


Figura 3.9: UC4.3: Visualizzazione cliente in dettaglio

- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha accesso alla tabella dei clienti.
- **Postcondizione:** L'attore ha visualizzato il dettaglio di un cliente.
- **Scenario principale:**
 1. L'attore visualizza l'ID del cliente [UC4.2.1].
 2. L'attore visualizza la stringa "codice - descrizione" del cliente [UC4.2.2].
 3. L'attore visualizza la partita iva del cliente [UC4.3.1].
 4. L'attore visualizza il codice fiscale del cliente [UC4.3.2].
 5. L'attore visualizza la lista dei progetti associati al cliente [UC2.2].
 6. L'attore visualizza le offerte correlate al cliente [UC4.3.3].

3.1.10 UC4.3.3: Visualizzazione offerte correlate**Figura 3.10:** UC4.3.3: Visualizzazione offerte correlate

- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha visualizzato il dettaglio di un cliente.
- **Postcondizione:** L'attore ha visualizzato le offerte correlate al cliente.
- **Scenario principale:**
 1. L'attore può creare una nuova offerta correlata al cliente [UC4.3.3.1].
 2. L'attore visualizza la lista delle offerte del cliente [UC5.2].

3.1.11 UC4.3.3.1: Crea-Modifica offerta

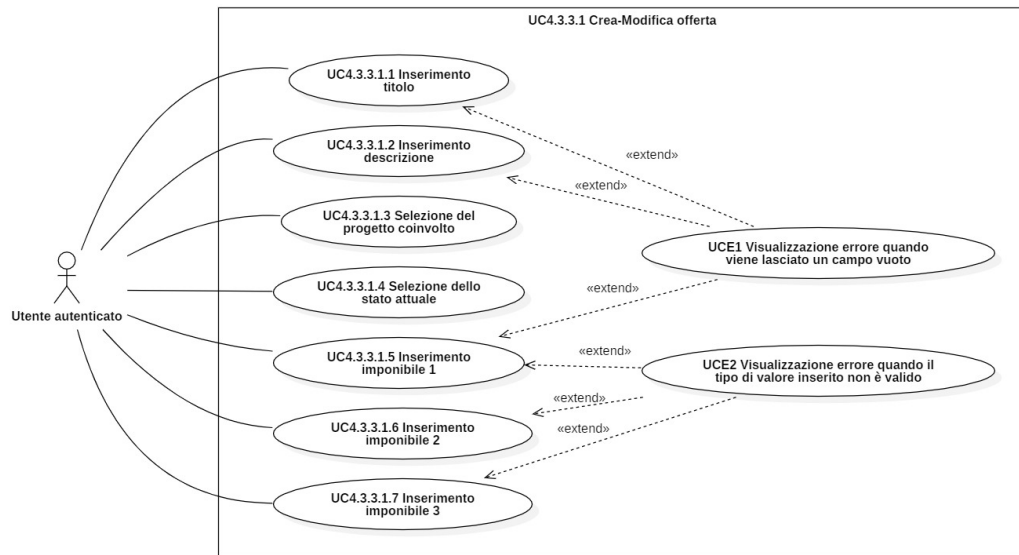


Figura 3.11: UC4.3.3.1: Crea-Modifica offerta

- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha selezionato:
 - l'opzione *nuova offerta* dal dettaglio di un cliente;
 - l'opzione *modifica* dal dettaglio di un'offerta.
- **Postcondizione:** L'attore ha creato/modificato un'offerta.
- **Scenario principale:**
 1. L'attore inserisce/modifica il titolo di un'offerta [UC4.3.3.1.1].
 2. L'attore inserisce/modifica la descrizione di un'offerta [UC4.3.3.1.2].
 3. L'attore seleziona/modifica il progetto di riferimento di un'offerta [UC4.3.3.1.3].
 4. L'attore seleziona/modifica lo stato attuale di un'offerta [UC4.3.3.1.4].
 5. L'attore inserisce/modifica l'imponibile 1 di un'offerta [UC4.3.3.1.5].
 6. L'attore inserisce/modifica l'imponibile 2 di un'offerta [UC4.3.3.1.6].
 7. L'attore inserisce/modifica l'imponibile 3 di un'offerta [UC4.3.3.1.7].
- **Estensioni:**
 - L'attore visualizza un messaggio di errore quando viene lasciato un campo vuoto [UCE1].
 - L'attore visualizza un messaggio di errore quando viene inserito un tipo di valore non compatibile con quello del campo di interesse [UCE2].

