

UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE ED
ELETTRICA E MATEMATICA APPLICATA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA



ELABORATO FINALE

SICUREZZA E AUTOMAZIONE DI UN WORKFLOW DOCUMENTALE CON FIRMA DIGITALE

Relatore

Prof. Francesco Moscato

Candidato

Domenico di Stasio
Matr. 0612704820

Tirocinio svolto presso: Ericsson Telecomunicazioni S.p.A.
Tutor Aziendale: Dott. Antonio Fresa

Anno Accademico 2021 – 2022

*A mia madre,
che mi ha insegnato
il coraggio e l'indipendenza*

Abstract

Descrizione del problema affrontato

La tesi è incentrata sulla progettazione e lo sviluppo di un software per automatizzare un workflow documentale procedurale dell'azienda Ericsson Telecomunicazioni S.p.A.

La problematica affrontata consiste nello sbilanciamento del carico di lavoro all'interno del workflow documentale, che per svariate motivazioni comporta un rallentamento del processo con possibili ripercussioni sulla produttività dell'azienda.

L'obiettivo della tesi è di mostrare come è stato progettato e sviluppato un prototipo in grado di automatizzare il processo e di rendere il workflow più efficiente.

In particolare, si intende esaminare le best practice e le tecnologie disponibili per implementare l'automazione del workflow, rendendo più agevole l'accesso alle informazioni e semplificando il processo di firma digitale.

Inquadramento dell'elaborato nello scenario tecnico contemporaneo

La diffusione sempre più capillare della digitalizzazione degli uffici e dei processi aziendali ha portato ad un crescente utilizzo di firme digitali per la sottoscrizione di documenti. Tuttavia, il loro utilizzo deve essere garantito in un contesto di sicurezza e privacy adeguate.

La seguente tesi si propone di analizzare le problematiche relative alla sicurezza e all'automazione di un workflow documentale con firma digitale, esaminando le problematiche proposte nel lavoro di tirocinio svolto e proponendo soluzioni tecniche per migliorare l'efficienza e la sicurezza del processo.

Contributo personale del candidato alla soluzione del problema descritto

Il lavoro svolto riguarda la soluzione alla problematica descritta e consiste in una applicazione web in grado di interfacciare un database di anagrafica aziendale in modo da automatizzare il flusso di lavoro inerente la firma digitale di un documento legale.

La soluzione proposta implementa alcune delle funzionalità più importanti per migliorare l'efficienza del processo, garantendo sicurezza e affidabilità.

Descrizione dei contenuti applicativi/sperimentali dell'elaborato

L'elaborato ha incluso una fase di analisi dei requisiti, seguita dalla progettazione della base di dati e dall'implementazione di un'applicazione web.

Durante la fase di analisi dei requisiti sono stati determinati i requisiti funzionali e non funzionali al fine di strutturare al meglio la progettazione del software.

La soluzione proposta ha previsto l'utilizzo di tecnologie per la programmazione front-end e back-end come JavaScript e PHP e l'utilizzo di SQL per la creazione e il popolamento iniziale del database.

Per il testing e le anteprime del prototipo è stato utilizzato l'applicativo XAMPP per Apache Web Server e MySQL.

INDICE

<i>Abstract</i>	3
CAPITOLO 1.	7
<i>OBIETTIVI E TEMATICHE DI RICERCA E SVILUPPO</i>	7
1.1 INTRODUZIONE	7
1.2 OBIETTIVO	8
1.3 TEMATICHE TRATTATE	9
1.4 ORGANIZZAZIONE DEI CONTENUTI	10
CAPITOLO 2.	11
<i>CONTESTO, PROBLEMA, MODELLI E TECNOLOGIE ABILITANTI</i>	11
2.1 INTRODUZIONE	11
2.2 IL CONTESTO	11
2.3 IL PROBLEMA	12
2.4 OVERVIEW DELLO STATO DELL'ARTE	13
2.5 BACKGROUND TECNOLOGICO	13
CAPITOLO 3.	15
<i>PROGETTAZIONE E IMPLEMENTAZIONE DEL PROTOTIPO</i>	15
3.1 INTRODUZIONE	15
3.2 IL METODO PER LA PROGETTAZIONE E LO SVILUPPO	15
3.3 I REQUISITI	16
3.3.1 LE FINALITA' DEL SISTEMA PROTOTIPALE	16
3.3.2 REQUISITI UTENTE	17
Requisiti funzionali	17
Requisiti non funzionali	18
3.4 DIAGRAMMA DEL PROCESSO PRINCIPALE	18

3.5 VISTA ARCHITETTURALE DI MASSIMA	20
3.6 PROGETTAZIONE DELLA BASE DI DATI	21
3.6.1 PROGETTAZIONE CONCETTUALE	21
3.6.2 PROGETTAZIONE LOGICA	23
3.7 TECNOLOGIE ABILITANTI	29
3.8 IMPLEMENTAZIONE	32
Connessione al database	32
Logout	33
Login	33
Register	36
Visualizzazione degli elementi con Javascript	37
Generazione documento precompilato	38
CAPITOLO 4.	40
<i>IL FUNZIONAMENTO DEL PROTOTIPO</i>	40
4.1 DATI	40
4.2 SCENARIO APPLICATIVO	41
4.3 CONCLUSIONI E SVILUPPI FUTURI	45
<i>Bibliografia</i>	47
<i>Indice delle Figure</i>	47
<i>Indice delle Tabelle</i>	48
<i>Ringraziamenti</i>	49

CAPITOLO 1.

OBIETTIVI E TEMATICHE DI RICERCA E SVILUPPO

1.1 INTRODUZIONE

Il presente elaborato è il risultato del lavoro che ho svolto durante il mio periodo di tirocinio presso l'azienda Ericsson Telecomunicazioni S.p.A., una rinomata società multinazionale leader nel settore delle tecnologie dell'informazione e della comunicazione, comunemente conosciuto come ICT (Information and Communication Technology).

Durante il periodo di tirocinio, ho avuto l'opportunità di operare nel ramo dell'innovazione di processo per la sicurezza delle informazioni. In particolare, il mio lavoro si è concentrato sulla progettazione e lo sviluppo di un sistema per l'automazione di un workflow documentale procedurale.

L'obiettivo principale del mio lavoro era la creazione di un prototipo di sistema che potesse facilitare il flusso di lavoro documentale all'interno dell'azienda, migliorando l'efficienza e riducendo il rischio di errori umani.

Per la realizzazione di questo sistema, ho optato per lo sviluppo di un'applicazione Web, in modo da garantire l'accessibilità e l'usabilità del sistema tramite un'interfaccia grafica intuitiva.

L'attività di progettazione ha richiesto la definizione di un processo principale e la definizione dei requisiti funzionali e non funzionali del sistema, la progettazione di una base di dati e l'ideazione di un design per le componenti dell'interfaccia utente.

Successivamente, ho proceduto con la fase di sviluppo del sistema, che ha coinvolto la scrittura del codice, la configurazione dei server e la creazione e il popolamento della base di dati.

Infine, ho condotto una serie di test per verificare l'efficacia e l'efficienza del sistema, insieme alla sua capacità di rispondere alle esigenze del cliente.

Grazie a questo lavoro, ho acquisito competenze tecniche e metodologiche utili per la mia futura carriera professionale nel settore dell'ICT.

In sintesi, il mio lavoro ha portato alla realizzazione di un prototipo di sistema innovativo ed efficiente per l'automazione di un workflow documentale procedurale, che potrebbe rappresentare un valore aggiunto per l'azienda Ericsson Telecomunicazioni S.p.A. e per il settore ICT in generale.

1.2 OBIETTIVO

L'obiettivo dell'elaborato che presento è la progettazione e la realizzazione di un prototipo per l'automazione di un flusso di lavoro documentale che coinvolge i dipendenti di un'azienda.

Lo scopo principale del prototipo è di semplificare e velocizzare il processo di lavoro, riducendo gli errori e aumentando l'efficienza del sistema. Per raggiungere l'obiettivo prefissato, è richiesta l'implementazione di un'interfaccia grafica che consenta a ogni utente di completare dei task, in modo da contribuire al workflow.

L'interfaccia grafica deve essere intuitiva e facile da usare, in modo da garantire un'esperienza d'uso piacevole e senza problemi.

Per la realizzazione del prototipo, è necessario svolgere diverse attività. In primo luogo, è necessario studiare e analizzare il workflow attuale dell'azienda, in modo da comprendere nel dettaglio le fasi in cui è articolato il processo aziendale e determinare i requisiti non funzionali necessari per il suo miglioramento.

In secondo luogo, è importante analizzare i requisiti di processo per definire una procedura automatizzata e determinare i requisiti funzionali necessari per lo sviluppo del prototipo. Questa fase richiede una profonda conoscenza del processo aziendale e dei requisiti dei dipendenti coinvolti nel flusso di lavoro.

Infine, la realizzazione del prototipo richiede la creazione di un'applicazione Web che possa interfacciarsi con un database di anagrafica aziendale, in modo da garantire la coerenza dei dati e la sicurezza dell'informazione.

Inoltre, l'applicazione deve essere sviluppata utilizzando delle buone pratiche di sviluppo software e deve essere testata per garantire la sua efficienza e la sua affidabilità.

In sostanza, l'obiettivo dell'elaborato è la progettazione e la realizzazione di un prototipo per l'automazione di un flusso di lavoro documentale che coinvolge i dipendenti di un'azienda. La realizzazione del prototipo richiede lo studio e l'analisi del workflow attuale, l'analisi dei requisiti di processo e la creazione di un'applicazione Web che possa interfacciarsi con un database.

1.3 TEMATICHE TRATTATE

Nell'elaborato sono state affrontate prima tematiche prettamente teoriche, per poi valutarne un risvolto pratico nell'ambito della progettazione e dello sviluppo software.

Nella trattazione delle tematiche, sono stati presi in considerazione argomenti relativi alla sicurezza e all'innovazione di processo, all'automazione di workflow documentali e allo studio di processi aziendali. Ogni aspetto è stato approfondito in modo trasversale, dando spazio alla progettazione di applicazioni Web e allo sviluppo front-end e back-end.

Per raggiungere l'obiettivo prefissato, è stato necessario studiare e analizzare il workflow allo stato dell'arte, al fine di comprendere nel dettaglio le fasi in cui è articolato il processo aziendale e determinare i requisiti non funzionali.

In seguito, è stata effettuata un'analisi dei requisiti di processo per definire una procedura automatizzata e determinare i requisiti funzionali. La realizzazione del prototipo è stata ottenuta mediante lo sviluppo di un'applicazione Web in grado di interfacciarsi ad un database di anagrafica aziendale. Inoltre, il prototipo è stato progettato in modo da garantire che, tramite un'interfaccia grafica, l'interazione con gli utenti sia facilitata.

Infine, nella trattazione dell'elaborato, sono stati coinvolti anche aspetti pratici inerenti lo sviluppo del prototipo.

