



Università degli Studi di Padova

Facoltà di Ingegneria

Corso di Laurea in Ingegneria Informatica

tesi di laurea

Analisi e progettazione di un sistema di gestione delle presenze

Relatore: Massimo Rumor

Laureando: Davide Zannirato

23 Settembre 2011

Ai miei genitori,

che mi hanno sempre sostenuto e motivato.

Sommario

Questa tesi rappresenta la relazione di fine tirocinio, il cui scopo finale è la progettazione e lo sviluppo di una web application con funzioni di rilevazione presenze e gestione del personale di un'azienda, imparando ad utilizzare gli strumenti necessari, come linguaggi di programmazione e framework, sotto la guida del tutor aziendale.

Al giorno d'oggi software con funzioni di questo tipo sono utilizzati sempre in più realtà lavorative, con benefici quantificabili subito come la riduzione del tempo impiegato per la timbratura e una migliore gestione dei dati, ottimizzando i compiti della segreteria, specie per la totalizzazione delle ore di lavoro e la preparazione degli stipendi.

Questi miglioramenti sono dovuti all'eliminazione del supporto cartaceo: i dipendenti non devono più segnare con penna orari di ingresso o uscita e la segreteria non deve sobbarcarsi di conti per ricavare il quantitativo netto di ore lavorative, tutto viene automatizzato.

L'attività svolta si divide in due parti, la prima è rappresentata da uno studio di fattibilità del progetto, dove vengono analizzati i requisiti raccolti a seguito di colloqui con il committente e individuate le principali funzioni richieste.

Inoltre tramite un'analisi del mercato viene ricercato software in grado di soddisfare tali caratteristiche, i prodotti già esistenti però sono troppo complessi rispetto alle reali necessità e per adattarli sarebbero necessari degli interventi di personalizzazione a pagamento.

Per garantire maggiore conformità alle richieste del cliente si opta quindi per lo sviluppo ad-hoc dell'applicazione, integrando le sole funzionalità richieste per evitare confusione durante il suo utilizzo.

La seconda parte dell'attività svolta concerne la realizzazione del software, iniziata con il delineare con più precisione i requisiti attraverso gli schemi dei casi d'uso dell'applicativo e con la progettazione del modello dati adatto al contesto.

Successivamente vengono scelti gli strumenti da utilizzare per la programmazione, descrivendone le varie caratteristiche, e pianificate le attività da svolgere per raggiungere il prodotto finale.

Infine nella fase di sviluppo viene costruita l'applicazione vera e propria, allegando delle anteprime dell'applicativo terminato.

Autore: Davide Zannirato

Indice

Sommario	V
1 Studio di fattibilità	1
1.1 Azienda ospitante	1
1.2 Obiettivi	1
1.3 Analisi dei Requisiti	2
1.4 Analisi del mercato	4
1.4.1 Soluzioni Proprietarie	4
1.4.2 Soluzioni FOSS	7
1.4.3 Conclusioni	9
2 Realizzazione	11
2.1 Progettazione di dettaglio con approfondimento dei requisiti	11
2.1.1 Casi d'uso	11
2.1.2 Modello dati	15
2.2 Strumenti utilizzati	15
2.2.1 PHP 5.3	18
2.2.2 Symfony 2.0	18
2.2.3 Doctrine	21
2.2.4 AdminBundle	21
2.2.5 MySQL 5.5	21
2.2.6 jQuery	22
2.2.7 GIT	22
2.3 Sviluppo del software	23
2.3.1 Preparazione piattaforma LAMP	23
2.3.2 Creazione Database	23
2.3.3 Sviluppo interfaccia di Front-end	24
2.3.4 Sezione di Login	25
2.3.5 Integrazione della dashboard di AdminBundle	27
2.3.6 Generazione del Cartellino	30
2.3.7 Esportazione in CSV	31
3 Conclusioni	35

Bibliografia	36
Elenco delle tabelle	38
Elenco delle figure	40
Ringraziamenti	42

Capitolo 1

Studio di fattibilità

1.1 Azienda ospitante

3DGIS srl è una software house con sedi a Rovigo e a Padova nelle quali operano giovani Ingegneri Informatici dell'Università di Padova.

Si tratta di una società giovane, ma costruita su un'esperienza pluriennale nel campo della progettazione e dello sviluppo di sistemi informativi territoriali e in generale di soluzioni nell'area ITC e GeoITC, rivolte soprattutto ad enti pubblici che hanno bisogno di sostegno per il mantenimento e gestione di dati territoriali.

Vengono offerte soluzioni per visualizzare e gestire le reti fognarie, idriche, stradali e dell'illuminazione pubblica, dal punto di vista tecnico ed economico, software per la visualizzazione tridimensionale dei dati geografici e per l'interrogazione e renderizzazione di database topografici.

In generale però viene trattato qualsiasi tipo di software, con possibilità di sviluppo sia in ambiente Open Source sia su piattaforme commerciali, in base alle esigenze rilevate a seguito di colloqui con il cliente.

1.2 Obiettivi

Lo scopo del tirocinio verte sull'analisi, progettazione e sviluppo di una web application commissionata da un'azienda di piccole/medie dimensioni erogatrice di servizi, che tramite colloqui ha specificato dei requisiti da soddisfare.

Si procederà inizialmente con uno studio di fattibilità, per valutare se è possibile acquistare e utilizzare un prodotto già esistente sul mercato o se è preferibile svilupparlo *ad-hoc*.

In seguito verrà progettato e sviluppato un prototipo funzionante dell'applicazione, scegliendo gli strumenti di programmazione adatti ed integrando le funzionalità richieste, precedentemente analizzate ed elaborate, fino al raggiungimento di un prodotto in linea con le richieste dell'utilizzatore finale.

1.3 Analisi dei Requisiti

A seguito di colloqui effettuati con il committente del software, sono stati definiti dei requisiti che il prodotto finale dovrà soddisfare.

Requisiti funzionali

Rappresentano le caratteristiche delle operazioni eseguibili tramite l'applicazione.

Interfaccia di front-end Da utilizzare in monitor touch-screen come chiosco fisso per la timbratura, azione che deve essere rapida, per far fronte ad eventuali code causate da orari di lavoro comuni ai dipendenti.

Graficamente deve essere minimale, con tastierino numerico per l'identificazione dell'utente e un orologio con orario corrente.

Questa schermata dev'essere raggiungibile da ogni computer connesso alla lan aziendale, in questo modo i dipendenti, terminato l'orario di lavoro, possono timbrare l'uscita direttamente dalla loro postazione.

Area personale Accessibile da ogni computer collegato alla rete interna dell'azienda tramite autenticazione con username e password, modificabili dal possessore in ogni momento attraverso l'area personale e in caso di dimenticanza l'utente si rivolgerà alla segreteria per la modifica.

In questa sezione possono essere invocate le funzioni alle quali l'utente connesso ha accesso, a seconda del suo ruolo.

Gli amministratori hanno la possibilità di accedere a tutti i dati presenti e di modificarli, inoltre possono creare nuovi utenti e assegnare loro un determinato profilo, creare nuovi ruoli aziendali o nuove tipologie di timbratura.

I normali utenti d'altro canto hanno accesso solo alla visualizzazione dei propri dati, senza la possibilità di modificarli.

Cartellino Funzionalità richiamabile dall'area personale, viene generato un riassunto delle timbrature di un utente in un determinato periodo temporale, con il formato della tabella 1.1.

<i>NomeUtente</i>	<i>Giorno</i>	<i>IN</i>	<i>OUT</i>	<i>IN</i>	<i>OUT</i>
	20/06/2011	07:55	12:03	13:00	17:08
	21/06/2011	08:00	12:01		
	22/06/2011	08:03		13:11	17:04

Tabella 1.1: Esempio di Cartellino

Il cartellino è composto dal nome del soggetto in questione e per ogni giorno la serie di coppie di orari ingresso/uscita relativi a quel giorno, con una

cella vuota in caso di mancanza e in quel caso solo l'amministrazione può aggiungerlo.

L'arco temporale analizzato nel cartellino può essere scelto tra "Settimanale", "Mensile" oppure "Personalizzato".

La segreteria per ovvi motivi legati al calcolo delle ore lavorative degli stipendi può richiedere cartellini di qualsiasi utente, mentre i dipendenti possono solo generarne di propri.

Esportazione Per ogni cartellino visualizzato dev'essere integrata la possibilità di salvarlo in locale in un file CSV, con le stesse impostazioni di formattazione della tabella 1.1 già analizzata.

Requisiti non funzionali

Ovvero i vincoli e le caratteristiche di utilizzo che il prodotto deve soddisfare.

Software È richiesta una web application accessibile tramite la rete interna di un'azienda di piccole/medie dimensioni per gestire le timbrature dei dipendenti e facilitare le mansioni della segreteria.

Per il mantenimento dei dati deve essere progettato un database apposito.

Usabilità L'applicazione deve avere interfaccia intuitiva e di facile comprensione anche per i meno avvezzi alla tecnologia, l'utilizzo deve essere veloce e le funzioni invocabili con pochi e semplici comandi.

Scalabilità Il software deve essere sviluppato con strumenti e metodologie che garantiscano una futura aggiunta di funzionalità o operazioni sui dati, a tale scopo viene raccomandato l'utilizzo di software Open-Source.

Efficienza ed efficacia Le funzioni implementate devono avere tempi di risposta brevi, soddisfacendo le richieste dell'utilizzatore con i risultati adeguati.

Sicurezza I dati confidenziali, come le password, devono essere protetti, inoltre ogni utente può accedere solo ai dati e alle funzioni abilitate per il suo profilo.

Utenti Hanno un ruolo interno all'azienda che ne identifica i privilegi.

Inizialmente sono richiesti i profili "Utente" e "Amministratore", ma nulla vieta aggiunte future, a seconda delle esigenze interne.

Timbrature Ognuna appartiene ad un singolo utente e deve contenere informazioni su data e ora di creazione. Si distinguono 2 tipi principali: "Ingresso" e "Uscita", ma non bisogna precludere la possibilità di aggiungerne altri a discrezione dell'azienda.

1.4 Analisi del mercato

La maggior parte dei prodotti esistenti sul mercato sono degli “Human Resources Management System” (HRMS), ovvero dei software multi-funzione per la gestione del personale (comprese timbrature e mansioni), delle commissioni e della gestione delle retribuzioni, di seguito vengono analizzati solo i principali prodotti, sia a livello nazionale che internazionale.

1.4.1 Soluzioni Proprietarie

Ovvero quei software che sono sviluppati da aziende private e rilasciati a terzi con restrizioni specificate nella licenza, riguardanti l'utilizzo, la modifica e la distribuzione del prodotto.