3.1.12 UC5: Tabella offerte

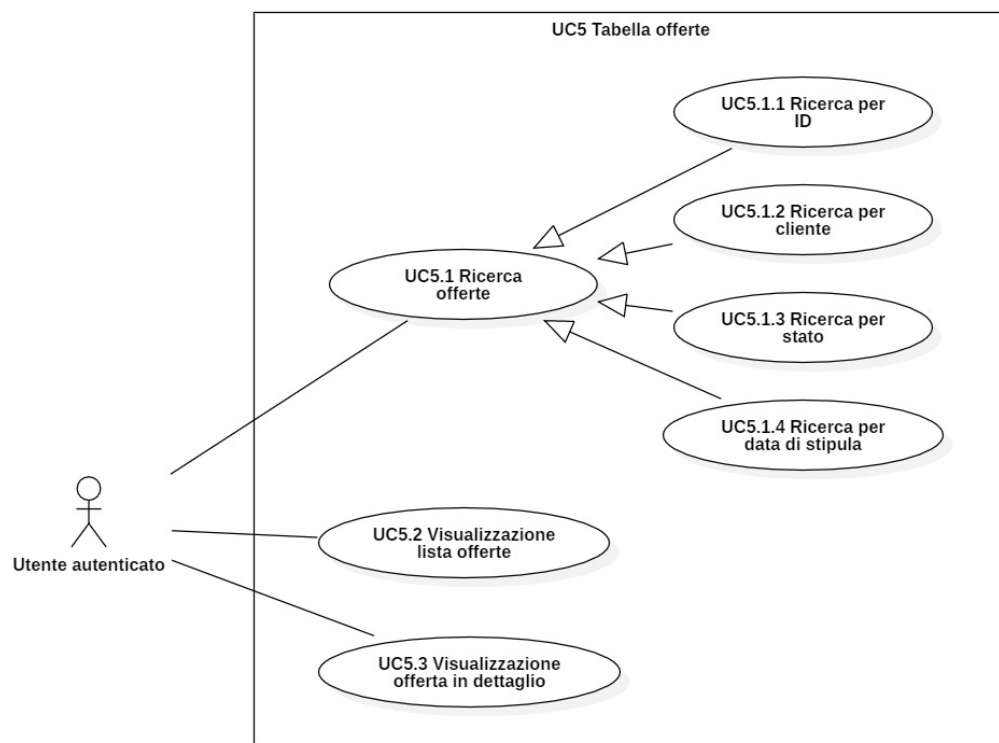
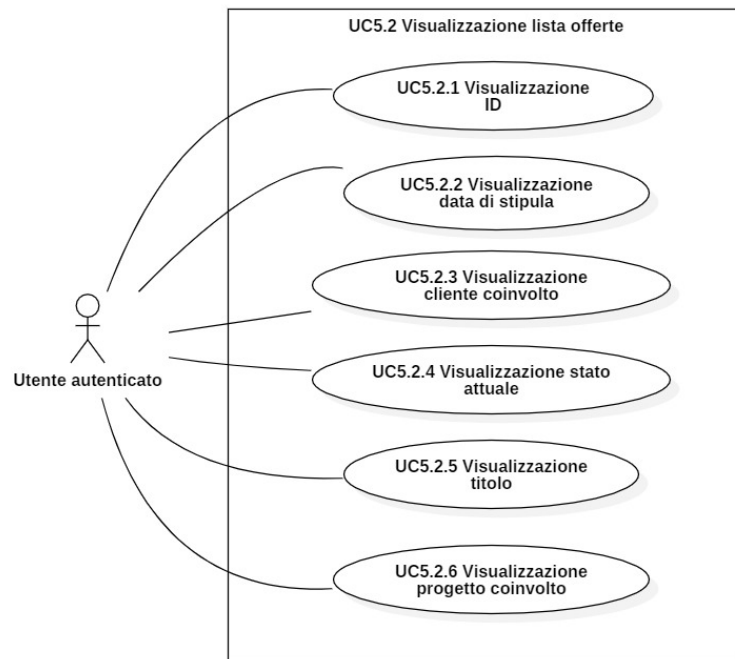
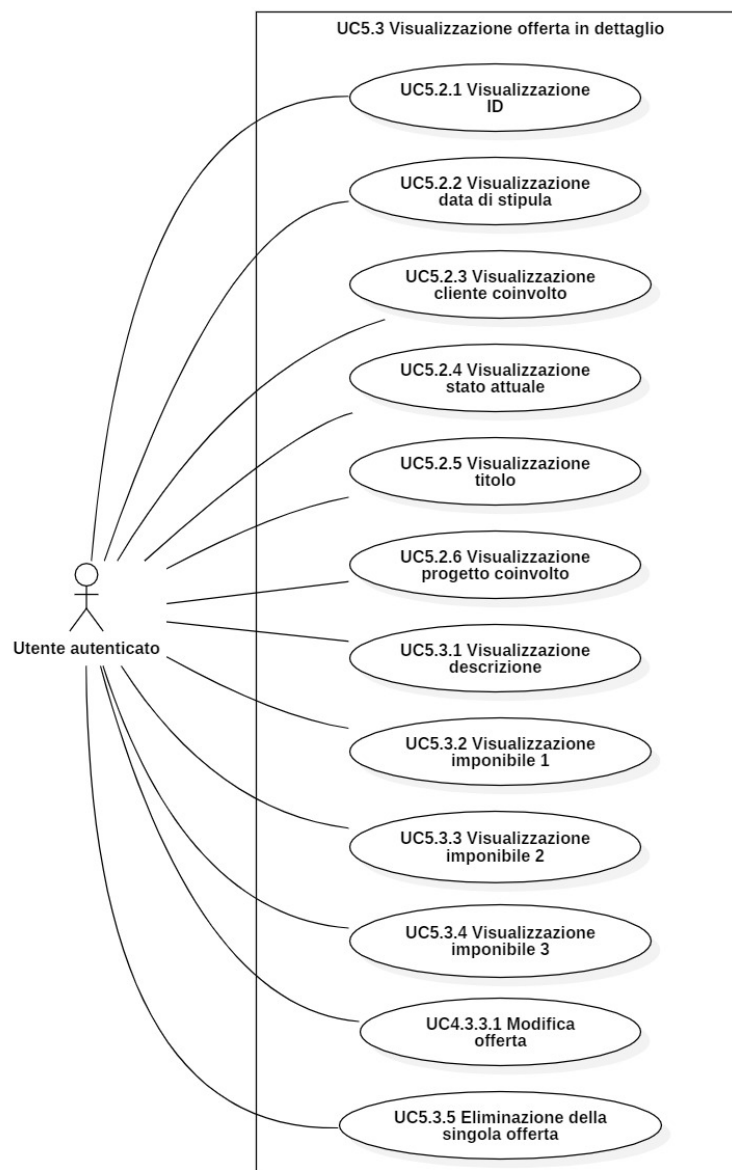