Grazie all'implementazione del sistema, è stato possibile dimostrare a grandi linee il funzionamento del sistema progettato e comprendere le potenzialità dell'automazione di un flusso di lavoro documentale per il miglioramento dei processi aziendali.

1.4 ORGANIZZAZIONE DEI CONTENUTI

Il presente lavoro è suddiviso in vari capitoli, ognuno dei quali mira ad approfondire in modo completo il tema della realizzazione del sistema per l'automazione del workflow documentale aziendale tramite l'utilizzo di tecnologie web e la creazione di un prototipo funzionante.

Nel capitolo 1 sono stati presentati gli obiettivi e le tematiche di ricerca e sviluppo.

Il capitolo 2 tratta del contesto, del problema, dei modelli e delle tecnologie abilitanti. Dopo un'introduzione, vengono analizzati il contesto e il problema affrontato, seguiti da una panoramica dello stato dell'arte e dal background tecnologico.

Il capitolo 3 si concentra sulla progettazione e l'implementazione del prototipo. Dopo un'introduzione, viene presentato il metodo utilizzato per la progettazione e lo sviluppo del sistema. Vengono inoltre delineati i requisiti del prototipo e viene presentato il diagramma del processo principale. La vista architetturale di massima e la progettazione della base di dati vengono successivamente analizzate, insieme alle tecnologie abilitanti utilizzate e alla fase di implementazione.

Infine, il capitolo 4 descrive il funzionamento del prototipo, attraverso la presentazione dei dati, lo scenario applicativo e le conclusioni sul lavoro svolto, insieme ad eventuali possibili sviluppi futuri.

CAPITOLO 2.

CONTESTO, PROBLEMA, MODELLI E TECNOLOGIE ABILITANTI

2.1 INTRODUZIONE

L'informatizzazione totale o parziale di un processo organizzativo che costituisce un "flusso di lavoro" [1] si riferisce all'automazione di processi in cui documenti, informazioni o altre attività sono distribuiti tra i partecipanti secondo regole definite congiuntamente al fine di raggiungere o contribuire a un obiettivo aziendale più ampio [3].

Per raggiungere questo obiettivo utilizzando gli strumenti della Information Technology possono essere realizzati dei sistemi per gestire questi workflow aziendali. Il sistema di gestione del flusso di lavoro è quello che definisce, gestisce ed esegue integralmente i "flussi di lavoro" mediante l'utilizzo di software, la cui esecuzione è controllata da una rappresentazione elettronica della logica del flusso di lavoro [2,3].

2.2 IL CONTESTO

Il contesto in cui è stato prodotto l'elaborato ruota intorno ad una necessità aziendale, costituita in parte dall'esigenza di innovare un processo burocratico interno e in parte dall'imposizione della sicurezza di processo, che in una realtà aziendale popolata da attività di Information Security risulta di vitale importanza.

Il workflow documentale da automatizzare è relativo a documenti di tipo legale, motivo per cui ogni riferimento dettagliato inerente al contesto reale verrà generalizzato.

Generalizzando, quindi, nell'ambito del problema affrontato vi sono tre attori, che per comodità sono definiti "Utente1", "Utente2" e "Utente3".

Il soggetto è costituito dal documento, che al fine di abilitare l'Utente1 ad avere accesso a determinate attività aziendali, deve essere firmato in sequenza con responsabilità crescente, prima dall'Utente1, poi dall'Utente2 e infine dall'Utente3.

Quando il documento presenta le tre firme, con i dati di ciascun utente, può essere archiviato e l'Utente1 può essere abilitato.

Oltre ai tre tipi di utenti, è presente un quarto utente, che definiamo Admin, cioè un dipendente dell'azienda che gestisce l'intero processo, amministrando una casella di posta elettronica in cui:

- Riceve le richieste dagli Utenti2 per attivare gli Utenti1
- Riceve richieste di chiarimenti dagli Utenti1
- Invia chiarimenti agli Utenti1
- Invia richieste di informazioni agli Utenti1
- Riceve informazioni degli Utenti1 e le registra
- Inoltra il documento firmato agli utenti successivi
- Archivia il documento completamente firmato
- Attiva gli Utenti1 il cui processo risulta correttamente completo

2.3 II PROBLEMA

Dalla descrizione del contesto trattata nel precedente paragrafo, risulta evidente come il carico di lavoro all'interno del workflow sia sbilanciato verso l'Admin, con un conseguente rallentamento dell'intero processo, anche dal punto di vista delle altre tipologie di utenti.

Il problema dello sbilanciamento del carico di lavoro nel workflow trattato può avere ripercussioni non solo sul tempo ma anche sulla produttività generale dell'azienda.

Infatti, se l'Admin è costretto a gestire tutte le fasi del processo, gli altri utenti rischiano di dover aspettare troppo a lungo per ottenere il supporto necessario, con conseguente ritardo nelle attività lavorative. Questo rallentamento può a sua volta generare un'escalation degli errori, che potrebbero essere evitati con una gestione più efficiente del workflow.

Una soluzione possibile per ridurre lo sbilanciamento del carico di lavoro potrebbe essere l'implementazione di un sistema di monitoraggio delle attività dei singoli utenti che potrebbe aiutare a mantenere alta la qualità del lavoro svolto, anche in assenza di un controllo diretto dell'Admin.

In conclusione, è importante tenere in considerazione gli effetti negativi dello sbilanciamento del carico di lavoro nel workflow, e adottare le misure necessarie per garantire un equilibrio tra le responsabilità degli utenti coinvolti nel processo. Ciò può contribuire a migliorare l'efficienza complessiva dell'azienda, ridurre i tempi di attesa e aumentare la qualità del lavoro svolto.

2.4 OVERVIEW DELLO STATO DELL'ARTE

Lo stato del sistema utile a gestire il workflow presentato consiste da un lato in una procedura ben definita, realizzata ad hoc per garantire il requisito di sicurezza e dall'altro lato vede un solo modulo applicativo implementato, cioè il sistema per la firma digitale del documento.

2.5 BACKGROUND TECNOLOGICO

Il background tecnologico dell'applicazione web sviluppata è il risultato di una scelta ponderata e mirata a fornire un'esperienza utente ottimale.

Infatti, il sistema si basa su una combinazione di tecnologie front-end e back-end che si integrano in modo sinergico per garantire la massima efficienza e sicurezza nella gestione del workflow documentale.

Per quanto riguarda la parte front-end dell'applicazione, è stato scelto di utilizzare l'HTML per definire la struttura delle pagine web, un linguaggio di markup che rappresenta lo standard de facto per la creazione di siti web statici. Inoltre, per garantire una presentazione visiva accattivante e user-friendly, è stata utilizzata la libreria Bootstrap, che ha permesso di realizzare un'interfaccia grafica responsive e altamente personalizzabile.

Per rendere la visualizzazione degli elementi dinamica e interattiva, è stato scelto di utilizzare il linguaggio di scripting JavaScript, che ha permesso di creare animazioni e di gestire gli eventi generati dall'utente in modo rapido ed efficiente.

Per quanto riguarda la parte back-end, invece, è stato scelto di utilizzare il linguaggio di programmazione PHP, che ha consentito di sviluppare le funzionalità del server-side

dell'applicazione in modo affidabile e performante. Inoltre, per la gestione dei dati e del database, è stato utilizzato il software phpMyAdmin e il linguaggio di interrogazione SQL, che hanno consentito di gestire in modo efficiente e sicuro le informazioni relative al workflow documentale.

In sintesi, la scelta di queste tecnologie si è rivelata vincente, poiché ha permesso di creare un'applicazione web performante, scalabile e sicura, in grado di gestire in modo efficiente il workflow documentale, che rappresenta un aspetto cruciale per molte realtà aziendali moderne.

CAPITOLO 3.

PROGETTAZIONE E IMPLEMENTAZIONE DEL PROTOTIPO

3.1 INTRODUZIONE

In questo capitolo, si procederà all'analisi dettagliata dei requisiti del sistema per la gestione del processo documentale con firma digitale, nonché alla definizione del processo principale e alla presentazione della vista architettuale di massima.

Verrà inoltre approfondita la progettazione della base di dati, i requisiti funzionali e non funzionali del sistema e il diagramma del processo principale che illustra il flusso di lavoro.

La vista architettuale di massima fornirà una panoramica completa delle componenti e dei sottosistemi del sistema, mentre la progettazione della base di dati descriverà in dettaglio come si è giunti a determinate scelte per la creazione della base di dati, lo schema dei dati del sistema e la gestione delle relazioni tra essi. Saranno inoltre presentate le tecnologie abilitanti utilizzate per lo sviluppo del sistema, tra cui linguaggi di programmazione, librerie e framework, insieme all'implementazione del sistema stesso, che mostrerà come le diverse componenti del sistema sono state integrate per garantire l'automazione e la sicurezza del workflow.

Questo capitolo fornirà una visione completa e dettagliata del processo di progettazione e implementazione del sistema per la gestione del processo documentale, mettendo in luce le tecnologie e le soluzioni adottate per garantire la massima efficienza e sicurezza del workflow.

3.2 II METODO PER LA PROGETTAZIONE E LO SVILUPPO

Come metodologia di progettazione del software sviluppato è stato utilizzato il modello a cascata.

Il modello a cascata è una delle metodologie di sviluppo software più tradizionali e viene utilizzato per sviluppare software su larga scala.

Questo modello prevede che lo sviluppo del software avvenga in fasi sequenziali, ognuna delle quali deve essere completata prima di procedere alla successiva. Le fasi principali del modello a cascata includono la pianificazione, l'analisi, la progettazione, lo sviluppo, il testing e la manutenzione del software.

In particolare, il modello a cascata prevede che il lavoro sul software inizi con la fase di pianificazione, in cui viene definito l'obiettivo del progetto, i requisiti del software e le risorse necessarie per completare il lavoro. Dopo la fase di pianificazione, segue la fase di analisi, in cui vengono raccolti e analizzati i requisiti del software.

La fase di progettazione prevede la definizione dell'architettura del software, dei componenti e delle interfacce.

La fase di sviluppo prevede la scrittura del codice del software, mentre la fase di testing viene utilizzata per verificare che il software funzioni correttamente.

Infine, la fase di manutenzione viene utilizzata per correggere eventuali problemi riscontrati durante l'utilizzo del software.

L'utilizzo del modello a cascata come metodologia di progettazione del software mi ha consentito di sviluppare un prototipo di un'applicazione web robusta, affidabile e di alta qualità, in grado di soddisfare i requisiti specifici del progetto.

3.3 I REQUISITI

3.3.1 LE FINALITÀ DEL SISTEMA PROTOTIPALE

Il prototipo deve consentire agli utenti di gestire in modo efficiente il flusso di lavoro dei documenti all'interno dell'organizzazione.

In generale, l'applicazione dovrebbe avere una serie di funzionalità che permettano agli utenti di svolgere tutte le azioni necessarie per il completamento del processo.

Ad esempio, dovrebbe essere possibile effettuare la registrazione degli utenti e accedere alla propria area privata all'interno dell'applicazione. Inoltre, gli utenti dovrebbero essere in grado di caricare e scaricare i documenti all'interno dell'applicazione, assegnando loro le informazioni necessarie per il processo.