Solitamente il prezzo varia a seconda delle funzionalità presenti, infatti l'applicazione di base è espandibile con l'acquisto di moduli che offrono features aggiuntive.

La ditta sviluppatrice inoltre garantisce assistenza in caso di problemi e spesso la disponibilità ad effettuare personalizzazioni o adattamenti.

OrangeHRM Live - OrangeHRM inc.

Prodotto SaaS (Software as a Services), cioè l'applicazione viene rilasciata e resa disponibile dal produttore tramite internet direttamente sul browser web, senza il bisogno di mantenere in azienda un'infrastruttura server, sistemi di backup o tecnici addetti.

Non essendo necessaria alcuna installazione o configurazione in locale, il servizio viene reso disponibile in tempi brevissimi.

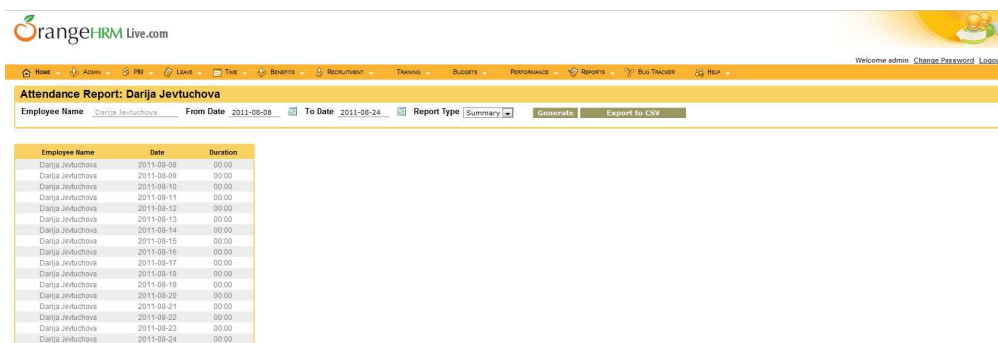
Il software è stato sviluppato in PHP e necessita di DBMS di tipo MySQL.

L'interfaccia è semplice e intuitiva, con la possibilità di generare dei totalizzatori esportabili in formati PDF o CSV, inoltre sono disponibili numerosi moduli da integrare nell'applicazione, per far fronte ad ogni esigenza aziendale.

I moduli interessanti per quanto riguarda i requisiti sono:

- *Admin module*: vengono aggiunte funzioni utili per l'amministrazione, come la gestione degli utenti, le loro informazioni, le loro mansioni, orari e altro, con possibilità di inviare notifiche.
- *Personal information module*: per inserire informazioni sulle capacità e conoscenze del personale, le lingue parlate, le specializzazioni, oltre alle informazioni sul salario.
- *Employee Self Service*: garantisce la possibilità di timbratura di ingressi e uscite ai dipendenti, con gestione di calendari personali e programmazione del tempo.

- *Reports module*: utile per la generazione di un vasto numero di reports secondo diversi criteri.
- *Time and attendance module*: per automatizzare il tracciamento delle presenze del personale e organizzare meglio il flusso lavorativo, riducendo errori e perdite di tempo.



The screenshot shows the OrangeHRM interface. At the top, there is a navigation bar with various menu items like Home, Admin, PM, Leave, Time, Reports, etc. Below the navigation bar, the main content area displays an "Attendance Report: Darja Jevtuchova". The report includes a table with the following data:

Employee Name	Date	Duration
Darja Jevtuchova	2011-08-09	00:00
Darja Jevtuchova	2011-08-09	00:00
Darja Jevtuchova	2011-08-10	00:00
Darja Jevtuchova	2011-08-11	00:00
Darja Jevtuchova	2011-08-12	00:00
Darja Jevtuchova	2011-08-13	00:00
Darja Jevtuchova	2011-08-14	00:00
Darja Jevtuchova	2011-08-15	00:00
Darja Jevtuchova	2011-08-16	00:00
Darja Jevtuchova	2011-08-17	00:00
Darja Jevtuchova	2011-08-18	00:00
Darja Jevtuchova	2011-08-19	00:00
Darja Jevtuchova	2011-08-20	00:00
Darja Jevtuchova	2011-08-21	00:00
Darja Jevtuchova	2011-08-22	00:00
Darja Jevtuchova	2011-08-23	00:00
Darja Jevtuchova	2011-08-24	00:00

Figura 1.1: Vista dell'interfaccia di OrangeHRM

SmartSuite Business Edition - Bioenable Technologies

Web application sviluppata in PHP, installabile sia in una Intranet aziendale sia in server web.

Le timbrature vengono memorizzate in DBMS di tipo MySQL, con possibilità di importarle ed esportarle, inoltre questa versione del software supporta diversi strumenti hardware di acquisizione, offerti dall'azienda produttrice stessa.

In caso di necessità è possibile acquistare degli interventi di personalizzazione del software a seconda delle esigenze e il supporto è garantito in diversi modi, come forum, email, chat e telefono.

Si possono definire più profili utente e relativi privilegi, con la possibilità di controllare le presenze e l'orario di lavoro tramite grafici e schemi esportabili.

La gestione degli stipendi viene semplificata notevolmente con calcoli automatici sulle informazioni salariali di ogni dipendente e sulle ore di lavoro, inoltre vengono pianificati i compiti lavorativi assegnati.

È possibile poi monitorare la situazione aziendale attraverso la generazione di più di 50 reports configurabili e personalizzabili.

Tulip - Bloomtech s.n.c.

Software prodotto da una ditta di Conegliano (TV) adottato da più di 250 aziende di ogni settore e dimensione, per far fronte alla gestione non solo delle timbrature ma anche delle commesse, è disponibile sia come software installabile sia come applicativo web accessibile da browser (con l'acquisto del relativo modulo).

magicsoftware [Main Menu](#)

Magic Software Pvt Ltd
 Magic Software Pvt Ltd
 9th Floor, Tower C, Tech Soutward
 Plot No 6, Sector 127
 Noida-201 301 (Near Delhi) INDIA
 Tel: +91 120 3054 300
 Fax: +91 120 3054 399
 eMail: mail@magicv.com, info@www.magicv.com

Logged on as user@bioenabletech.com [Log out](#) [Change password](#) [Advanced search](#)

Home - Company - Employees - Reports - Leaves - Attendance - Attendance Logs - Calendar - Help

Search for Any field Contains Search Show all Details found: 1121 Page 1 of 57 Records Per Page: 20

[Add new](#) [Delete](#) [Edit selected](#)

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Punch Date	Punch Time	Punch Type	Status	Device ID	Loc Type(EN/OUT)	Shift	Location	Remark
InLine Edit	View	<input type="checkbox"/>	8/4/2011	3:15:15 PM	Manual	Valid	0			Pune	
InLine Edit	View	<input type="checkbox"/>	8/2/2011	9:15:00 AM	Manual	Valid	0	IN/START		Pune	
InLine Edit	View	<input type="checkbox"/>	8/1/2011	9:28:00 AM		Valid	1	0	0	0	
InLine Edit	View	<input type="checkbox"/>	7/30/2011	6:22:00 PM		Valid	1	0	0	0	
InLine Edit	View	<input type="checkbox"/>	7/29/2011	9:28:00 AM		Valid	1	0	0	0	
InLine Edit	View	<input type="checkbox"/>	7/29/2011	6:45:00 PM		Valid	1	0	0	0	
InLine Edit	View	<input type="checkbox"/>	7/28/2011	9:31:00 AM		Valid	1	0	0	0	
InLine Edit	View	<input type="checkbox"/>	7/28/2011	9:40:00 AM	FingerPrint	Valid	1			0	**
InLine Edit	View	<input type="checkbox"/>	7/28/2011	1:43:00 PM	FingerPrint	Valid	1			0	**
InLine Edit	View	<input type="checkbox"/>	7/28/2011	2:01:00 PM	FingerPrint	Valid	1			0	**
InLine Edit	View	<input type="checkbox"/>	7/28/2011	7:06:00 PM		Valid	1	0	0	0	
InLine Edit	View	<input type="checkbox"/>	7/28/2011	7:11:00 PM	Password	Valid	1			0	**
InLine Edit	View	<input type="checkbox"/>	7/27/2011	9:40:00 AM	FingerPrint	Valid	1			0	**
InLine Edit	View	<input type="checkbox"/>	7/27/2011	9:42:00 AM		Valid	1	0	0	0	
InLine Edit	View	<input type="checkbox"/>	7/27/2011	1:43:00 PM	FingerPrint	Valid	1			0	**
InLine Edit	View	<input type="checkbox"/>	7/27/2011	2:01:00 PM	FingerPrint	Valid	1			0	**
InLine Edit	View	<input type="checkbox"/>	7/27/2011	7:11:00 PM	Password	Valid	1			0	**
InLine Edit	View	<input type="checkbox"/>	7/27/2011	7:29:00 PM		Valid	1	0	0	0	
InLine Edit	View	<input type="checkbox"/>	7/26/2011	9:40:00 AM	FingerPrint	Valid	1			0	**
InLine Edit	View	<input type="checkbox"/>	7/26/2011	1:43:00 PM	FingerPrint	Valid	1			0	**

[1 2 3 4 5 6 7 8 9 10] Next : Last

BioEnable SmartSuite version 2.01 (updated Dec 2010) © 2010 BioEnable. All rights reserved.

Figura 1.2: Vista dell'interfaccia di SmartSuite

Tulip è compatibile con i database Oracle, Microsoft Access o Microsoft SQL Server, le timbrature possono essere rilevate con numerosi sistemi, come quelli offerti dalla ditta *DBL* di Udine: codice a barre, banda magnetica, transponder in radiofrequenza, impronta biometrica, foto istantanea da fotocamera e riconoscimento del volto.

Tutte le funzioni di Tulip sono personalizzabili, come le informazioni del personale, gli orari di lavoro, le regole per gestire le pause e le varie tolleranze di orario, le ferie, i permessi e molto altro.

Si possono impostare dei totalizzatori, per il calcolo di orari o di buste paga, cartellini personalizzati, il tutto per velocizzare le mansioni della segreteria.

Vengono inoltre gestite le missioni e le trasferte del personale, con la possibilità di inserimento di giustificativi e ogni totalizzatore o cartellino può essere esportato in diversi formati.

Non tutte le funzionalità sopracitate però sono integrate nativamente nell'applicativo, bensì sono acquistabili come moduli da integrare, come “Rilevazione tempi commesse” per pianificare le commesse partendo dalle sole timbrature, “Controllo accessi” utile per impostare le regole e gli orari d’accesso del personale, “Risorse umane” per controllare e gestire tutto il personale e infine “TulipWeb” per far accedere i dipendenti autorizzati alle azioni di timbratura e controllo direttamente tramite browser web da qualsiasi computer connesso alla rete aziendale.