Figura 3.12: UC5: Tabella offerte

- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha selezionato l'opzione *tabella offerte* dal menu.
- **Postcondizione:** L'attore ha accesso alla tabella delle offerte.
- **Scenario principale:**
 1. L'attore cerca una o più offerte tramite la funzionalità di ricerca [UC5.1].
 2. L'attore visualizza la lista completa delle offerte presenti nel sistema [UC5.2].
 3. L'attore visualizza un'offerta nel dettaglio [UC5.3].
- **Generalizzazioni:**
 1. L'attore ricerca un'offerta per ID [UC5.1.1].
 2. L'attore ricerca un'offerta per cliente associato [UC5.1.2].
 3. L'attore ricerca un'offerta per stato attuale [UC5.1.3].
 4. L'attore ricerca un'offerta per data di stipula [UC5.1.4].

3.1.13 UC5.2: Visualizzazione lista offerte**Figura 3.13:** UC5.2: Visualizzazione lista offerte

- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha accesso alla tabella delle offerte.
- **Postcondizione:** L'attore ha visualizzato la lista delle offerte.
- **Scenario principale:**
 1. L'attore visualizza l'ID dell'offerta [UC5.2.1].
 2. L'attore visualizza la data di stipula dell'offerta [UC5.2.2].
 3. L'attore visualizza il cliente coinvolto nell'offerta [UC5.2.3].
 4. L'attore visualizza lo stato attuale dell'offerta [UC5.2.4].
 5. L'attore visualizza il titolo dell'offerta [UC5.2.5].
 6. L'attore visualizza il progetto coinvolto nell'offerta [UC5.2.6].

3.1.14 UC5.3: Visualizzazione offerta in dettaglio**Figura 3.14:** UC5.3: Visualizzazione offerta in dettaglio

- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha accesso alla tabella delle offerte.
- **Postcondizione:** L'attore ha visualizzato il dettaglio di un'offerta.
- **Scenario principale:**

1. L'attore visualizza l'ID dell'offerta [UC5.2.1].
2. L'attore visualizza la data di stipula dell'offerta [UC5.2.2].
3. L'attore visualizza il cliente coinvolto nell'offerta [UC5.2.3].
4. L'attore visualizza lo stato attuale dell'offerta [UC5.2.4].
5. L'attore visualizza il titolo dell'offerta [UC5.2.5].
6. L'attore visualizza il progetto coinvolto nell'offerta [UC5.2.6].
7. L'attore visualizza la descrizione dell'offerta [UC5.3.1].
8. L'attore visualizza l'imponibile 1 dell'offerta [UC5.3.2].
9. L'attore visualizza l'imponibile 2 dell'offerta [UC5.3.3].
10. L'attore visualizza l'imponibile 3 dell'offerta [UC5.3.4].
11. L'attore può modificare l'offerta [UC4.3.3.1].
12. L'attore può eliminare l'offerta [UC5.3.5].

3.2 Definizione e tracciamento dei requisiti

Per concordare gli obiettivi da raggiungere e i problemi da risolvere sono stati trovati una serie di requisiti. Per identificare ciascun [requisito](#)^[8] si è ricorso ad un codice identificativo composto **R** [**F/V**] **O** **N** dove:

- **R** = requisito;
- **F** = funzionale;
- **V** = di vincolo;
- **O** = obbligatorio;
- **N** = numero del requisito.

Di seguito si riporta una tabella contenente tutti i requisiti individuati con relativa descrizione e riferimento al relativo use case.