L'applicazione web dovrebbe consentire anche la gestione del workflow, ovvero permettere agli utenti di partecipare al flusso di lavoro del processo documentale e gestirlo.

Deve inoltre essere possibile firmare il documento, in modo da garantire l'autenticità e l'integrità dei dati. Per questo aspetto in particolare non è richiesta un'implementazione poiché allo stato dell'arte è già presente un modulo dedito alla firma digitale.

L'applicazione web dovrebbe permettere agli utenti di monitorare il processo di workflow, visualizzando lo stato dei documenti, i passaggi completati e quelli ancora da completare.

Infine, una volta completato il processo, i documenti dovrebbero poter essere archiviati all'interno dell'applicazione.

Per la gestione dei dati dei documenti e dei flussi di lavoro, l'applicazione web dovrebbe interfacciarsi con un database dedicato.

3.3.2 REQUISITI UTENTE

Requisiti funzionali

- RF01. Registrazione degli utenti: L'utente "Utente1" deve poter essere registrato da un utente "Utente2" inserendo userid, nome, cognome e una password.
- RF02. Caricamento dei documenti: L'utente deve poter caricare il documento all'interno dell'applicazione, assegnando le informazioni necessarie per il workflow. Deve altrettanto consentire di scaricare il documento contenente dati che vengono estratti dalla base di dati.
- RF03. Monitoraggio del processo: L'Utente2 e Utente3 devono essere in grado di monitorare il processo dalla propria area privata, visualizzando lo stato dei documenti, i passaggi completati e quelli ancora da completare.
- RF04. Archiviazione dei documenti: L'utente "Admin" deve archiviare i documenti una volta completato il processo di workflow.

- RF05. Gestione degli utenti: L'utente deve poter gestire i propri dati utente, tra cui le informazioni di registrazione e le informazioni aggiuntive inerenti al proprio profilo.

Requisiti non funzionali

- RNF01. Sicurezza: L'applicazione web deve garantire la sicurezza dei dati e dei documenti dell'utente, utilizzando protocolli di sicurezza e crittografia.
- RNF02. Affidabilità: L'applicazione web deve essere affidabile e disponibile in modo continuo per gli utenti.
- RNF03. Scalabilità: L'applicazione web deve essere in grado di gestire grandi volumi di dati e di documenti, di essere scalabile ed efficiente.
- RNF04. Performance: L'applicazione web deve avere una performance elevata, con tempi di risposta rapidi per l'utente.
- RNF05. Interfaccia utente: L'interfaccia utente dell'applicazione web deve essere intuitiva e user-friendly, garantendo un'esperienza utente positiva.
- RNF06. Compatibilità: L'applicazione web deve essere compatibile con diversi browser web e dispositivi, inclusi computer desktop, tablet e smartphone.
- RNF07. Architettura: L'applicazione web deve avere un'architettura modulare e scalabile, con la possibilità di aggiungere nuove funzionalità in futuro.

3.4 DIAGRAMMA DEL PROCESSO PRINCIPALE

Il diagramma per descrivere il processo principale è stato realizzato con il BPMN.

Il Business Process Model and Notation (BPMN) è uno standard per la modellizzazione dei processi di business sviluppato per fornire un linguaggio comune e condiviso per descrivere i processi aziendali in modo che i team di lavoro possano capirli e collaborare meglio per migliorarli.

In questo caso, il diagramma in BPMN illustrato nel paragrafo seguente è stato utilizzato per rappresentare il flusso di lavoro del processo automatizzato di firma digitale dei documenti all'interno dell'applicazione web sviluppata.

Il diagramma offre una visualizzazione grafica dei passaggi del processo e delle attività coinvolte, consentendo una migliore comprensione delle interazioni tra i vari componenti del sistema. Grazie alla rappresentazione del processo in BPMN, è possibile identificare facilmente eventuali problematiche e trovare soluzioni per migliorare l'efficienza del processo.

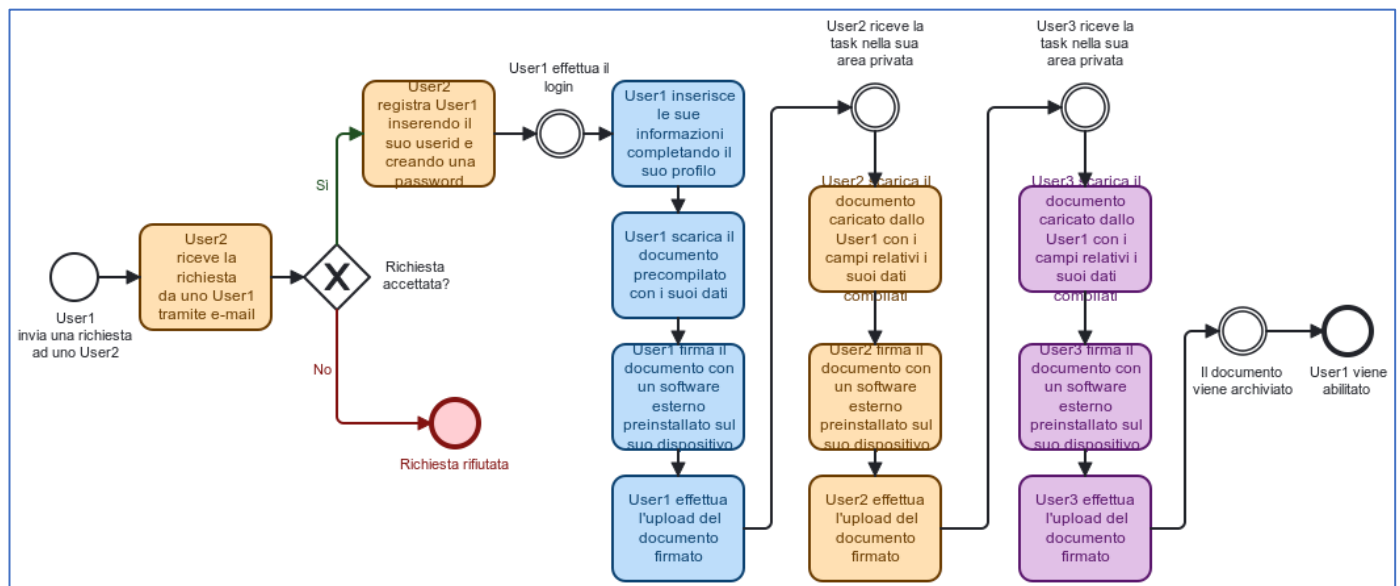


Figura 1 Diagramma BPMN del processo principale

3.5 VISTA ARCHITETTURALE DI MASSIMA

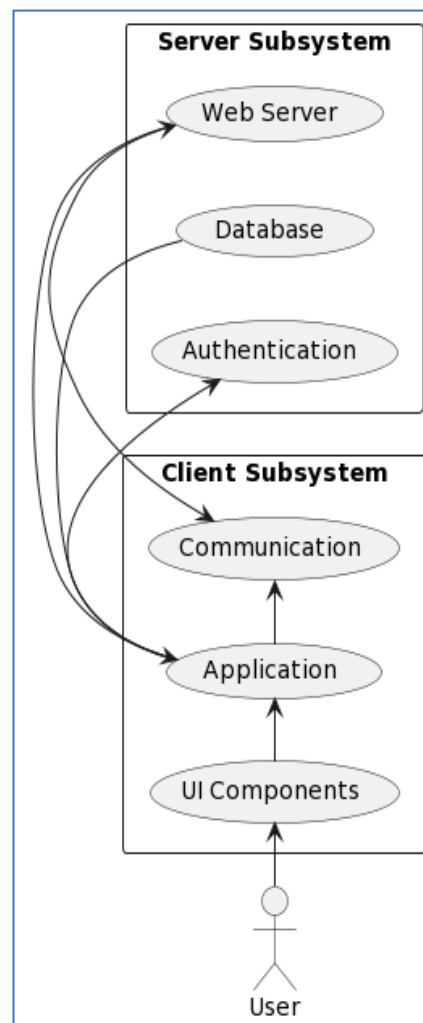


Figura 2 Vista architetturale del sistema

Questa vista architetturale rappresenta l'architettura a livello di sottosistemi dell'applicazione web che consiste di due sottosistemi: il sottosistema del Client e il sottosistema del Server.

Il sottosistema del Client gestisce la presentazione dei contenuti dell'applicazione all'utente. È composto da tre blocchi funzionali:

- **Interfaccia utente:** si occupa della presentazione dei contenuti dell'applicazione all'utente.
- **Logica del client:** gestisce la logica di interazione con l'utente, come la validazione dei dati inseriti.
- **Comunicazione con il server:** gestisce le richieste al server e le risposte ricevute.

Il sottosistema del Server gestisce il back-end dell'applicazione e la base di dati. È composto da quattro blocchi funzionali:

- Gestione dell'autenticazione: si occupa della gestione dell'autenticazione degli utenti e dell'autorizzazione ai servizi dell'applicazione.
- Web Server: fornisce i servizi web all'applicazione.
- Gestione del database: gestisce il database dell'applicazione.
- Applicazione: si occupa della logica dell'applicazione.

L'attore principale è l'utente, che interagisce con l'applicazione tramite il sottosistema del Client. Il sottosistema del Client comunica con il sottosistema del Server per fornire i servizi richiesti dall'utente. Il sottosistema del Server, a sua volta, gestisce l'autenticazione e l'autorizzazione degli utenti, gestisce il database dell'applicazione e fornisce i servizi web richiesti dal sottosistema del Client.