1.4.2 Soluzioni FOSS

Acronimo di “Free and Open Source Software”, i prodotti con questo tipo di licenze integrano i concetti di software libero e di Open-Source, ovvero le applicazioni possono essere utilizzate gratuitamente e il loro codice sorgente è disponibile a chiunque per modifiche o personalizzazioni.

Tuttavia per intervenire nel software è necessario del personale all’interno dell’azienda con conoscenze e capacità adatte, altrimenti bisogna affidare il lavoro ad una ditta esterna, con i costi annessi.

Molto spesso non è presente una documentazione adeguata, il prodotto evolve grazie ad una *community* di utilizzatori che eventualmente può fornire aiuto tramite forum, mentre l’assistenza ufficiale è a pagamento.

OrangeHRM - OrangeHRM inc.

Soluzione Open-Source leader internazionale per la gestione del personale per imprese di piccole o medie dimensioni, il programma è scaricabile liberamente mentre il supporto, i corsi di formazione, le personalizzazioni e i moduli aggiuntivi (che sono gli stessi della versione Live) sono a pagamento.

Si tratta della soluzione liberamente scaricabile dell’omonimo software già analizzato in precedenza, la community garantisce un prodotto stabile e continuamente aggiornato, però nativamente è privato di molte funzioni utili rispetto

tulipweb

HOME TIMBRATURE CARTELLINO

Riepilogo timbrature

Dal Al

Giorno	Timbrature	Giustificativi
sab 30/07/11	E11:52 U16:52	
dom 31/07/11		
lun 01/08/11		
mar 02/08/11		
mer 03/08/11		
gio 04/08/11		
ven 05/08/11		
sab 06/08/11		
dom 07/08/11		
lun 08/08/11		
mar 09/08/11		
mer 10/08/11		
gio 11/08/11		
ven 12/08/11		
sab 13/08/11		
dom 14/08/11		

PDF timbrature

Utente : Bianchi Mario Connesso come : demo - [Disconnetti](#)

Figura 1.3: Vista dell'interfaccia del modulo TulipWeb

alla versione a pagamento, come l'esportazione di resoconti in formati CSV o PDF.

SmartSuite Community Edition - Bioenable Technologies

La versione FOSS di SmartSuite viene privata della compatibilità con gli strumenti di marcatura di tipo biometrico o con carte di prossimità e del supporto esteso, lasciando comunque la possibilità di ricevere aiuto dalla community tramite il forum.

Per quanto concerne il resto le funzioni sono le medesime, in particolar modo è possibile eseguire la profilazione degli utenti e la generazione di totalizzatori esportabili.

1.4.3 Conclusioni

Il mercato è florido di applicativi di gestione del personale, ma non essendo soluzioni create o personalizzate ad-hoc implementano un surplus di funzionalità non richieste, che ostacolano pertanto un eventuale utilizzo quotidiano degli stessi, specie da persone meno votate all'utilizzo della tecnologia.

Inoltre nessun applicativo offre un'interfaccia per monitor touch-screen per la timbratura, in questo senso bisognerebbe acquistare interventi esterni di personalizzazione del software, o strumenti terzi di raccolta delle marcature.

Viste le esigenze e le dimensioni medio/piccole della società erogatrice di servizi che ha commissionato l'applicativo, si decide di scartare le soluzioni a pagamento, a causa dei costi di licenza e di personalizzazione.

Infatti per adempiere alle richieste del committente è necessario l'acquisto di diversi moduli, in quanto molte funzionalità non sono presenti nell'applicazione di base, facendo salire proporzionalmente il prezzo finale.

Le soluzioni Open Source sono interessanti, ma anche loro dotate di molte features non richieste dai requisiti, l'esportazione di files non sempre è presente, il che rende necessaria la spesa sia di moduli aggiuntivi che di personalizzazioni.

Viene deciso quindi di proseguire con lo sviluppo di un applicativo ad-hoc, per meglio rispondere alle esigenze e alle funzionalità richieste, senza avere funzioni non necessarie che probabilmente non verranno mai utilizzate, in questo modo il prodotto finale avrà un utilizzo più semplice e un'interfaccia più chiara.

Prodotto / Funzione	OrangeHRM Live	SmartSuite Business ed.	Tulip	OrangeHRM	SmartSuite Community ed.
DBMS	MySQL	MySQL	Oracle, MS Access, MS SQLServer	MySQL	MySQL
Web-application	✓	✓	Modulo aggiuntivo	✓	✓
FOSS	✗	✗	✗	✓	✓
Sezione Amministrazione	✓	✓	✓	✓	✓
Area Personale	✓	✓	✓	✓	✓
Interfaccia di front-end	✗	✗	✗	✗	✗
Profilazione utenti con ruoli	✓	✓	✓	✓	✓
Personalizzazione ruoli	✓	✓	✓	✓	✓
Personalizzazione tipi di timbratura	Note di ingresso e uscita	✓	Possibilità di aggiungere giustificativi	Note di ingresso e uscita	✓
Generazione cartellino	✓	✓	✓	✓	✓
Esportazione CSV	✓	✓	✓	✗	✓

Figura 1.4: Confronto degli applicativi analizzati sulle funzionalità richieste dai requisiti

Capitolo 2

Realizzazione

2.1 Progettazione di dettaglio con approfondimento dei requisiti

2.1.1 Casi d'uso

Come illustrato nella figura 2.1, il sistema innanzitutto deve garantire un'interfaccia atta alla raccolta delle timbrature senza restrizioni di utenza, in secondo luogo deve essere raggiungibile un'area ristretta, tramite login, nella quale si distinguono le diverse funzionalità disponibili per l'utente semplice o per l'amministratore.

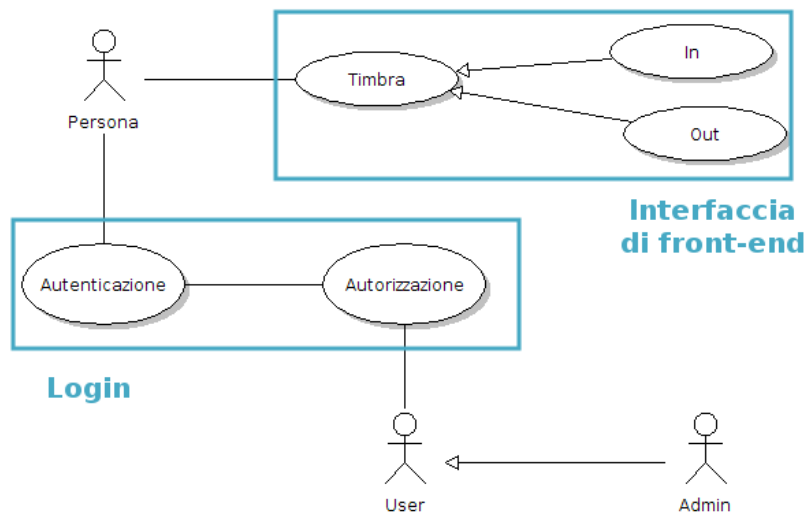


Figura 2.1: Casi d'uso della web-application

Interfaccia di Front-end

Ovvero una pagina web adatta ad essere visualizzata in monitor touch-screen fissi con l'unica funzione di raccogliere le timbrature degli utenti in entrata e in uscita.

Sarà composta da un tastierino numerico di dimensioni ragionevoli, un display dove visualizzare il codice che si sta inserendo, un tasto per eliminare l'ultimo carattere e i tasti per confermare le timbrature, inoltre sarà presente un orologio digitale come riscontro dell'ora attuale.

Al momento della timbratura il software dovrà controllare nel database degli utenti registrati la presenza del codice inserito, in caso affermativo verrà stampato a schermo un messaggio di avvenuta marcatura per l'utente, in caso contrario una notifica di errore.

Dal momento che questa pagina web dovrà rimanere fissa in un chiosco, non dovranno esserci possibilità di reindirizzamento o di apertura di nuove finestre, al contrario bisogna garantire un'alta velocità di timbratura.

Login

L'accesso per le aree ristrette avverrà tramite username e password, scelte in base alle preferenze dell'utente, con il vincolo dell'username di tipo numerico per garantirne la digitazione attraverso l'interfaccia touch-screen di front-end.

Una volta effettuato e confermato l'accesso seguirà la profilazione dell'utente, ovvero l'assegnazione dei privilegi garantiti per la categoria di appartenenza del soggetto.

I due ruoli principali sono "Utente" e "Amministratore", nel primo si possono individuare i dipendenti semplici dell'azienda, la seconda categoria invece è più adatta alla dirigenza con allegate funzioni di controllo, oppure alla segreteria.

Area personale

Sarà accessibile dagli utenti semplici che hanno effettuato il login, in questa pagina potranno richiamare la funzione per la generazione del cartellino personale.

In questo modo verrà offerta la possibilità di controllare le proprie presenze in un periodo temporale a scelta e con un pulsante apposito sarà possibile esportare i dati in un file CSV, ovvero in foglio di calcolo elettronico, utilizzabile con programmi come Microsoft Office Excel o LibreOffice Calc.

Infine l'utente potrà modificare le proprie credenziali di accesso, cioè username e password, e terminata la sessione si potrà disconnettere tramite il pulsante di logout.

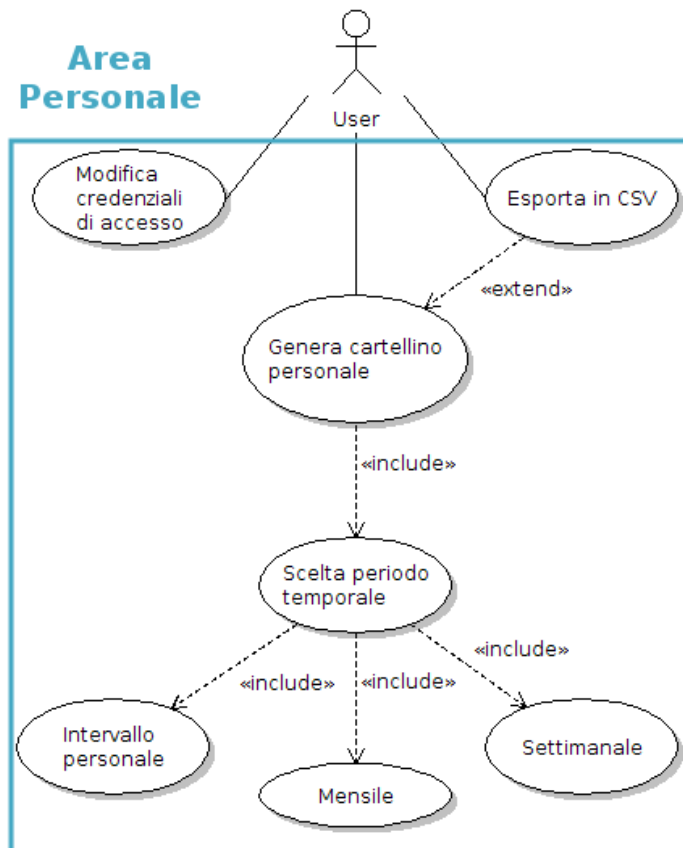


Figura 2.2: Casi d'uso per gli utenti

Sezione per gli amministratori

Se ad effettuare l'accesso sarà una persona con i privilegi di amministratore, saranno disponibili nuove funzionalità oltre a quelle già presenti per utenti semplici, infatti gli amministratori necessitano di avere il controllo completo dei dati presenti per adempiere alle proprie mansioni, come ad esempio il calcolo delle buste paga nel caso della segreteria.