3.2.1 Requisiti funzionali

Identificativo	Descrizione	Use case
RFO1	L'utente deve poter utilizzare il menu.	UC0
RFO2	L'utente deve poter creare un nuovo progetto.	UC1
RFO3	L'utente deve poter accedere alla tabella dei progetti.	UC2
RFO4	L'utente deve poter accedere alla tabella dei progetti aperti.	UC3
RFO5	L'utente deve poter accedere alla tabella dei clienti.	UC4
RFO6	L'utente deve poter accedere alla tabella delle offerte.	UC5
RFO7	L'utente inserisce il codice del progetto.	UC1.1
RFO8	L'utente inserisce il titolo del progetto.	UC1.2
RFO9	L'utente inserisce la descrizione del progetto.	UC1.3
RFO10	L'utente inserisce la data di inizio del progetto.	UC1.4
RFO11	L'utente seleziona il cliente committente del progetto.	UC1.5
RFO12	L'utente seleziona il responsabile interno del progetto.	UC1.6
RFO13	L'utente seleziona lo stato di avanzamento del progetto.	UC1.7
RFO14	L'utente inserisce i giorni di sviluppo previsti.	UC1.8
RFO15	L'utente seleziona la percentuale di avanzamento del progetto.	UC1.9
RFO16	L'utente visualizza un errore quando lascia vuoto il campo codice del progetto.	UCE1
RFO17	L'utente visualizza un errore quando lascia vuoto il campo titolo del progetto.	UCE1
RFO18	L'utente visualizza un errore quando lascia vuoto il campo descrizione del progetto.	UCE1
RFO19	L'utente visualizza un errore quando lascia vuoto il campo data di inizio del progetto.	UCE1
RFO20	L'utente visualizza un errore quando lascia vuoto il campo giorni di sviluppo del progetto.	UCE1
RFO21	L'utente può ricercare uno o più progetti.	UC2.1
RFO22	L'utente ricerca un progetto per ID.	UC2.1.1
RFO23	L'utente ricerca un progetto per cliente associato.	UC2.1.2
RFO24	L'utente ricerca un progetto per responsabile interno.	UC2.1.3
RFO25	L'utente ricerca un progetto per codice di progetto.	UC2.1.4
RFO26	L'utente ricerca un progetto per titolo.	UC2.1.5

Identificativo	Descrizione	Use case
RFO27	L'utente ricerca un progetto per stato di avanzamento.	UC2.1.6
RFO28	L'utente ricerca un progetto per data di creazione.	UC2.1.7
RFO29	L'utente visualizza la lista dei progetti.	UC2.2
RFO30	L'utente visualizza il dettaglio di un progetto.	UC2.3
RFO31	L'utente visualizza l'ID di un progetto.	UC2.2.1
RFO32	L'utente visualizza la stringa "codice - titolo" di un progetto.	UC2.2.2
RFO33	L'utente visualizza il committente di un progetto.	UC2.2.3
RFO34	L'utente visualizza il responsabile interno di un progetto.	UC2.2.4
RFO35	L'utente visualizza lo stato di avanzamento di un progetto.	UC2.2.5
RFO36	L'utente visualizza la data di inizio di un progetto.	UC2.2.6
RFO37	L'utente visualizza i giorni di sviluppo previsti di un progetto.	UC2.2.7
RFO38	L'utente visualizza la descrizione del progetto.	UC2.3.1
RFO39	L'utente visualizza la percentuale di avanzamento del progetto.	UC2.3.2
RFO40	L'utente visualizza e gestisce i clienti del progetto.	UC2.3.3
RFO41	L'utente modifica il progetto.	UC1
RFO42	L'utente elimina il progetto.	UC2.3.4
RFO43	L'utente ricerca uno o più clienti.	UC4.1
RFO44	L'utente ricerca un cliente per ID.	UC4.1.1
RFO45	L'utente ricerca un cliente per codice.	UC4.1.2
RFO46	L'utente ricerca un cliente per descrizione.	UC4.1.3
RFO47	L'utente visualizza la lista dei clienti.	UC4.2
RFO48	L'utente visualizza il dettaglio di un cliente.	UC4.3
RFO49	L'utente visualizza l'ID di un cliente.	UC4.2.1
RFO50	L'utente visualizza la stringa "codice - descrizione" di un cliente.	UC4.2.2
RFO51	L'utente visualizza partita iva del cliente.	UC4.3.1
RFO52	L'utente visualizza il codice fiscale del cliente.	UC4.3.2
RFO53	L'utente visualizza la lista dei progetti associati al cliente.	UC2.2

Identificativo	Descrizione	Use case
RFO54	L'utente visualizza la lista delle offerte associate al cliente.	UC4.3.3
RFO55	L'utente crea una nuova offerta.	UC4.3.3.1
RFO56	L'utente visualizza la lista delle offerte.	UC5.2
RFO57	L'utente inserisce il titolo dell'offerta.	UC4.3.3.1.1
RFO58	L'utente inserisce la descrizione dell'offerta.	UC4.3.3.1.2
RFO59	L'utente seleziona il progetto coinvolto nell'offerta.	UC4.3.3.1.3
RFO60	L'utente seleziona lo stato attuale dell'offerta.	UC4.3.3.1.4
RFO61	L'utente inserisce l'imponibile 1 dell'offerta.	UC4.3.3.1.5
RFO62	L'utente inserisce l'imponibile 2 dell'offerta.	UC4.3.3.1.6
RFO63	L'utente inserisce l'imponibile 3 dell'offerta.	UC4.3.3.1.7
RFO64	L'utente visualizza un errore quando lascia vuoto il campo titolo dell'offerta.	UCE1
RFO65	L'utente visualizza un errore quando lascia vuoto il campo descrizione dell'offerta.	UCE1
RFO66	L'utente visualizza un errore quando lascia vuoto il campo imponibile 1 dell'offerta.	UCE1
RFO67	L'utente visualizza un errore quando il tipo di valore inserito del campo imponibile 1 non è numerico.	UCE2
RFO68	L'utente visualizza un errore quando il tipo di valore inserito del campo imponibile 2 non è numerico.	UCE2
RFO69	L'utente visualizza un errore quando il tipo di valore inserito del campo imponibile 3 non è numerico.	UCE2
RFO70	L'utente ricerca una o più offerte.	UC5.1
RFO71	L'utente ricerca un'offerta per ID.	UC5.1.1
RFO72	L'utente ricerca un'offerta per cliente associato.	UC5.1.2
RFO73	L'utente ricerca un'offerta per stato attuale.	UC5.1.3
RFO74	L'utente ricerca un'offerta per data di stipula.	UC5.1.4
RFO75	L'utente visualizza il dettaglio di un'offerta.	UC5.3
RFO76	L'utente visualizza l'ID di un'offerta.	UC5.2.1
RFO77	L'utente visualizza la data di stipula di un'offerta.	UC5.2.2
RFO78	L'utente visualizza il cliente coinvolto di un'offerta.	UC5.2.3
RFO79	L'utente visualizza lo stato di un'offerta.	UC5.2.4