3.6 PROGETTAZIONE DELLA BASE DI DATI

3.6.1 PROGETTAZIONE CONCETTUALE

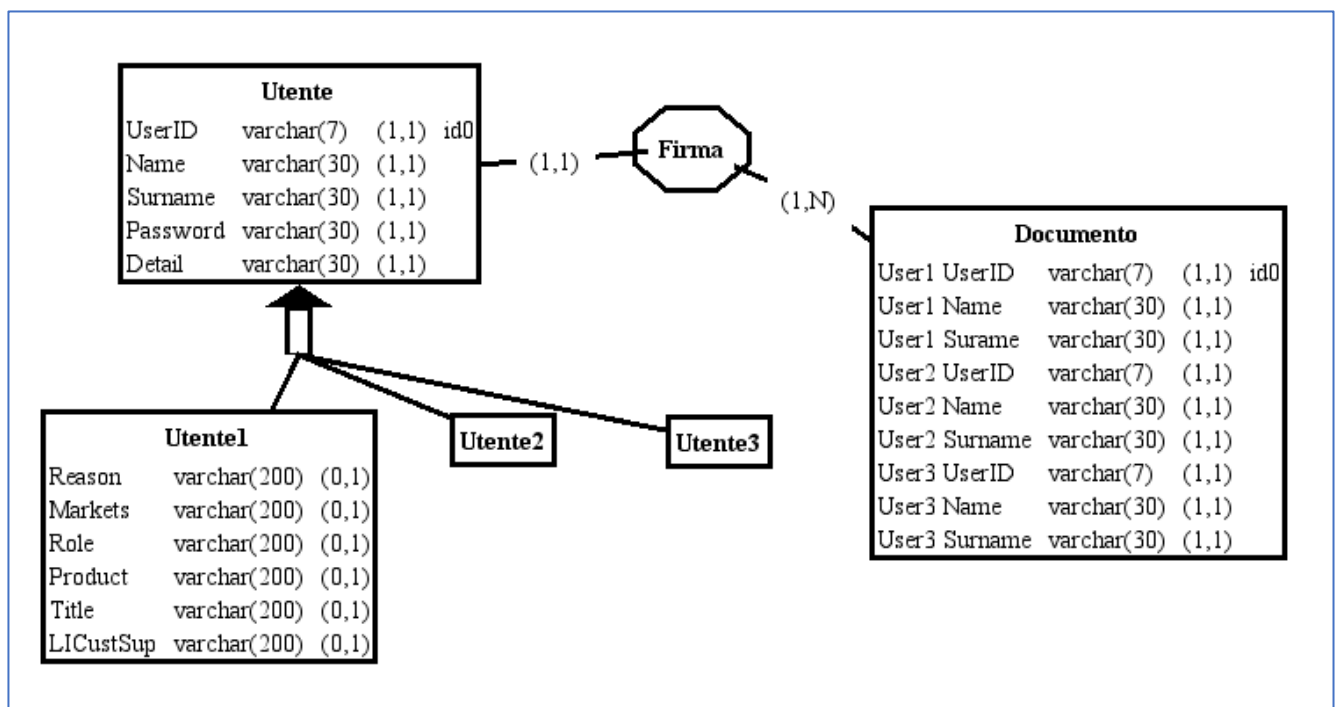


Figura 3 Schema concettuale della base di dati

Per la fase di progettazione concettuale ho deciso di adottare il modello Entity-Relationship.

Nella realizzazione dello schema ER ho adottato una strategia mista tra top-down e bottom-up partendo dai requisiti.

Ho deciso di applicare il Design Pattern “Part-of” poiché dai requisiti è emerso che ogni utente in questo contesto è “parte di” un documento.

Dizionario dei dati

Entità	Descrizione	Attributi	Identificatori
Documento	Documento associato ai tre attori Utente1, Utente2, Utente3	User1 Userid, User1 Name, User1 Surname, User2 Userid, User2 Name, User2 Surname, User3 Userid, User3 Name, User3 Surname	User1 UserID
Utente	Utente generico	UserID, Name, Surname, Password, Detail	UserID
Utente1	Utente che deve firmare per primo il documento	Reason, Role, Markets, LICustSup, Product, Title	UserID
Utente2	Utente che deve firmare per secondo il documento	-	UserID
Utente3	Utente che deve firmare per ultimo il documento	-	UserID

Tabella 1 Dizionario dei dati - Entità

Relazione	Descrizione	Entità coinvolte	Attributi
Firma	Associa gli utenti al documento	Utente, Utente1, Utente2, Utente3, Documento	-

Tabella 2 Dizionario dei dati - Relazioni

3.6.2 PROGETTAZIONE LOGICA

Analisi delle prestazioni: tavola dei volumi

Concetto	Tipo	Volume
Documento	E	1000
Utente	E	3000
Utente1	E	1000
Utente2	E	1000
Utente3	E	1000
Firma	R	3000

Tabella 3 Tavola dei volumi

Analisi delle prestazioni: tavola delle operazioni

Operazione	Tipo	Frequenza
Operazione 1: Registrazione di un nuovo utente	I	200 volte all'anno
Operazione 2: Selezione di un utente dato un certo userid e password (login)	I	600 volte all'anno
Operazione 3: Selezione di documenti relativi uno UserID	I	2000 volte all'anno
Operazione 4: Inserimento delle informazioni relative Utente1	I	200 volte all'anno
Operazione 5: Inserimento di dati relativi un documento	I	200 volte all'anno

Tabella 4 Tavola delle operazioni

Analisi delle ridondanze

Ho notato che una ridondanza è presente nell'entità Documento, in cui Name e Surname relativi alle tre tipologie di utenti possono essere ottenuti dall'entità Utente partendo dallo UserID.

Le operazioni che possono subire negativamente la presenza di questa ridondanza sono l'Operazione 3 e l'Operazione 5:

- Operazione 3: Selezione di documenti relativi uno UserID

Con ridondanza

Concetto	Costrutto	Accessi	Tipo
Documento	Entità	1000	L

Senza ridondanza

Concetto	Costrutto	Accessi	Tipo
Utente	Entità	3000	L
Firma	Relazione	3000	L
Documento	Entità	1000	L

Con il carico considerato, in presenza di ridondanza il costo dell'operazione richiede solo un accesso in lettura all'entità documento.

Il costo dell'operazione, calcolato su un regime annuale considerando la frequenza delle operazioni, equivale a circa 2000000 accessi annuali.

Considerando che in media nomi e cognomi hanno una lunghezza di 7 caratteri (7 byte), l'occupazione di memoria sarà di circa 42000 byte (42kB).

In assenza di ridondanza il costo dell'operazione richiede un accesso in lettura all'entità Utente²³ e Documento e alla relazione Firma. Considerando la frequenza delle operazioni, su un regime annuale il costo dell'operazione equivale a circa 14000000 di accessi annuali.

- Operazione 5: Inserimento di dati relativi un documento

Con ridondanza

Concetto	Costrutto	Accessi	Tipo
Documento	Entità	1	L
Documento	Entità	1	S

Senza ridondanza

Concetto	Costrutto	Accessi	Tipo
Utente	Entità	1	L
Firma	Relazione	1	L
Documento	Entità	1	L
Documento	Entità	1	S

Anche per l'operazione 5, eliminare la ridondanza, comporta un aumento degli accessi ragionevole.

Pertanto, si decide di mantenere la ridondanza in quanto riduce il numero di accessi, a discapito di una maggiore occupazione di memoria di circa 42kB, decisamente accettabile.

Eliminazione delle generalizzazioni

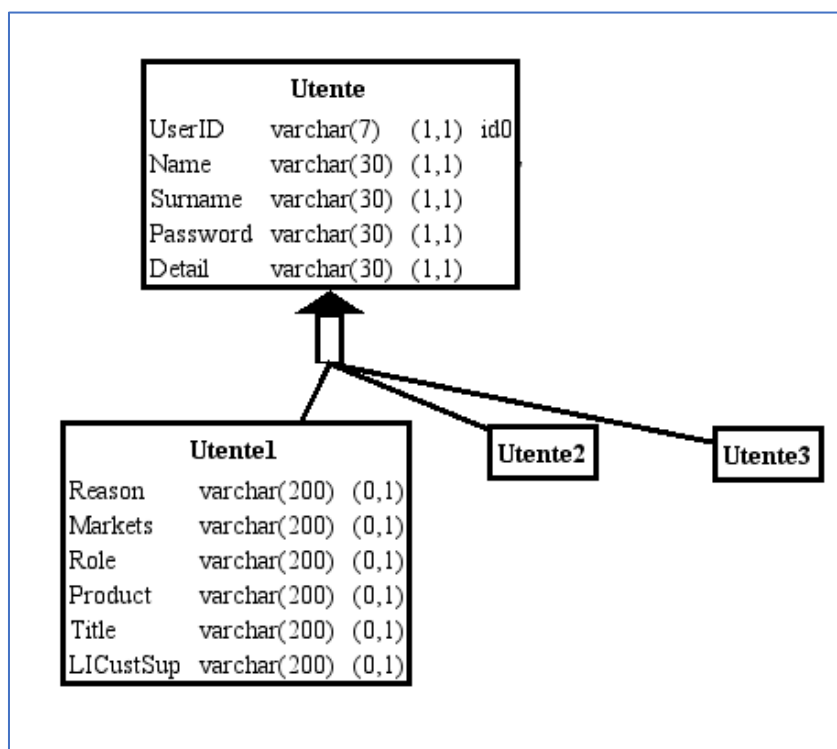


Figura 4 Schema concettuale senza generalizzazioni

Sulla base dei requisiti e del contesto applicativo decido di incorporare le specializzazioni nell'entità padre "Utente" inserendo "UserType" come attributo, e mantenere Utente1 che ha come attributi una serie di informazioni, un'entità separata "Informazioni":

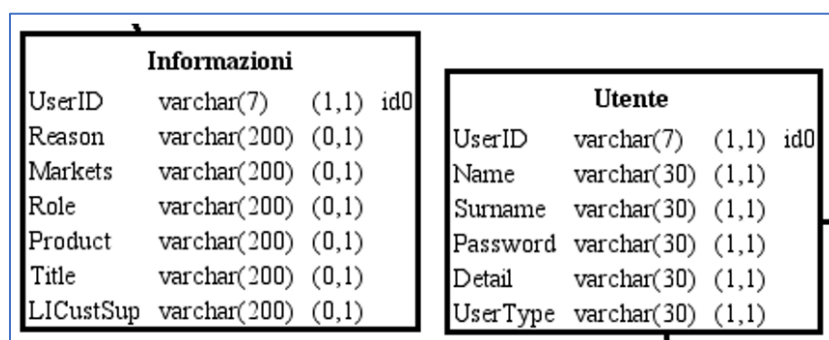


Figura 5 Entità "informazioni" ed "utente"