Oltre al cartellino personale verrà offerta la possibilità di richiedere e di esportare in CSV anche le timbrature di qualsiasi utente presente nel database, quindi prima di scegliere il periodo temporale di riferimento si dovrà selezionare da una lista il dipendente da controllare.

Inoltre tramite un'apposita interfaccia sarà possibile effettuare operazioni CRUD su ogni informazione presente nel database, come ad esempio modificare un utente, ricercare una determinata timbratura, oppure aggiungere nuovi profili di privilegi.

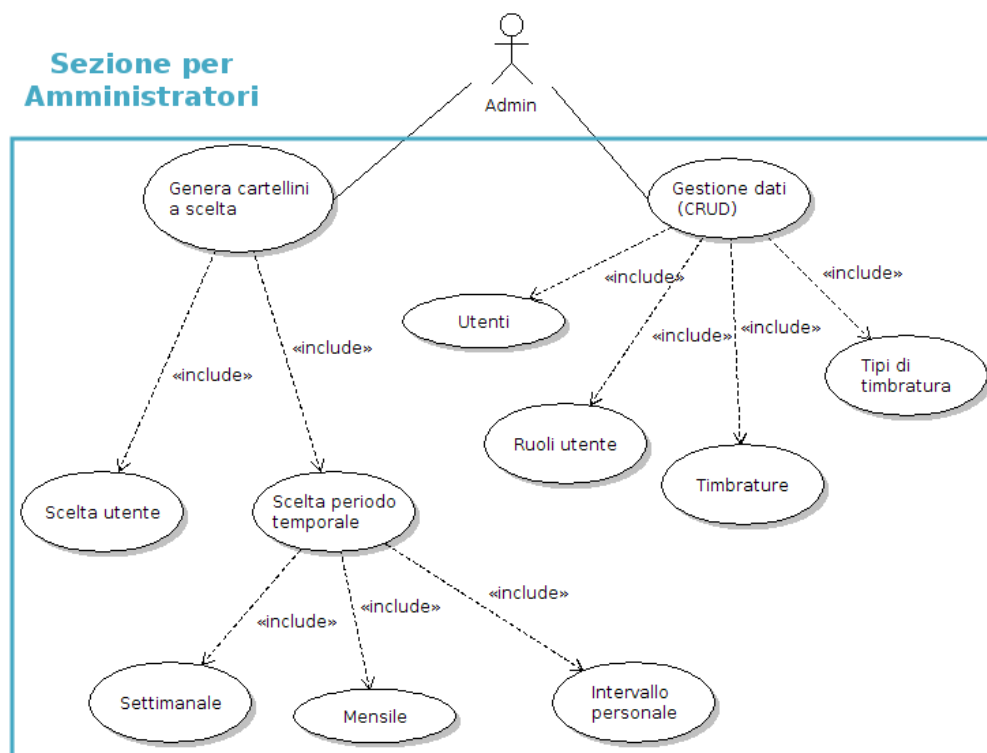


Figura 2.3: Casi d'uso per gli amministratori

2.1.2 Modello dati

Come già discusso, l'applicativo dovrà interfacciarsi con un database creato appositamente per archiviare i dati utili.

Analizzando i requisiti sui dati vengono identificate 4 entità da gestire, che verranno chiamate: *event*, *user*, *type* e *role*.

Sullo schema *event*, ogni evento, ovvero ogni timbratura, verrà gestito singolarmente, facendolo corrispondere ad una sola entry.

L'alternativa sarebbe stata la creazione di righe formate da coppie ingresso/uscita, ma è sicuramente più funzionale e scalabile mantenere i dati atomici, che hanno più possibilità di utilizzo in aggiunte funzionali future rispetto a dati "composti", più difficili da manipolare.

Gli attributi di *event* sono "time", di tipo DateTime, utile per salvare il timestamp relativo all'istante di timbratura e i campi "user_id" e "type_id", cioè le chiavi esterne che puntano all'utente e alla tipologia di marcatura, come suggeriscono le cardinalità (1,1) nello schema ER di figura 2.4.

Nella tabella *user* verrà gestito tutto il personale, dal dipendente normale all'amministratore, con i campi "name" e "surname" per il nome e cognome del soggetto, "username" per il codice numerico personale utilizzato in fase di login e marcatura, "password" e "salt", dove risiedono rispettivamente la password per l'accesso criptata e la stringa per decrittalarla e infine "role_id" per la chiave esterna del ruolo assegnato al soggetto.

Le rimanenti entità *role* e *type* contengono la lista rispettivamente dei profili utente e delle tipologie di timbratura assegnabili, ognuna con il campo "name" dove inserire il relativo nome.

2.2 Strumenti utilizzati

Nella scelta dei prodotti impiegati per lo sviluppo della web application si è tenuto conto di caratteristiche come:

- **Tipologia FOSS:** programmi Open Source il cui utilizzo è possibile senza costi di licenza e che sono supportati da una numerosa community.
- **Affidabilità:** software aggiornati e diffusi, il cui utilizzo è ben rodato.
- **Supporto informativo:** viene garantita la presenza di documentazione adeguata.

La web application verrà sviluppata su piattaforma LAMP, acronimo di *Linux-Apache-MySQL-PHP*, per indicare rispettivamente sistema operativo, web-server, DBMS e linguaggio di scripting utilizzati, inoltre verrà sfruttato l'ausilio di un framework: una serie di librerie con lo scopo di fornire supporto e velocizzare l'attività di programmazione.

<i>Entità</i>	<i>Descrizione</i>	<i>Attributi</i>	<i>Identificatore</i>
Event	Timbrature effettuate da ogni utente registrato	id, time, user_id, type_id	id (contatore)
User	Utenti abilitati alla timbratura	id, username, name, surname, role_id, password, salt	id (contatore)
Role	Profili che possono essere assegnati agli utenti	id, name	id (contatore)
Type	Tipi di timbrature effettuabili	id, name	id (contatore)

Tabella 2.1: Dizionario delle entità

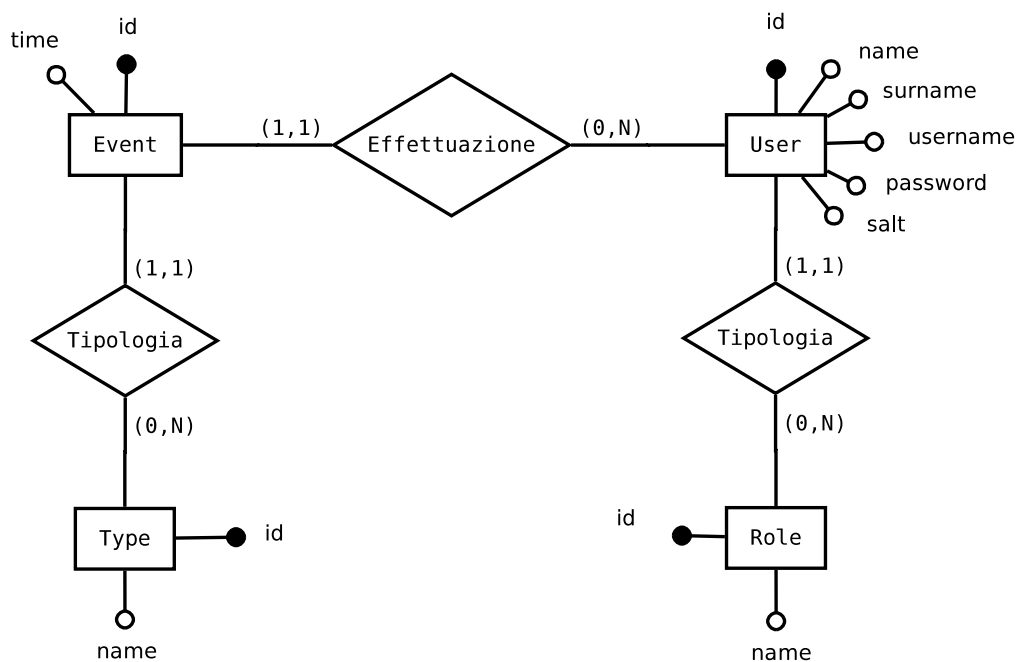
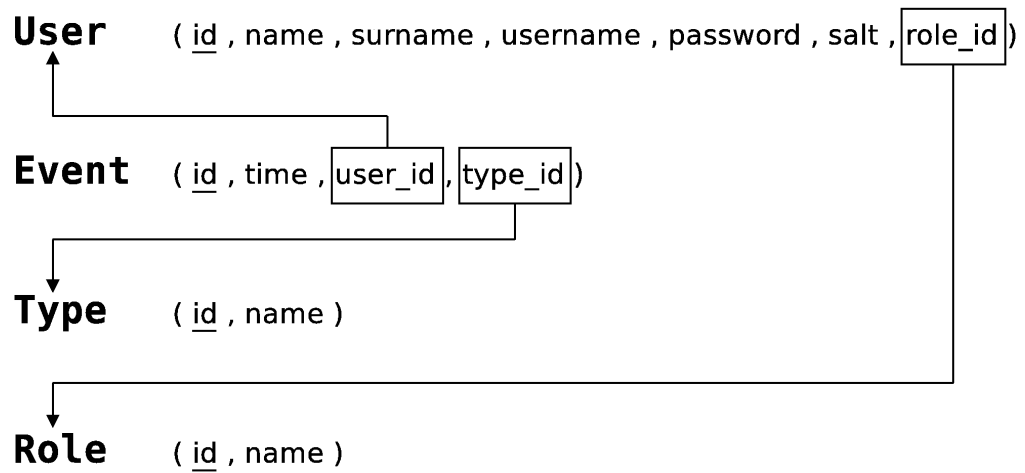


Figura 2.4: Schema ER



Gli attributi:

- "Username" di User
- "name" di Type
- "name" di Role

devono essere UNIQUE

Figura 2.5: Schema Logico

L'utilizzo di framework è indispensabile soprattutto per progetti di una certa dimensione, infatti il programmatore evita la riscrittura di codice per funzioni usate frequentemente, concentrandosi sulla logica di base dell'applicazione, senza perdite di tempo.