Identificativo	Descrizione	Use case
RFO80	L'utente visualizza il titolo di un'offerta.	UC5.2.5
RFO81	L'utente visualizza il progetto di un'offerta.	UC5.2.6
RFO82	L'utente visualizza la descrizione dell'offerta.	UC5.3.1
RFO83	L'utente visualizza l'imponibile 1 dell'offerta.	UC5.3.2
RFO84	L'utente visualizza l'imponibile 2 dell'offerta.	UC5.3.3
RFO85	L'utente visualizza l'imponibile 3 dell'offerta.	UC5.3.4
RFO86	L'utente modifica l'offerta.	UC4.3.3.1
RFO87	L'utente elimina l'offerta.	UC5.3.5

Tabella 3.1: Requisiti funzionali

3.2.2 Requisiti di vincolo

Identificativo	Descrizione
RVO1	L'applicativo lato back-end è realizzato in Java.
RVO2	L'applicativo lato back-end è realizzato mediante il framework Spring.
RVO3	L'applicativo lato back-end è realizzato mediante il framework Hibernate.
RVO4	L'applicativo lato fron-end è realizzato tramite HTML5, CSS3 e JavaScript
RVO5	L'applicativo lato front-end è realizzato mediante il framework Bootstrap.
RVO6	L'applicativo deve essere funzionante in tutte le sue componenti.

Tabella 3.2: Requisiti di vincolo

Capitolo 4

Strumenti utilizzati

Il seguente capitolo ha la funzione di introdurre tutti gli strumenti utilizzati come supporto alle attività di sviluppo dell'applicativo, tra cui: ambiente di lavoro, framework, linguaggi, librerie a supporto della codifica e IDE. Tutte le tecnologie elencate rappresentano vincolo di sviluppo del progetto.

4.1 Ambiente di lavoro

CWBI ha a disposizione circa 30 macchine fisiche con sistema operativo Debian, ciascuna delle quali viene utilizzata come supporto per delle macchine virtuali in ambiente Windows 7. Il lavoro quotidiano viene svolto in ambiente virtuale e non fisico; per la connessione alla propria macchina virtuale viene utilizzato il software *VMware Workstation*. Il vantaggio nell'utilizzare macchine virtuali sta nel fatto che tutti i dipendenti presentano la medesima configurazione dell'ambiente di sviluppo e, essendo il lavoro in azienda molto collaborativo, al presentarsi di un problema o di un dubbio sul "cosa fare" i colleghi possono connettersi alla macchina virtuale dove si è presentato il problema per aiutare e velocizzare lo scambio di idee sul come fare.

4.2 Framework

Nella seguente sezione sono riportati i [framework](#)^[g] utilizzati nelle attività di sviluppo del progetto commissionato.

4.2.1 Spring (core)



Figura 4.1: Logo di Spring

[Spring](#)^[b] è un framework open source nato per lo sviluppo di applicazioni Java EE. Una delle sue maggiori peculiarità risiede nell'essere **modulare**, il che consente di utilizzare anche solo una parte delle funzionalità che il framework mette a disposizione. Altro punto di forza è la facile integrazione con altri framework esistenti, come: Hibernate, Apache Struts, ecc.

Per lo sviluppo back-end viene utilizzato solamente il *core container* offerto dal framework:



Figura 4.2: Spring - core container

1. il modulo bean fornisce le classi factory per l'istanziamento dei bean;
2. il modulo core è la componente fondamentale del framework ed implementa la Dependency Injection;
3. il modulo context fornisce l'accesso a tutti i bean definiti in funzione dei relativi contesti;
4. il modulo SpEL fornisce un expression language per la manipolazione dei bean a runtime.

4.2.2 Hibernate



Figura 4.3: Logo di Hibernate

Hibernate è un framework open source che permette di rendere persistenti i dati dall'ambiente Java al database mappando gli oggetti in apposite tabelle di un database relazionale.