Schema ristrutturato finale

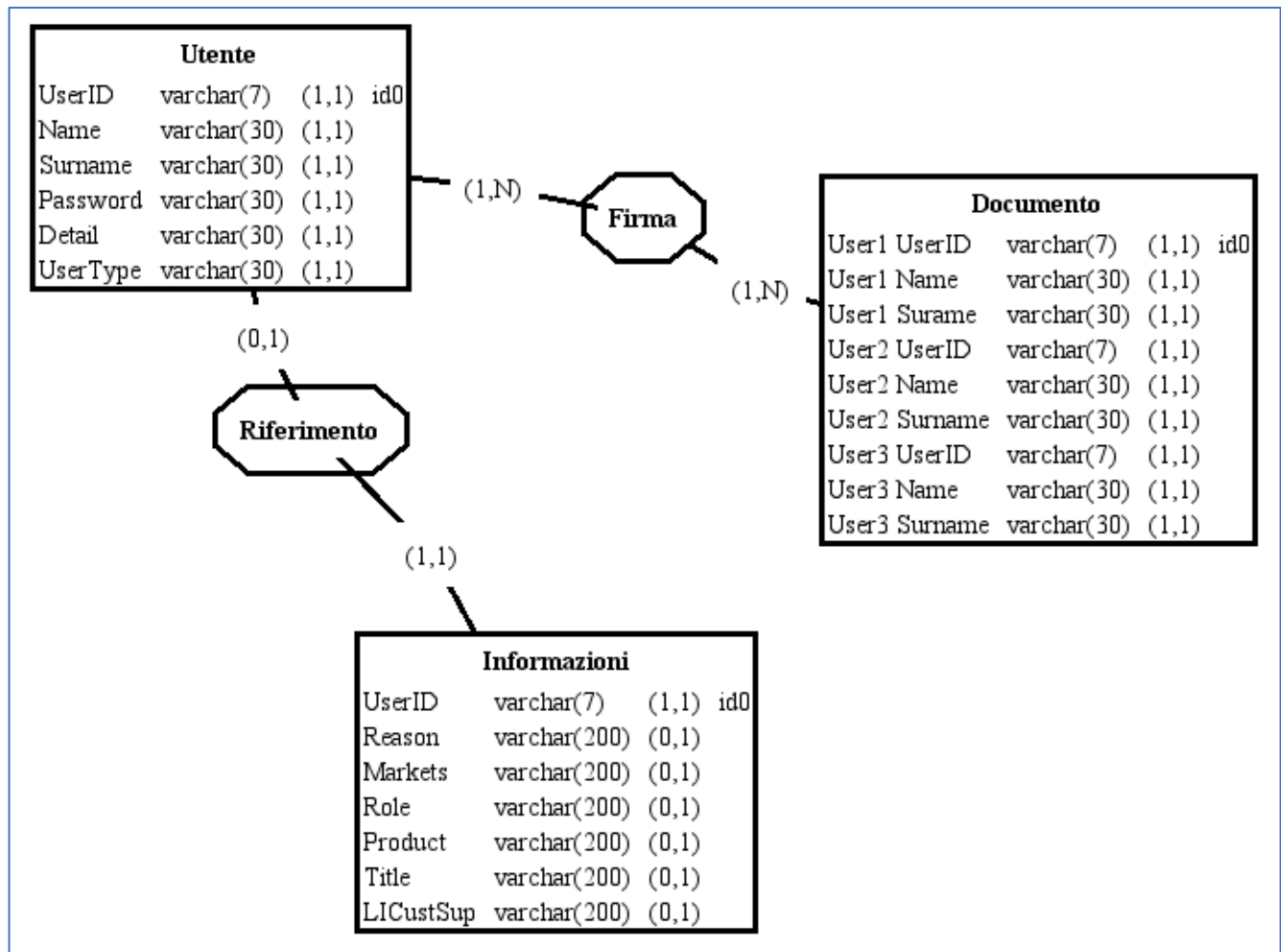


Figura 6 Schema concettuale ristrutturato

Schema logico

DOCUMENTO(user1_userid, user1_name, user1_surname, user2_userid, user2_name, user2_surname, user3_userid, user3_name, user3_surname)

FIRMA(user, document)

UTENTE(userid, name, surname, password, detail, userType)

INFORMAZIONI(userid, reason, markets, role, product, title, licustsup)

3.6.3 SCRIPT DI CREAZIONE DEL DATABASE

```
7 DROP TABLE IF EXISTS users CASCADE;
8 DROP TABLE IF EXISTS applicant CASCADE;
9 DROP TABLE IF EXISTS firma CASCADE;
10 DROP TABLE IF EXISTS ndamembers CASCADE;
11
12 CREATE TABLE users(
13     userid varchar(7) PRIMARY KEY,
14     name varchar(20) NOT NULL,
15     surname varchar(20) NOT NULL,
16     detail varchar(20) NOT NULL,
17     password varchar(100) NOT NULL,
18     user_type varchar(20) NOT NULL
19 )
20
21 CREATE TABLE applicant(
22     userid varchar(7) PRIMARY KEY,
23     reason varchar(100),
24     roleli varchar(100),
25     markets varchar(100),
26     licustsup varchar(100),
27     productcovered varchar(100),
28     CONSTRAINT fk_applicant FOREIGN KEY (userid) REFERENCES users(userid)
29     | | | | | on update CASCADE on delete RESTRICT
30 )
31
32 CREATE TABLE firma(
33     user varchar(7),
34     document varchar(7),
35     CONSTRAINT fk_user FOREIGN KEY (user) REFERENCES users(userid)
36     | | | | | on update CASCADE on delete RESTRICT,
37     CONSTRAINT fk_document FOREIGN KEY (document) REFERENCES ndamembers(applicant_userid)
38     | | | | | on update CASCADE on delete RESTRICT
39 )
40
41 CREATE TABLE ndamembers(
42     applicant_userid varchar(7),
43     applicant_name varchar(20),
44     applicant_surname varchar(20),
45     appmanager_userid varchar(7),
46     appmanager_name varchar(20),
47     appmanager_surname varchar(20),
48     csmlism_userid varchar(7),
49     csmlism_name varchar(20),
50     csmlism_surname varchar(20)
51 )
```

Figura 7 Script di creazione del database

3.7 TECNOLOGIE ABILITANTI

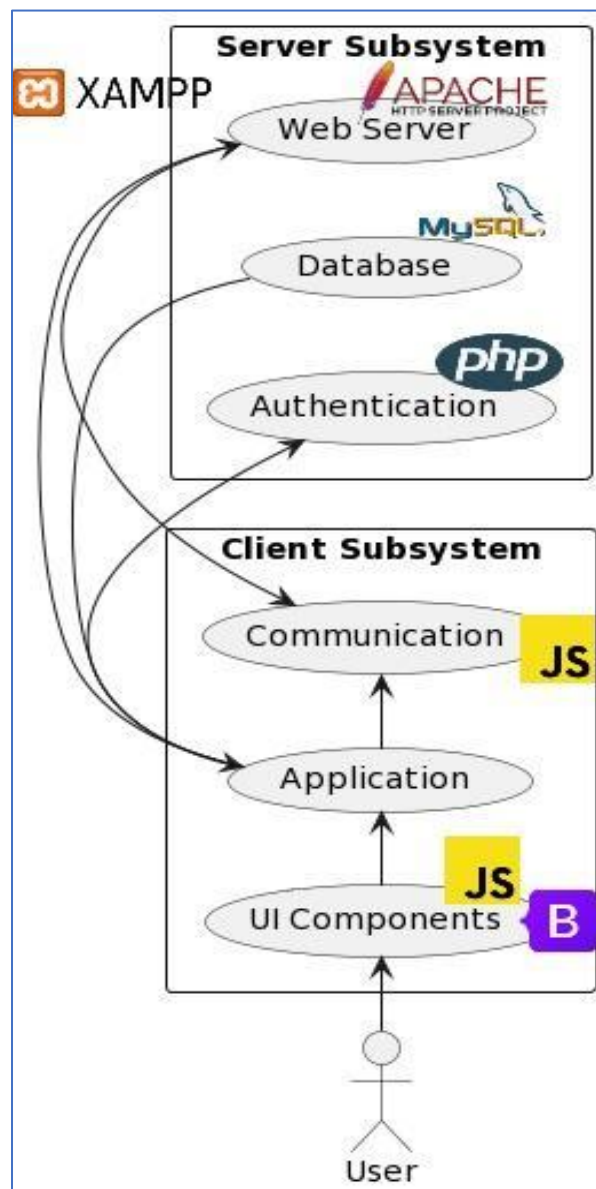


Figura 8 Tecnologie abilitanti

- **Apache:** Apache Web Server è un software open source che consente di gestire e distribuire siti web su Internet. Si tratta di uno dei server web più diffusi al mondo e viene utilizzato da molte aziende, organizzazioni e individui per ospitare siti web di diversa complessità.

Il software è stato sviluppato dalla Apache Software Foundation ed è disponibile per diverse piattaforme, tra cui Linux, Windows e macOS. L'Apache Web Server supporta una vasta gamma di protocolli di comunicazione, tra cui HTTP, HTTPS,

FTP e SMTP, e offre funzionalità avanzate per la gestione delle richieste dei client e per la sicurezza dei dati.

Tra le caratteristiche principali di Apache Web Server vi sono la possibilità di configurare il server attraverso file di configurazione, la compatibilità con i moduli esterni per estendere le funzionalità del server e la capacità di gestire grandi quantità di traffico web. Inoltre, Apache Web Server supporta molte tecnologie web, tra cui CGI, Perl, PHP e Ruby on Rails, consentendo ai webmaster di utilizzare i linguaggi di programmazione preferiti per sviluppare applicazioni web.

- **PHP:** PHP (acronimo di "Hypertext Preprocessor") è un linguaggio di programmazione open source utilizzato principalmente per lo sviluppo di applicazioni web dinamiche. Esso viene eseguito lato server, il che significa che il codice PHP viene elaborato dal server prima che la pagina web venga inviata al client.

Il linguaggio PHP è stato sviluppato nel 1994 da Rasmus Lerdorf, e da allora è diventato uno dei linguaggi di programmazione web più popolari al mondo. PHP è compatibile con molte piattaforme, tra cui Linux, Windows e macOS, ed è supportato da una vasta comunità di sviluppatori che creano costantemente nuovi pacchetti di estensione e librerie di supporto.

PHP può essere utilizzato per una vasta gamma di applicazioni web, tra cui la creazione di siti web dinamici, la gestione di database, la creazione di forum e la gestione di contenuti. Il linguaggio è in grado di comunicare con una vasta gamma di server web, tra cui Apache, Nginx e IIS.

Tra i punti forti di PHP vi sono la flessibilità, la facilità di apprendimento, la scalabilità e la compatibilità con molte tecnologie web. PHP supporta anche molti framework web popolari, come Laravel, Symfony e CodeIgniter, che semplificano lo sviluppo di applicazioni web complesse.

- **JavaScript:** JavaScript è un linguaggio di programmazione ad alto livello utilizzato principalmente per lo sviluppo di applicazioni web lato client, ovvero per la gestione di funzionalità interattive all'interno di una pagina web.

È stato sviluppato da Brendan Eich nel 1995 ed è diventato uno dei linguaggi di programmazione più popolari al mondo grazie alla sua ampia compatibilità con i browser web e alla sua capacità di aggiungere interattività e dinamicità alle pagine web.

JavaScript è un linguaggio di scripting interpretato che può essere incorporato direttamente all'interno delle pagine HTML. Ciò significa che i programmi JavaScript vengono eseguiti dal browser del client e non richiedono l'installazione di alcun software o plugin aggiuntivo.

Viene utilizzato per una vasta gamma di applicazioni web, tra cui la creazione di animazioni, la gestione degli eventi dell'utente, la validazione dei dati del form e la creazione di applicazioni web avanzate, come le Single Page Application (SPA).

JavaScript supporta anche molte librerie e framework, tra cui jQuery, React e Angular, che semplificano lo sviluppo di applicazioni web complesse e aumentano la produttività degli sviluppatori.

- **Bootstrap:** Bootstrap è un framework front-end open source che fornisce un insieme di strumenti, modelli e componenti per lo sviluppo di siti web responsive e mobile-first. Bootstrap è stato sviluppato da Twitter ed è diventato uno dei framework front-end più popolari e utilizzati al mondo.

Bootstrap utilizza HTML, CSS e JavaScript per creare interfacce utente moderne e dinamiche, con particolare attenzione alla compatibilità con i dispositivi mobili e alla facilità di utilizzo per gli sviluppatori. Il framework offre una vasta gamma di componenti predefiniti, come bottoni, form, navigazione, modali e alert, che

possono essere facilmente personalizzati per adattarsi alle esigenze specifiche del progetto.