Non è ritenuto necessario l'acquisto di costose licenze di DBMS proprietari come Oracle, verrà utilizzato MySQL, uno strumento Open Source affidabile, performante e semplice da utilizzare che al giorno d'oggi accompagna la maggior parte delle web application sviluppate.

2.2.1 PHP 5.3

“PHP: Hypertext Preprocessor” è il linguaggio di scripting Open Source server-side per eccellenza ideato dal danese Rasmus Lerdorf nel 1994.

È lo strumento principale per lo sviluppo di web application o pagine web dinamiche, infatti si integra perfettamente con il codice HTML, generandolo dinamicamente dal lato server e inviandolo al client.

La dinamicità dell'output si basa su condizioni impostate dal programmatore, ad esempio su informazioni del database o sull'input dell'utente, producendo codice HTML influenzato dall'elaborazione di questi dati.

I suoi utilizzi però sono molteplici, ad esempio è possibile creare scripting di righe di comando da usare senza alcun server o browser, oppure per creare applicazioni desktop con interfaccia grafica.

Features che hanno reso PHP lo strumento di lavoro ideale e hanno contribuito alla sua diffusione sono la compatibilità con la maggior parte dei DBMS esistenti, dei sistemi operativi, dei web-server, la documentazione ben fatta e sempre aggiornata e le numerose API esistenti.

Dal lato pratico l'utilizzo di PHP è semplice, la sintassi infatti deriva dal linguaggio C, come JAVA e C++, inoltre è possibile utilizzare un approccio alla programmazione sia di tipo OOP, ovvero orientato agli oggetti, sia di tipo procedurale, o volendo anche un misto di questi due.

2.2.2 Symfony 2.0

Esistono molti framework per PHP, come Zend, Symfony, CakePHP, Code Igniter e Yii, ognuno con caratteristiche diverse, per questo progetto viene scelto il nuovo Symfony 2.0[1], sviluppato da Fabien Potencier e dotato di licenza MIT: si tratta quindi di un prodotto Open Source, con un'ottima documentazione e una vasta community pronta a fornire supporto.

Al contrario degli altri, Symfony si può definire operante ad un livello di astrazione più elevato, orientato più alla semplificazione delle operazioni CRUD (create, read, update, delete) grazie agli strumenti integrati, garantendo un veloce caricamento delle pagine.

Lo scheletro operativo si basa sull'architettura MVC “*Model-View-Controller*”: il *controller* raccoglie la richiesta dell'utente e, a seconda del suo scopo, si in-

terfaccia ai dati tramite le funzioni fornite dal *model*, molto spesso al fine di accedere al database, infine la risposta viene visualizzata dall'utente attraverso *view*.

MVC quindi separa nettamente il template grafico, l'interfaccia, dalla logica operativa di fondo e dai metodi di accesso ai dati, assicurandone l'indipendenza reciproca, fondamentale per quanto concerne modifiche o aggiunte future.

Questa filosofia è perfettamente integrata in Symfony, le cui parti fondamentali sono:

- **Bundle:** Directory che rappresenta una o più funzionalità ben precise, contenente codice, impostazioni e stili necessari. Inoltre è condivisibile e utilizzabile in qualsiasi progetto si voglia, garantendo la scalabilità richiesta dai requisiti.
- **Routing:** File scritto in XML, PHP o YAML contenente le istruzioni riguardanti l'instradamento di una richiesta dell'utente, ovvero quale controller eseguire se l'URL corrisponde ad una certa label.
- **Controller:** Al suo interno vengono definite delle "Actions", ovvero funzioni PHP atte a gestire delle *Request* dell'utente, in base al routing, e che ritornano delle *Response*, spesso rappresentate da render di nuove pagine con i risultati.
- **Template:** File dove viene definita la struttura di una pagina, vengono renderizzati tramite il controller e contengono un misto di codice HTML e TWIG, un template engine per PHP sviluppato dallo stesso inventore di Symfony, con lo scopo di diminuire la verbosità del codice utile al rendering, per renderlo più conciso e semplice.

Ricapitolando, ecco la logica Symfony: Un utente richiede un URL, la richiesta viene raccolta dal front controller di Symfony che tramite il kernel del framework verifica se è presente quel percorso nel file di routing, dove è anche indicato il controller da invocare.

Il controller eseguito risiederà all'interno di un bundle dell'applicazione ed effettuerà le operazioni per le quali è stato creato, terminando con la renderizzazione del template di una nuova pagina, eventualmente contenente i risultati o altro.

Questo funzionamento applicato al software in via di sviluppo è illustrato nell'immagine 2.6.

Symfony2 integra una console (accessibile attraverso il terminale di linux) molto utile per automatizzare ed evitare passaggi ricorrenti, come la creazione di bundle, database e schemi, e impartire altri comandi come la pulizia della cache, il caricamento degli assets (CSS, immagini, ...), il caricamento di data fixtures (per popolare il database) e altro.

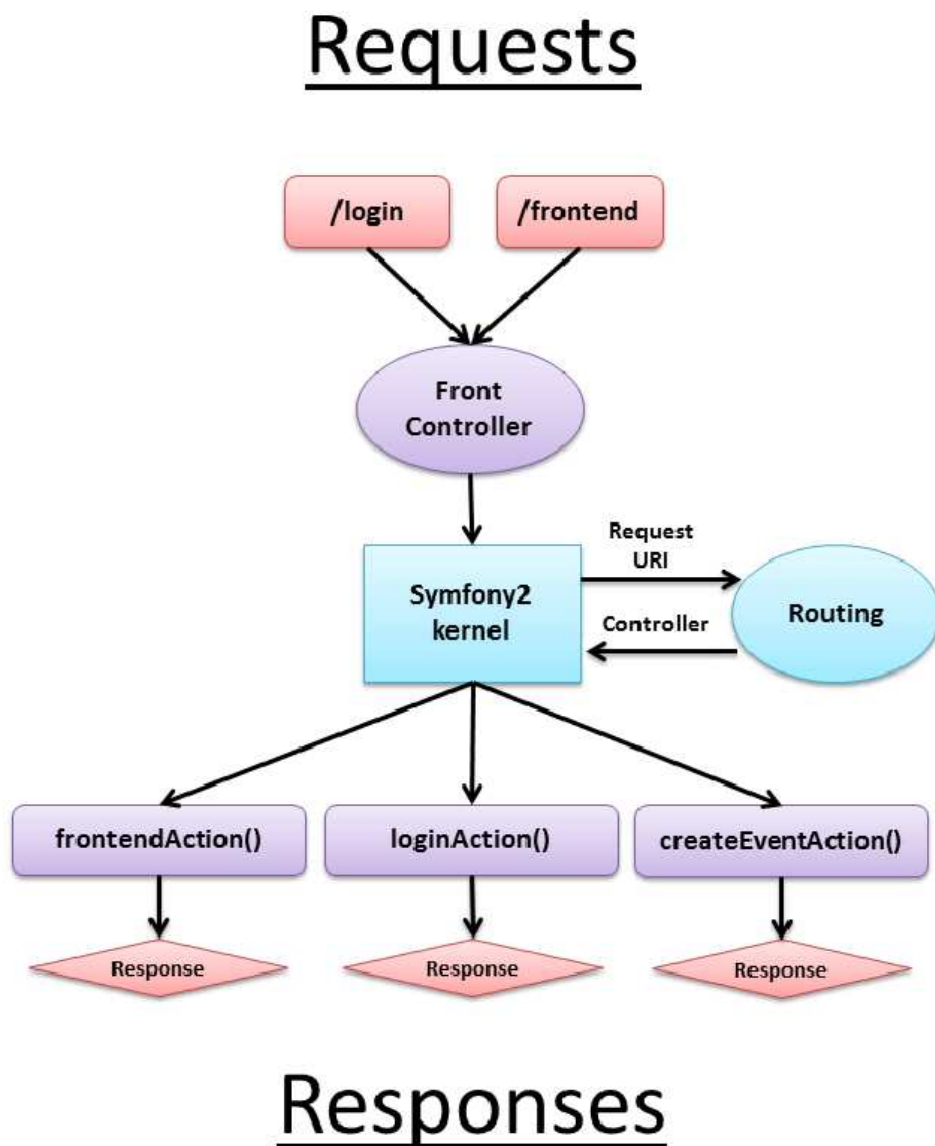


Figura 2.6: Pattern di funzionamento del framework per rispondere alle richieste dell'utente

2.2.3 Doctrine

È una libreria integrata nativamente in Symfony con funzione di Object Relational Mapper (ORM)[2] per PHP, ovvero vengono creati oggetti di “collegamento” al database, con possibilità di far persistere oggetti in modo trasparente, direttamente dal codice PHP usando i paradigmi della programmazione OOP.

Inoltre integra la funzione di Database Abstraction Layer (DBAL), ovvero mette in comunicazione un’applicazione qualsiasi ai database più diffusi, come MySQL, Oracle, PostgreSQL, ecc...

Per interrogare il database è possibile utilizzare sia il linguaggio SQL classico, sia il Doctrine Query Language (DQL), un linguaggio orientato agli oggetti che semplifica e snellisce il codice senza andare ad incidere sulla flessibilità.

2.2.4 AdminBundle

Bundle sviluppato da Thomas Rabaix, successore del famoso Admin Generator dalle simili funzioni ma compatibile con la prima versione di Symfony.

Grazie alle funzionalità di AdminBundle è possibile sviluppare con semplicità e velocità un’interfaccia di back-end, adatta in special modo agli amministratori, viene infatti offerta la possibilità di eseguire operazioni CRUD sui dati.

Viene creata nell’applicazione una sezione chiamata “Dashboard”, nella quale vengono visualizzati gli schemi presenti nel database, con possibilità di aggiungere, modificare, eliminare e ricercare dati, oppure ordinarli a seconda dell’attributo.

2.2.5 MySQL 5.5

Si tratta del più diffuso “Relational database management system” (RDBMS) Open source utilizzato soprattutto in piattaforme LAMP (Linux-Apache-MySQL-PHP), rilasciato su licenza GNU GPL, scritto in C e C++ e ceduto da Sun Microsystem a Oracle in via definitiva il 27 Gennaio 2010.