[Hibernate](#)^[b] gestisce inoltre l'accesso e la manipolazione dei dati stessi generando automaticamente le query in linguaggio SQL; questo agevola il lavoro dello sviluppatore, che non si deve occupare della scrittura manuale delle query di accesso ai dati del database ma solamente di definire gli oggetti che andranno memorizzati e gestiti nel database.

4.2.3 Bootstrap



Figura 4.4: Logo di Bootstrap

[Bootstrap](#)^[b] è un framework open source per lo sviluppo di interfacce web basato su CSS. Bootstrap mette a disposizione una serie di file CSS contenenti stili e regole applicabili ai vari componenti HTML5.

Gli stili che bootstrap propone sono ottimizzati per la visualizzazione smartphone, tablet e desktop. Inoltre, dato che la maggior parte degli accessi ad applicativi web avviene tramite cellulare, il framework mette al primo posto il layout per smartphone. Bootstrap è inoltre compatibile con i browser più moderni.

4.2.4 Apache Struts



Figura 4.5: Logo di Apache Struts

[Apache Struts](#)^[b] è un framework open source Java che aiuta gli sviluppatori a creare applicazioni web flessibili, gestibili e sicure in Java. Il framework è caratterizzato da un architettura MVC che consente di raccogliere le richieste provenienti dalla vista e richiamare attraverso il controller le operazioni del modello necessarie a rendere persistenti i dati nel database. Tale architettura si integra con il framework Spring sopra citato.

4.3 Linguaggi utilizzati

Nella seguente sezione sono riportati i linguaggi utilizzati nella codifica dell'applicativo commissionato.

4.3.1 Java



Figura 4.6: Logo di Java

Per lo sviluppo dell'applicativo lato back-end si è utilizzato Java, un linguaggio di programmazione orientato agli oggetti. La caratteristica principale di questo linguaggio risiede nel rendere indipendente la scrittura di codice dall'ambiente di esecuzione fisico istanziando un ambiente di esecuzione virtuale del codice noto come JVM.

4.3.2 HTML5, CSS3 e JavaScript



Figura 4.7: Logo dello stack tecnologico del front-end

Per lo sviluppo dell'applicativo lato front-end si è utilizzato lo stack tecnologico costituito da:

- HTML5: linguaggio di markup per la creazione di siti web;
- CSS3: linguaggio utilizzato per definire la formattazione dei documenti HTML attraverso apposite regole di stile;
- JavaScript: linguaggio di programmazione multi paradigma orientato agli eventi.

4.4 Librerie a supporto della codifica

Nella seguente sezione sono riportate le librerie utilizzate nelle attività di sviluppo del progetto commissionato.

4.4.1 Apache Commons



Figura 4.8: Logo di Apache Commons

[Apache Commons](http://commons.apache.org/)^[b] è una libreria Java molto utilizzata nello sviluppo di applicazioni basate su Java. Questa libreria offre molte classi caratterizzate da diversi metodi che possono essere richiamati dallo sviluppatore semplicemente importando la relativa classe di interesse. Alcune tra le classi offerte sono le seguenti:

- `StringUtils`: implementa una serie di operazioni sulle stringhe che completano/estendono quelle offerte dalla classe standard `java.lang.String`;
- `ArrayUtils`: implementa dei metodi che consentono di processare e controllare gli array;
- `NumberUtils`: implementa una serie di metodi a supporto dei tipi numerici offerti da Java.

4.4.2 JSTL

JavaServer Pages Standard Tag Library (JSTL) è una libreria utilizzata nello sviluppo di applicazioni web Java EE. È un'estensione di Java Server Page (JSP) ed incorpora un insieme di tag HTML definiti tramite file XML e programmati in linguaggio Java. Oltre ai tag standard proposti dalla libreria, JSTL permette allo sviluppatore di definire dei tag custom.

Il vantaggio della creazione di tag risiede nel definire dei comportamenti univoci dello stesso tag in contesti diversi eliminando la necessità di definire funzioni che si ripetono all'interno delle pagine.

4.5 IDE



Figura 4.9: Logo di Eclipse

Eclipse è un ambiente di sviluppo che supporta diversi linguaggi, tra cui Java. Questo [IDE](#)^[g] è stato utilizzato per lo sviluppo dell'intero progetto. Tra le diverse funzionalità offerte da questo ambiente troviamo:

- importare un progetto esistente;
- creare un nuovo progetto;
- attività di refactor;
- attività di debugging.

Capitolo 5

Progettazione

Il seguente capitolo ha la funzione di illustrare l'approccio alla progettazione e all'architettura del sistema commissionato.

5.1 Approccio MDA

La filosofia dell'azienda per fronteggiare nuove attività e progetti si basa su un approccio di tipo MDA (Model Driven Architecture). L'idea alla base di tale approccio risiede nel fatto che il cuore di ogni applicazione si traduce in un modello solido e ben strutturato.

L'approccio MDA è stato utilizzato anche per affrontare la fase di progettazione del sistema commissionato. In particolare la prima attività è stata quella di definire i modelli su cui l'intero sistema si basa. Per modello si intende un oggetto che deve essere mappato e gestito in una opportuna tabella di un database relazionale.

I modelli vengono rappresentati dalla seguente stringa:

NomeModello (TipoAttributo1 nomeAttributo1, ..., TipoAttributoN nomeAttributoN).