Inoltre, Bootstrap utilizza una griglia flessibile basata su 12 colonne per disporre i contenuti sulla pagina, semplificando la creazione di layout complessi e garantendo una buona leggibilità su diverse dimensioni di schermo. Inoltre, Bootstrap offre una vasta gamma di classi CSS per la gestione della tipografia, delle immagini e degli elementi interattivi, consentendo agli sviluppatori di creare interfacce utente attraenti e coerenti in modo rapido e intuitivo.

Un'altra importante caratteristica consiste nel fatto che Bootstrap è anche altamente personalizzabile, consentendo agli sviluppatori di scegliere tra diverse opzioni di stile, colori e layout per creare siti web unici e personalizzati. Inoltre, Bootstrap offre una vasta gamma di plugin e librerie JavaScript, come jQuery e Popper.js, per la gestione di funzionalità avanzate come le animazioni e le tooltip.

3.8 IMPLEMENTAZIONE

L'applicazione web sviluppata offre una serie di funzionalità che permettono agli utenti di svolgere molte attività del workflow in modo efficiente e intuitivo.

Connessione al database

Il file “dbconnection.php” include il codice per stabilire una connessione con il database, utile ogni qual volta sia richiesto di scrivere o leggere dati all'interno di esso.


```

1  <?php
2      // Dati del database per la connessione
3      $host = "127.0.0.1";
4      $user = "root";
5      $password = "";
6      $database = "ericsson2";
7
8      $connection = new mysqli($host, $user, $password, $database);
9
10     if($connection === false){
11         die("Database connection error");
12     }else{
13         //echo "Successfully connected to database:".$connection->host_info;
14     }
15
16  ?>

```

Figura 9 Codice per stabilire una connessione con il database

Logout

Il file “logout.php” contiene poche righe di codice per terminare la sessione e reindirizzare l’utente nella pagina di login ogni qual volta venga richiesto il logout.

```

<?php
    session_start();

    if(session_destroy()){
        header("location: ../index.php");
    }

?>

```

Figura 10 Codice per terminare la sessione quando viene richiesto il logout

Login

Il frammento di codice sottostante è parte del file “index.php” contenente il codice della schermata di login.

È una funzionalità molto semplice ma vale la pena sottolineare le motivazioni per cui sono state fatte delle scelte implementative inerenti due dettagli:

- Utilizzo del prepared statement: L'utilizzo del prepared statement al posto di uno statement normale fornisce una maggiore sicurezza, prestazioni migliori e codice più pulito e manutenibile.

Un prepared statement precompila un'istruzione SQL in modo da poterla riutilizzare molte volte con diversi valori dei parametri. Ciò significa che il database ottimizza l'istruzione SQL e la memorizza in una cache, riducendo il tempo di elaborazione e aumentando le prestazioni complessive dell'applicazione.

Usare un prepared statement protegge da attacchi di SQL injection, una tecnica di hacking in cui gli aggressori inseriscono codice SQL dannoso in un'applicazione web, al fine di accedere, manipolare o eliminare i dati. Il prepared statement usa un'interfaccia sicura per passare i parametri alla query SQL, evitando che gli attaccanti inseriscano codice dannoso nel database.

Infine, l'utilizzo del prepared statement può semplificare il codice e renderlo più leggibile e manutenibile, poiché l'istruzione SQL e i parametri vengono separati in modo chiaro e distinto.

- Crittografia della password: Come algoritmo per crittografare le password ho scelto SHA1. SHA1 è un algoritmo di hash crittografico ampiamente utilizzato per la sicurezza delle password. È stato creato dalla National Security Agency (NSA) degli Stati Uniti ed è stato progettato per essere veloce, efficiente e sicuro.

L'algoritmo SHA1 prende in input una stringa di dati di qualsiasi lunghezza e restituisce un hash crittografico a 160 bit. Questo hash è un valore univoco e irreversibile, che rappresenta i dati di input. In altre parole, due stringhe di input diverse non possono generare lo stesso hash.

Ciò lo rende un meccanismo di crittografia sicuro per le password. Quando un utente immette una password, il sistema applica l'algoritmo SHA1 per creare l'hash della password. Questo hash viene quindi memorizzato nel database anziché la password in chiaro. Quando l'utente tenta di accedere, la password immessa viene nuovamente convertita in hash e confrontata con l'hash memorizzato nel database. Se i due hash corrispondono, l'utente ha inserito la password corretta.

Ci sono altri algoritmi di hash crittografici disponibili, ma SHA1 è ancora considerato uno standard di settore. Molti esperti di sicurezza lo considerano sicuro e affidabile per la crittografia delle password, a patto che siano rispettate alcune condizioni come la lunghezza della password e l'utilizzo di tecniche di salatura.

Inoltre, SHA1 è supportato da molte librerie di programmazione e linguaggi di scripting, rendendolo facile da implementare in molti sistemi. Tuttavia, è importante notare che ci sono anche algoritmi di hash più recenti e più sicuri, come SHA256 o SHA3, che potrebbero essere preferiti in alcune applicazioni.

```

4      require_once('PHP/dbconnection.php'); //Connessione al database (creazione di $connection)
5
6      $msg="";
7
8      if(isset($_POST['login'])){
9          //Salvo i valori del form nelle variabili
10         $userid = $_POST['userid'];
11         $password = $_POST['password'];
12         $password = sha1($password);
13         $userType = $_POST['userType'];
14
15         //Statement SQL (prepared)
16         $sql = "SELECT * FROM users WHERE userid=? AND password=? AND user_type=?";
17         $stmt=$connection->prepare($sql);
18         $stmt->bind_param("sss", $userid, $password, $userType);
19         $stmt->execute();
20         $result = $stmt->get_result();
21         $row = $result->fetch_assoc();
22
23         session_regenerate_id();
24         $_SESSION['userid'] = $row['userid'];
25         $_SESSION['role'] = $row['user_type'];
26         session_write_close();
27
28         if($result->num_rows==1 && $_SESSION['role']=="Applicant"){
29             header("location: ./applicant.php");
30         }else if($result->num_rows==1 && $_SESSION['role']=="Manager"){
31             header("location: ./manager.php");
32         }else if($result->num_rows==1 && $_SESSION['role']=="CSM"){
33             header("location: ./manager.php");
34         }else{
35             $msg = "Invalid credentials.";
36         }
37
38     }

```

Figura 11 Codice del login

Register

Il frammento di codice di sotto riportato è il codice PHP relativo alla registrazione di un nuovo “Utente1” in piattaforma. Questo codice è contenuto nel file “manager.php” poiché solo gli “Utente2” e gli “Utente3” possono registrare un nuovo “Utente1” al fine di fornirgli le credenziali di accesso e iniziare il flusso di lavoro.

Anche in questo caso per l’inserimento dei dati nelle tabelle del database è stato usato il prepared statement.

```

10 if (isset($_POST['submit'])) {
11     //Salvo i valori del form nelle variabili
12     $userid = $_POST['userid'];
13     $name = $_POST['name'];
14     $surname = $_POST['surname'];
15     $password = sha1($_POST['password']);
16     $detail = $_POST['appType'];
17     $type = "Applicant";
18     $sessionuserid = $_SESSION['userid'];
19
20     $sql_select = "SELECT * FROM users WHERE userid='$sessionuserid'";
21     if ($result = $connection->query($sql_select)) {
22         if ($result->num_rows == 1) {
23             $row = $result->fetch_array(MYSQLI_ASSOC);
24             $_SESSION['name'] = $row['name'];
25             $_SESSION['surname'] = $row['surname'];
26         }
27     }
28
29     $appmanageruserid = $_SESSION['userid'];
30     $appmanagername = $_SESSION['name'];
31     $appmanagersurname = $_SESSION['surname'];
32     $csmuserid = $_POST['csmliismuserid'];
33     $csmname = $_POST['csmliismname'];
34     $csmsurname = $_POST['csmliismsurname'];
35
36     //Statement SQL (prepared)
37     $sql = "INSERT INTO users() VALUES (?, ?, ?, ?, ?, ?)";
38     $stmt = $connection->prepare($sql);
39     $stmt->bind_param("ssssss", $userid, $name, $surname, $detail, $password, $type);
40     $stmt->execute();
41
42     $sql2 = "INSERT INTO ndamembers() VALUES (?, ?, ?, ?, ?, ?, ?)";
43     $stmt2 = $connection->prepare($sql2);
44     $stmt2->bind_param("ssssssss", $userid, $name, $surname, $appmanageruserid, $appmanagername, $appmanagersurname, $csmuserid, $csmname, $csmsurname);
45     $stmt2->execute();
46 }

```

Figura 12 Codice relativo la registrazione di un nuovo “Utente1” in piattaforma

Visualizzazione degli elementi con Javascript

Presente sia nell’area privata degli “Utente2” che in quella degli “Utente1” al fine di una visualizzazione più comoda degli elementi delle pagine

```

139 <script>
140     const button = document.getElementById("reg-newapp-btn");
141     const form = document.getElementById("insert-data-form");
142     const submit = document.getElementById("insert-data-submit");
143     const signbtn = document.getElementById("nda-button");
144
145     form.style.display = "none";
146
147     button.addEventListener("click", (event) => {
148         if (form.style.display == "none") {
149             form.style.display = "block";
150             button.style.display = "none";
151             signbtn.style.display = "none";
152         }
153     });
154     submit.addEventListener("click", (e) => {
155         form.style.display = "none";
156         button.style.display = "block";
157         signbtn.style.display = "block";
158     });
159 })
160 </script>

```

Figura 13 Codice JavaScript per la visualizzazione di elementi nella pagina manager.php

Generazione documento precompilato

Tramite il seguente codice in JavaScript gli “Utente1” possono scaricare il proprio documento precompilato secondo i dati contenuti nel database.