Le caratteristiche che hanno decretato il suo successo sono la compatibilità con praticamente tutti i sistemi operativi, la facilità di utilizzo e la natura FOSS dell’applicativo, rendendo possibile risparmi notevoli di tempo e denaro non solo ai singoli sviluppatori, ma anche a realtà di alto livello, come Wikipedia, Google, Youtube, Yahoo! e tantissime altre, che devono gestire ingenti quantità di dati.

Spesso viene utilizzato in coppia con *phpMyAdmin*, un’applicazione web con funzione di MySQL manager, ovvero atta all’amministrazione di database, tramite interfaccia grafica, con possibilità di eseguire query e modifiche direttamente agli schemi.

PhpMyAdmin verrà utilizzato anche nel presente progetto, soprattutto con scopi di controllo e verifica della corretta impostazione della base di dati.

È necessario citare anche la possibilità di scegliere la tipologia delle tabelle, ovvero lo storage engine, i più famosi sono senza dubbio InnoDB e MyISAM,

ma sono disponibili tantissimi altri, con pro e contro diversi a seconda del tipo di dati contenuti.

2.2.6 jQuery

Libreria di funzioni JavaScript sviluppata da John Resig nel 2006 con l'intento di rendere il codice più sintetico, senza il bisogno di toccare gli oggetti nativi di JavaScript, aumentando la compatibilità con altre librerie.

jQuery offre una vasta gamma di funzionalità, come la manipolazione dei CSS e degli elementi HTML, gli effetti grafici e i comodi metodi per effettuare chiamate AJAX cross-browser.

Grazie alla sua compatibilità e al suo possibile utilizzo in tutti i progetti, alla sua gestione delle estensioni e alla sintassi sintetica ed efficiente, questa libreria si è diffusa velocemente e si è creata attorno al team di sviluppo una numerosa community che mette a disposizione plugin e supporto.

Di particolare interesse sono le features messe a disposizione da jQuery UI, una libreria integrata nel framework e adatta per la realizzazione di interfacce utente funzionali e dall'estetica piacevole.

Sono infatti presenti diversi effetti grafici aggiuntivi, dei comodi widget come calendario, finestre modali e navigazione a schede e la gestione di alcune interazioni complesse, come l'ordinamento e il drag & drop di oggetti, il tutto integrabile in maniera semplice e con poche righe di codice.

2.2.7 GIT

Software di controllo della versione, creato da Linus Torvalds e di tipo Open Source con licenza GPL, pensato per agevolare lo sviluppo in un ambiente distribuito, dove ognuno necessita di essere a conoscenza delle ultime modifiche o di aggiornarsi sullo stato raggiunto in un certo progetto.

L'utilità principale è la storicizzazione dei cambiamenti che avvengono in via di sviluppo di ogni documento, dal codice sorgente di un applicativo, al kernel di un sistema operativo.

Esiste poi la possibilità di sdoppiare una linea di sviluppo, meccanismo chiamato "Branching", e di proseguire con lavorazioni separate dei due rami, con l'eventualità di unirle in futuro tramite "Merging".

Ciò che più lo differenzia dagli altri strumenti di controllo revisione è il fatto che cattura istantanee di interi alberi di directory di file invece che di singoli file, rendendolo più adatto a progetti di una certa dimensione.

Lo sviluppo dell'applicativo sarà affiancato dall'utilizzo di GIT per storicizzare i cambiamenti, in caso di necessità verranno creati nuovi branch, nei quali testare eventuali alternative funzionali o grafiche, il tutto con la possibilità di effettuare rollback a stati precedenti, fino a realizzare il prototipo finale in linea con le aspettative.

2.3 Sviluppo del software

Lo sviluppo inizierà con l'installazione e la configurazione del software necessario e verrà creato il database seguendo il modello dati visto nel paragrafo 2.1.2.

Si procederà poi con la creazione dell'interfaccia di front-end per le postazioni touch-screen, con il relativo layout e la funzione di creazione delle timbrature corredata con i controlli necessari.

Verrà poi sviluppata la sezione di login per l'area personale e il form per l'accesso, integrando la sicurezza necessaria per la separazione dei diversi profili di privilegi.

Infine verranno inserite tutte le funzioni richieste dai requisiti, come la generazione del cartellino e l'esportazione di quest'ultimo in formato CSV, testando di volta in volta i risultati ottenuti.

2.3.1 Preparazione piattaforma LAMP

La distribuzione di Linux scelta per lo sviluppo è Ubuntu 10.04 LTS.

Successivamente si procede prima con l'installazione del web server *Apache2*, del database server *MySQL*, di *PHP5* e delle librerie di questi ultimi per Apache2, dando per assunto che nel sistema sia già installato un *Java Development Kit* (JDK).

Dopo aver apportato le necessarie configurazioni, vengono preparati altri strumenti che verranno utilizzati come *phpMyAdmin* ed *Eclipse PHP Helios*, un ambiente di sviluppo integrato (IDE) consigliato per programmare in PHP.

In seguito viene configurato *Symfony2*, per il suo funzionamento infatti è richiesto qualche componente aggiuntivo e delle impostazioni specifiche, come suggerito dalla pagina di controllo installazione.

L'integrazione di *AdminBundle* al suo interno è facile e veloce e dopo aver verificato il corretto funzionamento di ogni componente si è pronti per lo sviluppo.

2.3.2 Creazione Database

Per prima cosa occorre creare il bundle per il progetto tramite il comando *init:bundle* direttamente da console.

Come già accennato, tramite *Doctrine ORM* è possibile gestire le entry del database come dei veri e propri oggetti, le cui classi entità risiedono nella cartella "Entity" del progetto.

Le classi entità sono delle semplici classi PHP, con variabili che rappresentano gli attributi dello schema e commenti dove vengono indicate le caratteristiche, come il tipo di valore salvato (Integer, DateTime, String...) e altro, il tutto corredata da metodi *getter* e *setter*, utili rispettivamente per ottenere e per impostare il valore dell'attributo corrispondente.

Non è tutto, Symfony può auto-generare queste classi con un semplice comando da console, a partire da file con estensione *.orm.yml*, scritti in YAML, quindi molto meno verbosi.

Altra features che viene sfruttata è la gestione della sicurezza secondo Symfony, configurata nel file *security.yml* nel quale bisogna definire dove risiedono gli utenti con relative password e i ruoli utili per l'accesso (ad esempio in quale schema del database) e gli indirizzi privati, non raggiungibili da chiunque attraverso il browser.

Per automatizzare il controllo dell'accesso tramite framework le classi riguardanti gli utenti e i ruoli devono implementare le interfacce *UserInterface* e *RoleInterface* fornite da *Symfony2*.

Si procede quindi con la creazione dei file *.orm.yml* che rappresentano le entità necessarie, in seguito tramite il comando da console *doctrine:generate:entities* vengono create le classi commentate.

Con i comandi *doctrine:database:create* e *doctrine:schema:create* vengono generati rispettivamente il database, che avrà nome e tipologia definiti nel file di configurazione, e gli schemi a partire dalle classi entità.

In questo modo il database è pronto all'utilizzo senza aver scritto una riga di codice SQL e può già essere popolato, in caso di sviluppi o cambiamenti futuri basta modificare i file YAML e ripetere i comandi da console per applicare le modifiche, risparmiando molto tempo.

2.3.3 Sviluppo interfaccia di Front-end

Per quanto concerne la pagina web fissa utile per le timbrature, viene impostato inizialmente il layout, con i bottoni numerici, il tasto "Cancella", "Entrata" e "Uscita", tutto corredato da fogli di stile relativi, per adattarla meglio alla visualizzazione e alla digitazione su monitor touch screen.

Utilizzando codice JavaScript si aggiunge in testa alla pagina uno script che visualizza un orologio digitale.

Quando viene premuto il pulsante relativo all'entrata o all'uscita, viene effettuata una richiesta POST con parametri il codice digitato e la tipologia del pulsante azionato (IN o OUT).

La richiesta viene gestita da un metodo action del controller, che si occupa di accedere al database e controllare che l'username sia valido (cercandolo nella tabella "user" e campo "username").

In caso di mancata corrispondenza viene visualizzato un avviso di errore nella schermata di front-end, in caso di match invece viene creato un nuovo "event", ovvero una timbratura con i dati passati e impostando il campo *time* al timestamp corrente, successivamente viene stampato un riscontro di avvenuta timbratura con il nome dell'utente.

L'aggiunta di una nuova entry nel database si effettua attraverso una classe di *Doctrine* chiamata *EntityManager* (access point principale alle funzionalità di ORM) tramite la quale si possono ad esempio far persistere nuovi oggetti in

uno schema oppure ottenere degli array di elementi già presenti, i cui attributi sono accessibili tramite i metodi *getter/setter*.

Per mantenere l'interfaccia di front-end pulita, ogni dieci secondi vengono eliminati tutti i messaggi presenti ed eventuali codici pendenti nel display.

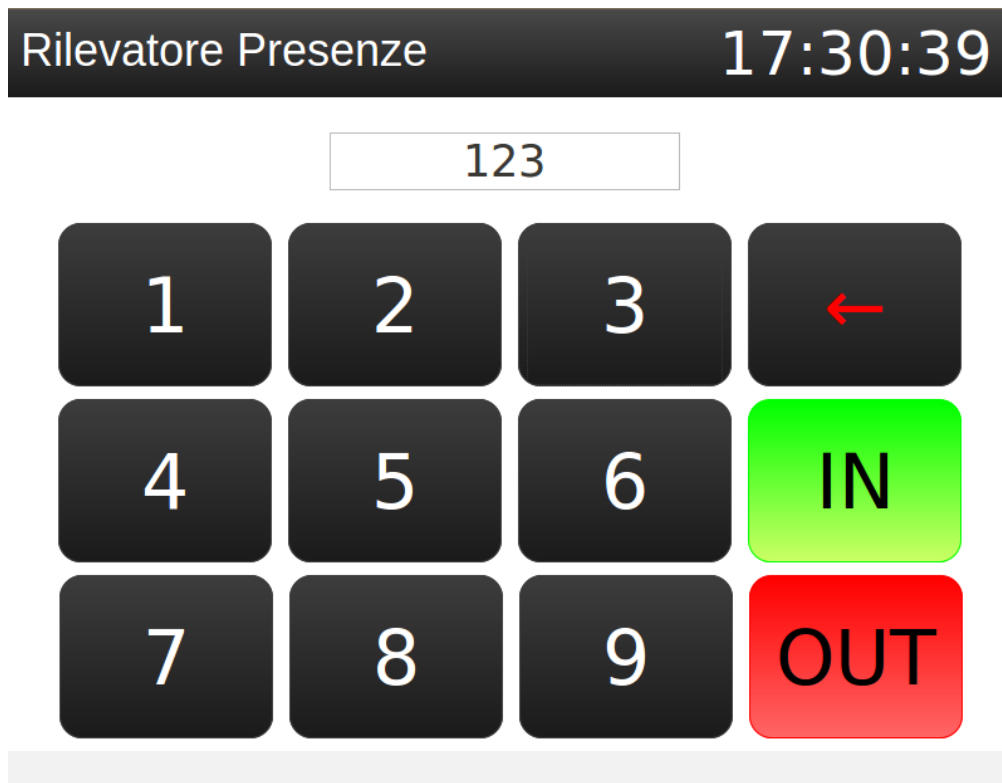


Figura 2.7: Interfaccia di front-end per monitor touch-screen

2.3.4 Sezione di Login

É costituita da un form nel quale inserire username e password, in caso di accesso segue il reindirizzamento all'area personale, qui viene sfruttato il layer di sicurezza di Symfony2: la pagina di login è visualizzabile da tutti, mentre l'area personale solo da un utente loggato e, a seconda del ruolo, diverse funzioni sono invocabili.