In particolare i modelli identificati in base alle richieste del committente sono i seguenti:

- **Progetto** (Long idProponente, Long idReferente, String codice, String titolo, String descrizione, String statoAvanzamento, Integer giorni, Integer percentualeAvanzamento, GregorianCalendar dataInizio)

DESCRIZIONE: questo modello rappresenta un progetto commissionato all'azienda da parte di un cliente o ideato dall'azienda per raggiungere nuovi clienti.

ATTRIBUTI:

- *idProponente* corrisponde all'identificativo del proponente, ovvero l'azienda da cui nasce l'idea del progetto (CWBI o altro cliente);
- *idReferente* corrisponde all'identificativo di un dipendente di CWBI che assume il ruolo di responsabile del progetto;
- *codice* corrisponde ad un identificativo dei progetti (es. 2023.70);
- *titolo* corrisponde al titolo del progetto;
- *descrizione* corrisponde alla descrizione generale progetto che illustra l'obiettivo e le funzionalità;
- *statoAvanzamento* corrisponde allo stato attuale del progetto. Lo stato può assumere i seguenti valori: aperto, chiuso, approvato, rifiutato, sospeso;
- *giorni* corrisponde ai giorni di sviluppo previsti per portare a termine il progetto;
- *percentualeAvanzamento* corrisponde alla percentuale di avanzamento delle attività di progetto;
- *dataInizio* corrisponde alla data di inizio delle attività di progetto.

- **CorrelazioneProgettoCliente** (Long idCliente, Long idProgettoCliente)

DESCRIZIONE: questo modello rappresenta la correlazione che persiste tra un progetto e un certo cliente. Si fa notare che uno stesso progetto può raggiungere più di un cliente.

ATTRIBUTI:

- *idCliente* corrisponde all'identificativo di un cliente;
- *idProgettoCliente* corrisponde all'identificativo di un progetto esistente.

- **Offerta** (Long idCliente, Long idProgetto, String titolo, String descrizione, String stato, GregorianCalendar dataStipula, BigDecimal imponible1, BigDecimal imponible2, BigDecimal imponible3)

DESCRIZIONE: questo modello rappresenta l'offerta che viene stipulata per un certo cliente in merito ad uno dei progetti a disposizione.

ATTRIBUTI:

- *idCliente* corrisponde all'identificativo di un cliente;
- *idProgettoCliente* corrisponde all'identificativo di un progetto esistente;
- *titolo* corrisponde al titolo dell'offerta (es. Contratto n.7/2023);
- *descrizione* corrisponde alla descrizione dell'offerta emessa (note, informazioni utili, ecc.); *stato* corrisponde allo stato attuale dell'offerta. Lo stato può assumere i seguenti valori: in preparazione, in accettazione, accettata;
- *dataStipula* corrisponde alla data in cui l'offerta è stata emessa;
- *imponible1* corrisponde al primo prezzo proposto per firmare l'offerta;
- *imponible2* corrisponde al secondo prezzo proposto per firmare l'offerta;
- *imponible3* corrisponde al terzo prezzo proposto per firmare l'offerta.

I modelli **Cliente** e **Utente**, necessari per realizzare il progetto commissionato, erano già stati definiti per un altro modulo dell'applicazione CW GEST. In particolare questi modelli rappresentano rispettivamente un cliente e un dipendente di CWBI. Dato che tali modelli presentano le medesime caratteristiche pensate per il progetto in questione non si è reso necessario alterarli per l'utilizzo.

5.2 Architettura MVC

Come citato in precedenza, il framework Apache Struts mette a disposizione un'architettura **MVC** (Model View Controller) per costruire applicazioni web.

L'architettura di Struts non si differenzia particolarmente dalla filosofia di MVC: il controller gestisce tutte le operazioni della webapp e ha un riferimento sia al modello sia alla vista.

L'architettura MVC messa a disposizione da Apache Struts viene rappresentata dal seguente schema:

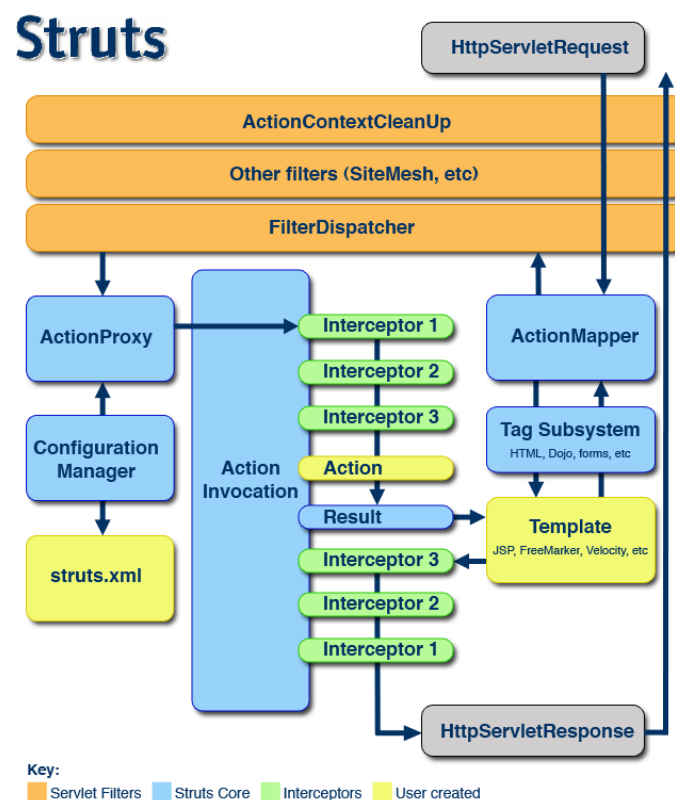


Figura 5.1: Architettura MVC di Apache Struts

I tre componenti principali della filosofia MVC corrispondono a:

- **FilterDispatcher** (Controller);
- **Action** (Model);
- **Template** (View).