Il documento così precompilato sarà pronto per essere immesso nel processo di workflow ed essere firmato dai tre utenti.

```
195 let applicantFullName = applicantName + ' ' + applicantSurname
196 let managerFullName = managerName + ' ' + managerSurname
197 let csmFullName = csmName + ' ' + csmSurname
198
199 var today = new Date();
200 let day = today.getDate().toString();
201 let month = (today.getMonth() + 1).toString();
202 if ((today.getMonth() + 1) < 10) {
203   month = '0' + month;
204 }
205 let year = today.getFullYear().toString();
206 const date = day + '/' + month + '/' + year;
207
208
209 const { degrees, PDFDocument, rgb, StandardFonts } = PDFLib
210
211 async function modifyPdf() {
212   // Fetch an existing PDF document
213   const url = 'http://localhost/ericsson2/NDA.pdf'
214   const existingPdfBytes = await fetch(url).then(res => res.arrayBuffer())
215
216   // Load a PDFDocument from the existing PDF bytes
217   const pdfDoc = await PDFDocument.load(existingPdfBytes)
218
219   // Embed the Helvetica font
220   const helveticaFont = await pdfDoc.embedFont(StandardFonts.Helvetica)
221
222   // Get the first page of the document
223   const pages = pdfDoc.getPages()
224   const firstPage = pages[0]
225
226   // Get the width and height of the first page
227   const { width, height } = firstPage.getSize()
228
229   // Reason field
230   firstPage.drawText('X', {
231     x: 50,
232     y: height - 250,
233     size: 10,
234     font: helveticaFont,
235     color: rgb(0, 0, 0),
236   })
237   firstPage.drawText(reason, {
238     x: 50,
239     y: height - 290,
240     size: 8,
241     font: helveticaFont,
242     color: rgb(0, 0, 0),
243   })
244
245   // Applicant field
246   firstPage.drawText(applicantFullName, {
247     x: width - 100,
248     y: height - 140,
249     size: 10,
250     font: helveticaFont,
251     color: rgb(0, 0, 0),
252   })
253   firstPage.drawText(applicantUserId, {
254     x: width - 100,
255     y: height - 180,
256     size: 10,
257     font: helveticaFont,
258     color: rgb(0, 0, 0),
259   })
260
261   // Manager field
262   firstPage.drawText(managerFullName, {
263     x: width - 100,
264     y: height - 270,
265     size: 10,
266     font: helveticaFont,
267     color: rgb(0, 0, 0),
268   })
269   firstPage.drawText(managerUserId, {
270     x: width - 100,
271     y: height - 310,
272     size: 10,
273     font: helveticaFont,
274     color: rgb(0, 0, 0),
275   })
276
277   // Security Manager field
278   firstPage.drawText(csmFullName, {
279     x: width - 100,
280     y: height - 400,
281     size: 10,
282     font: helveticaFont,
283     color: rgb(0, 0, 0),
284   })
285   firstPage.drawText(csmUserId, {
286     x: width - 100,
287     y: height - 440,
288     size: 10,
289     font: helveticaFont,
290     color: rgb(0, 0, 0),
291   })
292
293   // Date field
294   firstPage.drawText(date, {
295     x: width - 100,
296     y: height - 650,
297     size: 10,
298     font: helveticaFont,
299     color: rgb(0, 0, 0),
300   })
301 }
```

Figura 14 Codice in JavaScript per la generazione del documento precompilato

I dati dal database vengono recuperati nel modo seguente:

```
39 $sql_select = "SELECT * FROM ndamembers WHERE userid='$userid'";
40 if ($result = $connection->query($sql_select)) {
41     if ($result->num_rows == 1) {
42         $row = $result->fetch_array(MYSQLI_ASSOC);
43         $_SESSION['name'] = $row['applicant_name'];
44         $_SESSION['surname'] = $row['applicant_surname'];
45         $_SESSION['manuserid'] = $row['appmanager_userid'];
46         $_SESSION['manname'] = $row['appmanager_name'];
47         $_SESSION['mansurname'] = $row['appmanager_surname'];
48         $_SESSION['csmuserid'] = $row['csmlism_userid'];
49         $_SESSION['csmname'] = $row['csmlism_name'];
50         $_SESSION['csmsurname'] = $row['csmlism_surname'];
51     } else {
52         echo "Errore.";
53     }
54 } else {
55     echo "Non hai completato il tuo profilo.";
56 }
57
58 echo "<script>const applicantUserId = " . json_encode($_SESSION['userid']) . "</script>";
59 echo "<script>const applicantName = " . json_encode($_SESSION['name']) . "</script>";
60 echo "<script>const applicantSurname = " . json_encode($_SESSION['surname']) . "</script>";
61 echo "<script>const reason = " . json_encode($_SESSION['reason']) . "</script>";
62 echo "<script>const managerUserId = " . json_encode($_SESSION['manuserid']) . "</script>";
63 echo "<script>const managerName = " . json_encode($_SESSION['manname']) . "</script>";
64 echo "<script>const managerSurname = " . json_encode($_SESSION['mansurname']) . "</script>";
65 echo "<script>const csmUserId = " . json_encode($_SESSION['csmuserid']) . "</script>";
66 echo "<script>const csmName = " . json_encode($_SESSION['csmname']) . "</script>";
67 echo "<script>const csmSurname = " . json_encode($_SESSION['csmsurname']) . "</script>";
```

Figura 15 Codice in PHP per recuperare dei dati dal database

CAPITOLO 4.

IL FUNZIONAMENTO DEL PROTOTIPO

4.1 DATI

In questa sezione viene fornita un'overview delle tabelle presenti nel database popolate, sia prima che dopo l'esempio mostrato nello scenario applicativo. In particolare, descriverò brevemente le informazioni contenute in ciascuna tabella e le relazioni che esistono tra di esse.

Tabella “users”

La tabella users contiene i dati per il login e un dettaglio che specifica se l'utente è “Interno” o “Esterno” all'azienda:

<div><div><div>←</div><div>T</div><div>→</div></div><div></div></div>													userid	name	surname	detail	password	user_type
<input type="checkbox"/>		Modifica		Copia		Elimina	EDOMDIS	Domenico	di Stasio	External	8cb2237d0679ca88db6464eac60da96345513964	Applicant						
<input type="checkbox"/>		Modifica		Copia		Elimina	EGIUVER	Giuseppe	Verdi	Internal	8cb2237d0679ca88db6464eac60da96345513964	CSM						
<input type="checkbox"/>		Modifica		Copia		Elimina	EMARROS	Mario	Rossi	Internal	8cb2237d0679ca88db6464eac60da96345513964	Manager						

Figura 16 Tabella “users”

Tabella “document/ndamembers”

La tabella “document” contiene le informazioni per compilare il documento. L'attributo “applicant_userid” è la chiave primaria della tabella poiché il documento è concettualmente relativo all'Utente1 (applicant):

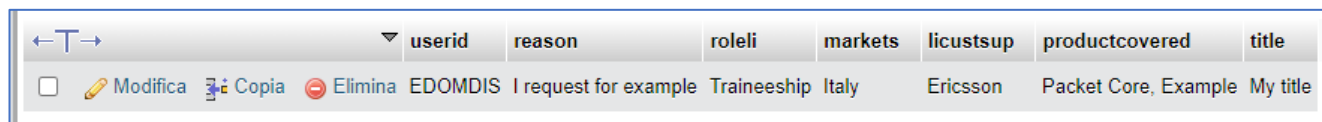


applicant_userid	applicant_name	applicant_surname	appmanager_userid	appmanager_name	appmanager_surname	csmlism_userid	csmlism_name	csmlism_surname
ESEMPIO	Sem	Pio	EANTFRE	Antonio	Fresa	ECSMLIS	Csmname	Csmsurname
EDOMDIS	Domenico	di Stasio	EMARROS	Mario	Rossi	EGIUVER	Giuseppe	Verdi

Figura 17 Tabella “document”

Tabella “applicant/informazioni”

La tabella “applicant” contiene le informazioni aggiuntive richieste per poter abilitare l’Utente1 nel modo corretto. Inoltre, l’attributo “reason” contiene un valore che verrà inserito nel documento:

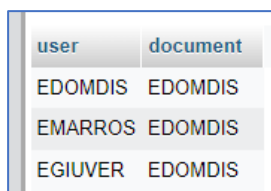


	userid	reason	roleli	markets	licustsup	productcovered	title
<input type="checkbox"/>	EDOMDIS	I request for example	Traineeship	Italy	Ericsson	Packet Core, Example	My title

Figura 18 Tabella “applicant”

Tabella “firma”

La tabella firma contiene le associazioni Utente-Documento. L’attributo “user” contiene il riferimento all’attributo “userid” della tabella “users” e l’attributo “document” contiene il riferimento all’attributo “applicant_userid” della tabella “document”:



user	document
EDOMDIS	EDOMDIS
EMARROS	EDOMDIS
EGIUVER	EDOMDIS

Figura 19 Tabella “firma”

4.2 SCENARIO APPLICATIVO

Questa sezione è incentrata sull'analisi dello scenario applicativo attraverso un caso studio dettagliato. Viene mostrato come un utente interagisce con il prototipo e le azioni che l'utente compie per raggiungere i propri obiettivi.

Attraverso questo caso studio, viene illustrato in maniera più profonda il funzionamento dell’applicazione web. Inoltre, una visione pratica su un caso d’uso del prototipo mostra come le azioni dell'utente possono essere ottimizzate per migliorare l'esperienza complessiva.

Inizio del workflow

L'Utente2 riceve la richiesta di essere abilitato da parte dell'Utente1. Utente2 accetta la richiesta e inizia la procedura di registrazione per l'Utente1:

- Utente2 effettua il login e accede alla sua area privata:

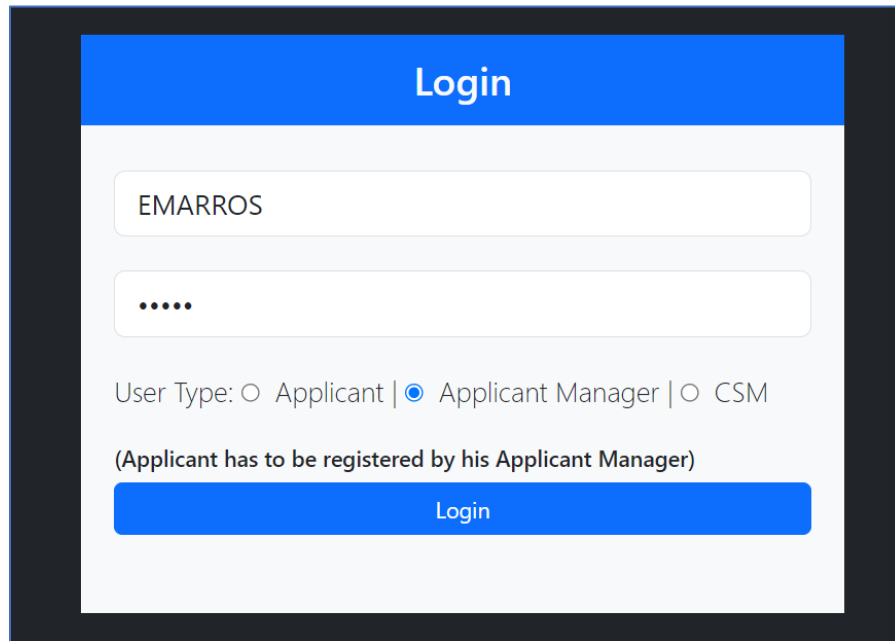
A login form with a blue header bar containing the word "Login". Below the header, there is a white input field containing the text "EMARROS". Underneath that is another white input field containing five dots. Below the input fields, there is a line of text: "User Type: ☐ Applicant | ☒ Applicant Manager | ☐ CSM". Below this text is a note in parentheses: "(Applicant has to be registered by his Applicant Manager)". At the bottom of the form is a blue button with the text "Login".