Non occorre scrivere altro codice, sono direttamente le funzioni di Symfony a preoccuparsi di controllare la correttezza delle credenziali (dal "distributore" di utenti impostato) e ad assegnare il profilo adatto per l'utente connesso.

Nel file di configurazione della sicurezza viene impostata anche la route specifica per il logout, per comodità dunque vengono posizionati nella barra del titolo due collegamenti, il primo per consentire la disconnessione dell'utente (azione

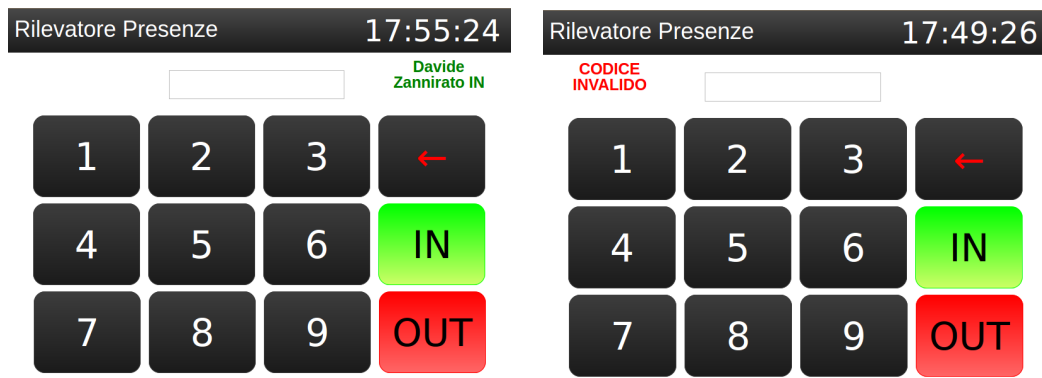
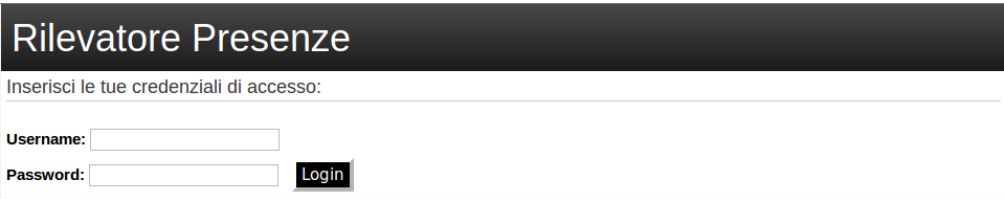


Figura 2.8: Interfaccia di front-end con messaggi di avvenuta timbratura e di errore.

gestita in automatico dal framework), mentre il secondo indirizza l'utente in una nuova pagina, nella quale modificare le proprie credenziali di accesso.

Quest'ultima funzionalità viene gestita con una nuova action del controller, nella quale viene creato il form per l'inserimento dei nuovi valori, che prima di essere salvati vengono sottoposti ad alcuni controlli, ad esempio se sono già utilizzati da qualcun'altro.

Se non ci sono problemi da segnalare vengono aggiornati i valori e il nuovo oggetto è poi fatto persistere tramite i metodi opportuni per rendere effettive le modifiche, notificando a schermo un messaggio di conferma.



Rilevatore Presenze

Inserisci le tue credenziali di accesso:

Username:

Password:

Figura 2.9: Sezione di login per l'autenticazione e l'autorizzazione

2.3.5 Integrazione della dashboard di AdminBundle

Per iniziare viene aggiunto un link all'area personale, visualizzabile solo dagli amministratori, che indirizza alla dashboard, ovvero l'interfaccia che consente l'interazione con le informazioni del database.

AdminBundle[3] per garantire le funzioni CRUD tramite la dashboard necessita di alcuni componenti aggiuntivi per ogni schema presente:

- Un controller che estende la classe “CRUDController”, nel quale si possono ridefinire le azioni di AdminBundle eseguibili sullo schema in considerazione, come la creazione, modifica o eliminazione.
Nella presente applicazione ad esempio, per quanto concerne la tabella user, sono state riprogettate le azioni di “edit” e “create”, per garantire la corretta codifica della password utilizzando il salt dell'utente, impostato in fase di creazione.
- Una classe che estende “Admin”, nella quale sono presenti 3 variabili protected: *\$list*, nella quale vanno definiti gli attributi dello schema da visualizzare nella tabella della dashboard, *\$form*, per indicare quali possono essere modificati e *\$filter* per impostare i campi che possono essere sfruttati per effettuare ricerche.
- Un nuovo servizio nel *Dependency Injection Container* (DIC), ovvero nel file *config.yml* di Symfony, i servizi non sono altro che oggetti PHP che

eseguono un task globale, creati per far fronte a specifiche funzionalità e caricati al momento dell'inizializzazione del framework, in questo caso servono per garantire il corretto caricamento degli schemi nella dashboard.

In seguito, indirizzando il browser all'URL della dashboard, sarà visualizzata l'interfaccia per l'amministrazione, composta da una vista delle tabelle presenti nel database, che sono esplorabili e modificabili come nelle figure 2.10, 2.11 e 2.12.

Figura 2.10: Dashboard per amministratori con tabelle consultabili e modificabili

<input type="checkbox"/>	Id	Username	Nome	Cognome	Ruolo	Azioni
<input type="checkbox"/>	1	123	Davide	Zannirato	ROLE_ADMIN	
<input type="checkbox"/>	2	456	Barbara	Tumiatti	ROLE_USER	
<input type="checkbox"/>	3	789	Alberto	Nale	ROLE_USER	
<input type="checkbox"/>	4	411	Simone	Cuberli	ROLE_USER	
<input type="checkbox"/>	5	454	Francesco	Bergo	ROLE_USER	
<input type="checkbox"/>	6	777	Lorenzo	Mattiolo	ROLE_USER	
<input type="checkbox"/>	7	892	Paolo	Albieri	ROLE_USER	
<input type="checkbox"/>	8	671	Riccardo	Braga	ROLE_USER	
<input type="checkbox"/>	9	971	Fabio	Callegaro	ROLE_USER	
<input type="checkbox"/>	10	523	Alice	Zannirato	ROLE_USER	

Figura 2.11: Esempio di tabella utenti per amministratori, con funzioni di ricerca, creazione, modifica ed eliminazione

The screenshot shows a web application interface for creating a presence stamp. The header is dark with the title "Rilevatore Presenze" on the left and "Disconnetti" and "Modifica Credenziali" on the right. Below the header, there is a breadcrumb trail "link_event_list > link_event_create" and two buttons: "Nuovo" and "Ritorna alla lista". The main form area is divided into sections: "Utente" with a dropdown menu showing "123" and a green checkmark; "Orario" with three dropdown menus for "01", "gen", and "2011"; and "Tipo" with a dropdown menu showing "IN" and a green checkmark. At the bottom of the form, there are two buttons: "Crea" and "Crea e aggiungi un altro".

Figura 2.12: Esempio di creazione timbratura per amministratori

2.3.6 Generazione del Cartellino

Il cartellino è essenzialmente composto da due parti: il titolo e i dati veri e propri divisi per giorno.

Il titolo è formato da una stringa che identifica il soggetto (ad esempio “Nome_Cognome_Username”), una che indica la data e una serie variabile di colonne IN e OUT nelle quali vengono inseriti gli orari se presenti.

Inizialmente viene invocato il metodo per la costruzione della parte dei dati e nel relativo algoritmo vengono conteggiate le coppie di colonne IN/OUT necessarie nel titolo, a seconda del numero massimo di entrate e uscite in un singolo giorno.

In base a questo parametro viene generato il titolo che viene successivamente unito con i dati in una stringa unica.

Se la persona che ha effettuato l’accesso è un utente senza privilegi allora può scegliere direttamente il periodo temporale per controllare le proprie timbrature, altrimenti se è amministratore ha la possibilità di controllare tutti gli utenti, che per maggiore chiarezza vengono visualizzati come una lista.

La lista degli utenti è ottenuta interrogando il database tramite EntityManager, richiedendo tutte le entry della tabella User e inserendole nel layout della pagina tramite una semplice tabella HTML.

Per ogni riga della tabella, quindi per ogni utente, si aggiungono inoltre tre pulsanti, due dei quali per periodi temporali predefiniti (settimanale e mensile), che una volta premuti richiamano l’algoritmo di creazione del cartellino passando come parametri i valori DateTime della fine e dell’inizio dell’analisi, ovvero il timestamp corrente e quest’ultimo meno un periodo di tempo (sette giorni o un mese), grazie alle funzioni PHP per oggetti di tipo DateTime e DateInterval.

L’ultimo pulsante integra la scelta personalizzata del periodo di tempo, sfruttando le funzionalità offerte da JQuery UI, come dialog e datepicker, il primo è utilizzato per far comparire a schermo una finestra mobile contenente due text-box che con un click richiamano datepicker, facendo comparire un calendario, con le dovute restrizioni, nel quale selezionare le date di inizio e di fine a piacere, il risultato è visibile nell’immagine 2.13.

Ricapitolando, se dal Pannello Personale viene richiesta la funzione cartellino, vengono prima controllati i permessi dell’utente richiedente e viene renderizzata o la lista con tutti gli utenti con allegate scelte temporali se amministratore oppure solo le scelte temporali personali, al momento della conferma del periodo viene effettuata una richiesta POST gestita da un controller che estrae i parametri “start”, “end” e “user” e li utilizza a sua volta per estrarre le timbrature dal database e passarle come parametro array per l’algoritmo di generazione cartellino, ricevendo in risposta da esso un file testuale formattato per essere inserito in una tabella HTML con i risultati.