Come si può notare dalla [figura 5.1](#), l'architettura offerta dal framework è caratterizzata da diversi componenti che risulta necessario menzionare per apprendere a fondo il

pieno funzionamento del sistema MVC di Apache Struts. Questi componenti sono i seguenti:

- **FilterDispatcher**: rappresenta il controller del framework ed è il primo componente ad agire nel ciclo di elaborazione della richiesta. Il controller ispeziona ogni richiesta in arrivo per determinare quale action dovrebbe gestire la richiesta.
- **Action**: rappresenta una funzione dell'applicativo ed è qui che viene definita la logica applicativa del sistema.
In particolare ogni modello è caratterizzato da n action che gestiscono le funzionalità che devono essere garantite all'utente (es. una action per la pagina di dettaglio, una per la creazione/modifica dell'oggetto, una per la ricerca dell'oggetto attraverso dei filtri, ecc.).
- **ActionMapper**: viene utilizzata da FilterDispatcher per determinare se la richiesta deve richiamare una action o meno. Quando viene fornito un `HttpServletRequest`, ActionMapper può restituire null se nessuna action corrisponde alla richiesta oppure può restituire un `ActionMapping` che descrive la chiamata.
- **ActionProxy**: se ActionMapper determina che deve essere richiamata una action per tale richiesta, FilterDispatcher delega il controllo ad ActionProxy. L'ActionProxy si consulta con ConfigurationManager, un'interfaccia che descrive la configurazione del framework, e crea una `ActionInvocation` che si occupa di richiamare l'opportuna action che deve gestire la richiesta entrante.
- **Interceptor**: corrispondono a dei componenti che vengono richiamati prima di eseguire la action e hanno il compito di preparare e inizializzare le action stesse.
- **Result**: corrisponde al risultato costruito e ritornato dalla action e che viene rappresentato da una pagina JSP.
- **Template**: corrispondono alle pagine di visualizzazione che effettivamente restituiscono la risposta all'utente.

Il ciclo che parte da una richiesta entrante risulta essere il seguente:

1. la richiesta viene raccolta da `ActionMapper`;
2. viene chiamato il filtro `FilterDispatcher` che consulta `ActionMapper` per determinare se un'azione deve essere richiamata;
3. se `ActionMapper` trova che un'azione deve essere richiamata, `FilterDispatcher` delega il controllo ad `ActionProxy`;
4. `ActionProxy` legge il file di configurazione (`struts.xml`). `ActionProxy` crea un'istanza della classe `ActionInvocation` e delega il controllo.
5. `ActionInvocation` gli `Interceptor` uno per uno (se richiesto) e quindi invoca la `action`;
6. la `action` ritorna il risultato, gli `Interceptor` vengono eseguiti nuovamente in ordine inverso e la risposta viene restituita a `FilterDispatcher`;
7. Il risultato viene quindi inviato al contenitore `servlet` che a sua volta lo rispedisce al client.

Capitolo 6

Prodotto finale

Il seguente capitolo ha la funzione di illustrare il prodotto realizzato in tutte le sue componenti.

6.1 Pagina 1 della webapp

6.2 Pagina 2 della webapp

6.3 Pagina 3 della webapp

6.4 Pagina 4 della webapp

6.5 Pagina 5 della webapp

6.6 Pagina N della webapp

Capitolo 7

Conclusioni

Il seguente capitolo ha la funzione di illustrare l'attività di test eseguita sul prodotto finale. Viene proposto anche un resoconto complessivo sull'attività di stage.

7.1 Test e validazione

7.2 Resoconto dello stage

Sitografia

Spring <https://spring.io/>

Hibernate <https://hibernate.org/>

Bootstrap <https://getbootstrap.com/>

Apache Struts <https://struts.apache.org/>

Apache Commons <https://commons.apache.org/>

Glossario

UML (Unified Modeling Language)

Linguaggio che permette, tramite l'utilizzo di modelli visuali, di analizzare, descrivere, specificare e documentare un sistema software anche complesso.

Requisito

Descrizione dei servizi che un sistema software deve fornire, insieme ai vincoli da rispettare sia in fase di sviluppo che durante la fase di operatività del software.

Framework

Architettura logica di supporto sulla quale un software può essere progettato e realizzato, spesso facilitandone lo sviluppo da parte del programmatore.

IDE (Integrated Development Environment)

Software che offre agli sviluppatori un ambiente per lo sviluppo, il test e il debug di un'applicazione.