Figura 20 Login form

- Dall'area privata, Utente2 clicca su "Register new applicant":

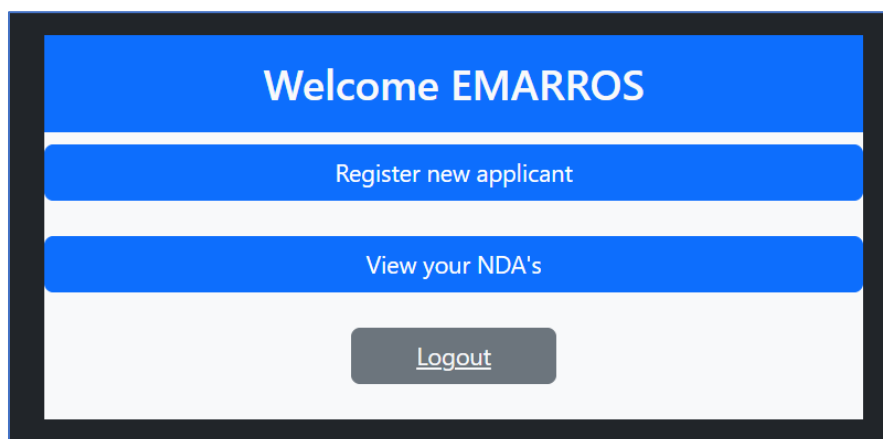
A private area interface with a blue header bar containing the text "Welcome EMARROS". Below the header, there are two blue buttons: "Register new applicant" and "View your NDA's". At the bottom of the interface is a grey button with the text "Logout".

Figura 21 Area privata Utente2

- Utente2 inserisce i dati dell'Utente1 e i dati dell'Utente3 associato, in modo da creare l'associazione per il documento nel database:

The image shows two identical registration forms side-by-side, both titled "Welcome EMARROS". Each form contains the following fields and controls:

- Inser UserID of the new applicant:** A text input field. In the left form, it contains "es. ENAMSUR"; in the right form, it contains "EDOMDIS".
- Name:** A text input field. In the left form, it contains "Insert name"; in the right form, it contains "Domenico".
- Surname:** A text input field. In the left form, it contains "Insert surname"; in the right form, it contains "di Stasio".
- Create a password:** A text input field. In the left form, it is empty; in the right form, it contains "*****".
- Applicant Type:** Radio buttons for "Internal" and "External". The "External" option is selected in both forms.
- Inser UserID of CSM/LISM:** A text input field. In the left form, it contains "es. ENAMSUR"; in the right form, it contains "EGIUVER".
- CSM/LISM Name:** A text input field. In the left form, it contains "Insert name"; in the right form, it contains "Giuseppe".
- CSM/LISM Surname:** A text input field. In the left form, it contains "Insert surname"; in the right form, it contains "Verdi".
- Submit:** A blue button at the bottom of each form.

Figura 22 Registration form

Utente1 riceve una mail di avvenuta registrazione dall'Utente2

- Utente1 effettua il login ed entra nella sua area privata:

The image shows a private area dashboard titled "Welcome EDOMDIS". It contains the following elements:

- Header:** A blue bar with the text "Welcome EDOMDIS".
- Instruction Box:** A white box with a blue border containing the text: "In this page you can complete your profile inserting all your info's to generate your NDA document and upload the NDA digitally signed."
- Buttons:** Four buttons are arranged vertically:
 - "Insert your info's" (blue button)
 - "Generate NDA" (blue button)
 - "Upload signed NDA" (white button with a black border)
 - "Logout" (grey button)

Figura 23 Area privata Utente1

- Tramite “Insert your info’s” l’Utente1 completa il proprio profilo prima di poter generare il documento:

Figura 24 Information form

- Tramite “Generate NDA” l’Utente1 ottiene il proprio documento precompilato contenente i dati reperiti dal database:

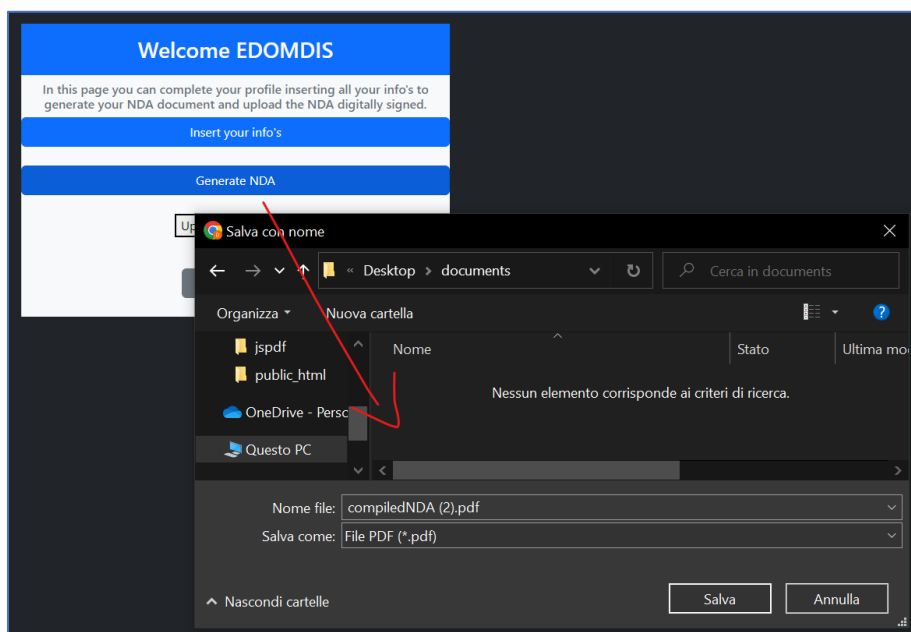


Figura 25 Generazione del documento

- Ora l'Utente1 ha il documento precompilato da firmare con un software dell'azienda e caricare tramite "Upload signed NDA"

Fine del workflow

Quando l'Utente1 carica il documento firmato, l'Utente2 può scaricarlo dalla sua area privata e ripetere il procedimento di firma e upload.

L'Utente3, nello stesso modo dell'Utente2 ripete il procedimento, il documento contenente tutte le firme viene archiviato e l'Utente1 viene abilitato.

4.3 CONCLUSIONI E SVILUPPI FUTURI

Durante il corso del progetto è stato sviluppato un prototipo di un'applicazione Web con l'obiettivo di automatizzare il workflow documentale di un processo aziendale della Ericsson Telecomunicazioni S.p.A.

Questo progetto si è rivelato coerente con le analisi svolte a priori e i requisiti stabiliti durante la fase preliminare; tuttavia, alcune funzionalità non sono state implementate.

In particolare, è possibile identificare una serie di funzionalità che potrebbero essere implementate in futuri sviluppi dell'applicazione.

Ad esempio, sarebbe auspicabile la realizzazione di una pagina dei tasks nell'area privata dei Manager, in modo che essi possano visualizzare i documenti da firmare e lo stato del processo. In questo modo, si avrebbe un maggiore controllo sulle attività in corso e si potrebbero identificare eventuali problemi o ritardi.

Un'altra funzionalità che potrebbe essere utile per l'applicazione sarebbe quella di implementare un metodo di archiviazione sicuro per i documenti che hanno terminato il flusso di lavoro. Questo permetterebbe di mantenere un archivio affidabile e sicuro dei documenti aziendali, garantendo al contempo la loro disponibilità e la loro integrità nel tempo.

Infine, una terza funzionalità che potrebbe essere implementata in futuro sarebbe quella di integrare un sistema di reminder per gli utenti che non completano i task entro

un certo intervallo di tempo. In questo modo, si potrebbe migliorare la tempestività del processo e ridurre il rischio di eventuali ritardi o problemi.

Bibliografia

- [1] WfMC. Wf-XML2.0 .2004
- [2] WfMC. Workflow Reference Model. Document Number WFMC-TC00-1003 1995
- [3] SHIMeiLin,YANGuangXin,XIANYong,WUShangGuang. WfMS : Workflow management system. Chinese Journal of Computers 1999 Vol.20 No.3 P.325-334

Indice delle Figure

Figura 1 Diagramma BPMN del processo principale	19
Figura 2 Vista architetturale del sistema	20
Figura 3 Schema concettuale della base di dati	21
Figura 4 Schema concettuale senza generalizzazioni	26
Figura 5 Entità “informazioni” ed “utente”	26
Figura 6 Schema concettuale ristrutturato	27
Figura 7 Script di creazione del database	28
Figura 8 Tecnologie abilitanti	29
Figura 9 Codice per stabilire una connessione con il database	33
Figura 10 Codice per terminare la sessione quando viene richiesto il logout	33
Figura 11 Codice del login	36
Figura 12 Codice relativo la registrazione di un nuovo “Utente1” in piattaforma	37
Figura 13 Codice JavaScript per la visualizzazione di elementi nella pagina manager.php	37
Figura 14 Codice in JavaScript per la generazione del documento precompilato	38
Figura 15 Codice in PHP per recuperare dei dati dal database	39
Figura 16 Tabella “users”	40
Figura 17 Tabella “document”	40
Figura 18 Tabella “applicant”	41
Figura 19 Tabella “firma”	41
Figura 20 Login form	42

Figura 21 Area privata Utente2	42
Figura 22 Registration form	43
Figura 23 Area privata Utente1	43
Figura 24 Information form	44
Figura 25 Generazione del documento	44

Indice delle Tabelle

Tabella 1 Dizionario dei dati - Entità	22
Tabella 2 Dizionario dei dati - Relazioni	23
Tabella 3 Tavola dei volumi	23
Tabella 4 Tavola delle operazioni	23

Ringraziamenti

I ringraziamenti più sentiti e sinceri alle persone che mi hanno supportato durante tutto il percorso di studi.

Ringrazio il professore Francesco Moscato per avermi seguito durante la stesura dell'elaborato.

Ringrazio i miei genitori, che mi hanno dato la possibilità di intraprendere questo percorso di studi e di portarlo a termine, rispettando i miei tempi e ogni mia scelta.

Ringrazio i miei nonni, per avermi sostenuto. Un ringraziamento speciale a Nonna Angela, che non ha mai smesso di crescermi e prendersi cura di me. Grazie a lei ho imparato a superare ogni difficoltà.

Ringrazio Chiara, per avermi trasmesso la sua forza quotidianamente.

Ringrazio Ludovica, per essermi stata vicina e avermi supportato nei periodi più delicati, per aver condiviso con me ogni successo e ogni tentativo, per essere stata un riferimento e per aver creduto in me.

Ringrazio tutti i miei zii, che mi hanno sempre voluto bene come un figlio.

Ringrazio mio cugino Riccardo, per essere stato come un fratello in un periodo buio.

Ringrazio tutti i miei amici, che mi hanno regalato dei momenti indimenticabili e con naturalezza sono sempre stati in grado di farmi dimenticare di ogni problema.

Ringrazio tutti i rappresentanti di StudentIngegneria, per il lavoro che hanno svolto e che svolgono ogni giorno per aiutare gli studenti.

Ringrazio chi mi è stato vicino quando era più difficile, per aver contribuito a ciò che sono, ringrazio chi mi ha accompagnato ad iniziare questo percorso e chi mi ha accompagnato a terminarlo.