L’algoritmo di generazione e formattazione del cartellino analizza un elemento per volta dell’array di timbrature ricevuto e possono presentarsi le seguenti possibilità:

- La data della timbratura non è cambiata rispetto all'ultimo elemento ispezionato, non occorre iniziare una nuova riga, la marcatura può essere:
 - *Ingresso avvenuto dopo un altro ingresso*: Errato, viene lasciata una casella vuota per il campo OUT della timbratura precedente.
 - *Ingresso avvenuto dopo un'uscita*: Giusto, viene inserito subito l'orario.
 - *Uscita avvenuta dopo un ingresso*: Giusto, viene inserito subito l'orario.
 - *Uscita avvenuta dopo un'altra uscita*: Errato, viene lasciata una casella vuota per il campo IN della presente timbratura.
- La data della timbratura è cambiata rispetto alla precedente, viene iniziata una nuova riga, saltata la cella del nome utente e scritta la nuova data. La marcatura può essere:
 - *Ingresso*: Giusto, viene inserito subito l'orario.
 - *Uscita*: Errato, un giorno non può iniziare con una uscita, viene lasciata una casella vuota per il campo IN della presente timbratura.

Terminata l'analisi si riceve la stringa che funge da corpo dell'elemento table nella pagina HTML, ovvero il "tbody", preceduto dall'intestazione con i titoli, "thead", ottenuta da un altro metodo scritto ad-hoc che a seconda dell'utente e del numero di coppie IN/OUT la genera.

Per concludere vengono unite queste due parti e passate come parametro per la renderizzazione di una nuova pagina dei risultati, nella quale viene visualizzata la tabella, come in figura 2.14.

2.3.7 Esportazione in CSV

Per ogni cartellino visualizzato viene aggiunto un link (nel template relativo alla visualizzazione dei risultati) collegato alla route che richiama la funzione di esportazione nel formato CSV.

Questa volta viene scelto il passaggio dei parametri tramite metodo "GET", quindi direttamente nella URL, facendo sempre i dovuti controlli sui permessi dell'utente richiedente.

Il funzionamento di base è lo stesso della generazione del cartellino, non appena viene premuto il pulsante di esportazione la richiesta GET viene analizzata dal controller che estrae i parametri, interroga il database e l'array di risultati ottenuto lo utilizza come parametro per il metodo di generazione del file CSV.

L'algoritmo è il medesimo della generazione del cartellino, con la sola differenza della formattazione della stringa, non più a tag per la tabella HTML ma con caratteri appositi dei file CSV, ovvero ";" per passare alla cella successiva e "\n" per iniziare una nuova riga.

The screenshot shows the 'Rilevatore Presenze' interface. At the top right, there are links for 'Disconnetti' and 'Modifica Credenziali'. Below the header, the text 'Scegli un intervallo temporale' is displayed. On the left, there is a gear icon and a calendar icon showing the date '15'. The main area contains a table of users with columns for 'Username', 'Nome', 'Cognome', and 'Timbrature'. Each user row has three buttons: 'Settimana', 'Mese', and 'Personalizza'. A modal window titled 'Seleziona intervallo' is open, showing date pickers for 'Da' (2011-07-06) and 'A' (2011-07-18). A calendar is displayed below the date pickers, with the month 'Lug' and year '2011' selected. The calendar shows the days of the week (Lu, Ma, Me, Gi, Ve, Sa, Do) and the dates from 1 to 31.

Username	Nome	Cognome	Timbrature
123	Davide	Zannirato	Settimana Mese Personalizza
456	Foo	Bar	Settimana Mese Personalizza
789	Mario	Ciao	Settimana Mese Personalizza

Figura 2.13: Come viene selezionato l'utente e la data per la generazione del cartellino

The screenshot shows the 'Rilevatore Presenze' interface with the title 'Timbrature'. At the top right, there are links for 'Disconnetti' and 'Modifica Credenziali'. On the left, there is a gear icon and a calendar icon showing the date '15'. The main area contains a table of attendance records for the user 'Davide_Zannirato_123'. The table has columns for 'Giorno', 'IN', and 'OUT'. A button 'Esporta in CSV' is located at the top right of the table. The table shows two rows of attendance data for the dates 10-07-2011 and 11-07-2011.

Davide_Zannirato_123	Giorno	IN	OUT	IN	OUT
	10-07-2011	07:56:00	12:08:00		17:11:00
	11-07-2011	08:00:00	12:01:00	13:05:00	17:03:00

Figura 2.14: Come viene visualizzato il cartellino di un utente

Vengono poi inserite le istruzioni per reindirizzare il browser e allegare il file appena creato, consentendo il salvataggio in locale.

	A	B	C	D	E	F	G
1	Davide Zannirato 123	Giorno	IN	OUT	IN	OUT	
2		10-07-2011	07:56:00	12:08:00		17:11:00	
3		11-07-2011	08:00:00	12:01:00	13:05:00	17:03:00	
4							
5							
6							

Figura 2.15: Esempio di cartellino esportato in un file CSV

Capitolo 3

Conclusioni

L'applicativo creato soddisfa tutti i requisiti, il suo sviluppo ad-hoc ha fatto sì che in poco tempo sia stata creata una web application su misura e con tutti i presupposti di modularità e scalabilità per implementazioni future, grazie anche all'utilizzo di soli strumenti Open Source e alla struttura a bundle del framework Symfony 2.0.

L'interfaccia finale è chiara e pulita, con pochi pulsanti per le funzionalità, le quali possono essere richiamate in breve tempo e con pochi click.

Gli amministratori hanno a disposizione una sezione apposita che come richiesto offre un controllo completo sui dati, e i cartellini generabili rispettano il formato discusso nei requisiti dettati dal committente.

I semplici utenti potranno sempre controllare i loro dati dal proprio computer e generare dei resoconti, nella più completa trasparenza.

L'implementazione del nuovo sistema di timbratura velocizza e ottimizza tutte le azioni ad esso collegate, dalla semplice marcatura di ingresso e uscita ai calcoli per adempiere alle mansioni della segreteria, infatti grazie all'eliminazione del supporto cartaceo vengono ridotti gli errori, velocizzati tutti i processi e salvaguardate tutte le informazioni in modo migliore e sicuro, grazie al database apposito.

Gli sviluppi futuri potrebbero riguardare la generazione automatica di altri moduli altrimenti cartacei, come l'inserimento di missioni di lavoro o trasferte e la richiesta di ferie o permessi.

Oppure gestire le rimanenti ferie di ogni utente, le ore di straordinari, di permessi, di malattia o di maternità e il calcolo delle ore nette e arrotondate di lavoro, con l'aggiunta delle tolleranze di orario accettabili.

Inoltre potrebbe essere interessante la possibilità di timbrare con dispositivi di prossimità, come i transponder RFID, diminuendo ulteriormente i tempi di timbratura, oppure tramite un sistema di lettura di impronte digitali.

Bibliografia

- [1] “Symfony 2.0 the book.” Online available <http://symfony.com/doc/current/book/index.html>.
- [2] “Doctrine 2.1 reference documentation.” Online available <http://www.doctrine-project.org/docs/orm/2.1/en/>.
- [3] “Adminbundle documentation.” Online available <http://rabaix.net/AdminBundle/html/index.html>.
- [4] C. Ltd., “Ubuntu server guide.” Online available <https://help.ubuntu.com/10.04/serverguide/C/index.html>.
- [5] “Orangehrm.” Online available <http://www.orangehrm.com/>.
- [6] “Smartsuite - bioenable technologies.” Online available <http://smartsuite.bioenabletech.com/>.
- [7] “Tulip - bloomtech.” Online available <http://www.bloomtech.it/introduzione-tulip.html>.
- [8] “Php documentation.” Online available <http://www.php.net/manual/en/>.
- [9] “Mysql 5.6 documentation.” Online available <http://dev.mysql.com/doc/refman/5.6/en/index.html>.
- [10] “jquery documentation.” Online available http://docs.jquery.com/Main_Page.

Elenco delle tabelle

1.1	Esempio di Cartellino	2
2.1	Dizionario delle entità	16

Elenco delle figure

1.1	OrangeHRM screenshot	5
1.2	SmartSuite screenshot	6
1.3	TulipWeb screenshot	8
1.4	Confronto dei software sulle funzioni offerte	10
2.1	Casi d'uso	11
2.2	Casi d'uso per gli utenti	13
2.3	Casi d'uso per gli amministratori	14
2.4	Schema ER	16
2.5	Schema Logico	17
2.6	Flusso operativo di Symfony 2.0	20
2.7	Interfaccia di front-end	25
2.8	Messaggi nell'interfaccia di front-end	26
2.9	Sezione di Login	27
2.10	Dashboard per amministratori	28
2.11	Tabella utenti per amministratori	28
2.12	Creazione timbratura per amministratori	29
2.13	Selezione data per generare il cartellino	32
2.14	Cartellino generato	32
2.15	Cartellino esportato in un file CSV	33

Ringraziamenti

Un grazie particolare al relatore, il Prof. Massimo Rumor, per la disponibilità concessa sia durante le lezioni che nel periodo di stesura della tesi.

Un ringraziamento al tutor aziendale, l'ingegner Eduard Roccatello, e agli ingegneri Davide Grossato, Luca Giandoso e Francesco Dolcetto, il tirocinio è stato piacevole e molto istruttivo.

Un grazie di cuore ai miei genitori Claudio ed Emanuela e a mia sorella Alice, senza la vostra fiducia e il vostro sostegno nei periodi di difficoltà tutto questo non sarebbe mai stato possibile, spero di aver ripagato i vostri sacrifici, finalmente il sogno si è avverato.

Per i weekend passati in casa a studiare nei periodi d'esame, per l'interesse e la pazienza ringrazio Barbara, la mia ragazza, la sua dolcezza è stata la giusta cura alle mie preoccupazioni.

Un ringraziamento speciale a tutti gli zii e parenti, in particolare a Giuseppe e Mario, a nonna Giovanna e ai nonni Giuseppe e Lina, le vostre preghiere alla fine sono state ascoltate e premiate.

Un grazie ai miei compagni universitari, i "Los Mejores", le lezioni in compagnia vostra scorrevano come niente fosse, l'aiuto reciproco per gli esami è stato indispensabile e le continue risate hanno fatto nascere delle amicizie che dureranno nel tempo.

Ringrazio tutta la compagnia di amici, soprattutto quelli storici Sandro, Marco "Sak" e Denis, siamo cresciuti assieme e non dimenticherò mai le mille avventure, la nostra amicizia non finirà mai.

I sacrifici sono stati molti, i periodi di stress innumerevoli, ma è stato grazie alle persone che mi hanno seguito e che sono state al mio fianco che sono giunto a questo traguardo.

Grazie di cuore a tutti.

Davide.