

Formulario di valutazione LPI

Titolo del progetto: Formulario di valutazione LPI

Alunno/a: Matteo Forni

Classe: Info 4AA

Anno scolastico: 2019/2020

Docente responsabile: Massimo Sartori

1	Introduzione	5
1.1	Informazioni sul progetto	5
1.2	Abstract	5
1.3	Scopo	5
2	Analisi	5
2.1	Analisi del dominio	5
2.2	Analisi e specifica dei requisiti	6
2.3	Use case	9
2.4	Pianificazione	9
2.4.1	Sezione analisi	12
2.4.2	Sezione progettazione	12
2.4.3	Sezione implementazione	13
2.4.4	Sezione test	13
2.5	Analisi dei mezzi	14
2.5.1	Software	14
2.5.2	Hardware	14
3	Progettazione	15
3.1	Design dell'architettura del sistema	15
3.2	Design dei dati e database	16
3.2.1	Schema ER	16
3.2.2	Schema logico	17
3.2.3	Specifica dei dati	18
3.3	Design delle interfacce	19
3.3.1	Pagina di login	19
3.3.2	Pagina di registrazione	20
3.3.3	Pagina di reset della propria password	21
3.3.4	Pagina contenente lo storico	21
3.3.5	Pagina di aggiunta di un formulario	22
3.3.6	Pagina di aggiunta delle motivazioni	23
3.3.7	Pagina di resoconto finale	24
3.3.8	Pagina di amministrazione	25
4	Implementazione	26
4.1	Lumen	26

4.1.1	Installazione	27
4.1.2	La struttura delle cartelle.....	27
4.1.3	Blade template	29
4.1.4	Gestione delle dipendenze	29
4.2	Gestione delle versioni	29
4.3	Database	29
4.4	Connessione con il database	30
4.5	Model.....	30
4.6	Routes	31
4.7	Middleware	31
4.8	Controllers	32
4.8.1	Validazione dei dati	32
4.8.2	Lettura di un elemento	33
4.8.3	Aggiunta di un elemento	34
4.8.4	Modifica di un elemento	34
4.8.5	Eliminazione.....	35
4.8.6	Query strutturate	35
4.8.7	Creazione dei PDF.....	36
4.9	Views	39
4.9.1	Pagina di login	39
4.9.2	Pagina di registrazione	39
4.9.3	Pagina di amministrazione.....	40
4.9.4	Pagina dello storico.....	41
4.9.5	Pagina di aggiunta	42
4.9.6	Pagina di aggiunta delle motivazioni	43
4.9.7	Pagina dei risultati.....	44
5	Test.....	44
5.1	Protocollo di test.....	44
5.2	Risultati test.....	51
5.3	Mancanze/limitazioni conosciute.....	52
6	Consuntivo.....	53
7	Conclusioni	54
7.1	Sviluppi futuri.....	54

7.2	Considerazioni personali	54
8	Bibliografia	55
8.1	Sitografia	55
9	Glossario	56
10	Indice delle figure	57
11	Allegati	57

1 Introduzione

1.1 Informazioni sul progetto

- Titolo: Formulario di valutazione LPI
- Durata: 23.01.2020-06.04.2020
- Allievo coinvolto: Matteo Forni
- Classe: Info 4AA
- Anno scolastico: 2019-2020
- Docente responsabile: Massimo Sartori

1.2 Abstract

The aim of this project is to create a website that allows the creation and management of evaluation forms for the LPIs carried out by computer science students in their last year of school. Thanks to this product, it will be possible to add motivations, combined with the evaluation points that will be used later by the teachers to justify the assigned grades. The site will therefore be useful for those responsible for evaluations but also for students who will have clearer explanations. Teachers will be able to select the most suitable explanations through filters and these will be inserted in the document and they will modify the final evaluation which can then be saved in PDF and printed out.

1.3 Scopo

Lo scopo di questo progetto è quello di sviluppare un sito web che consenta di creare e gestire dei formulari di valutazione degli LPI svolti dagli apprendisti di informatica all'ultimo anno di formazione. I moduli saranno redatti secondo il nuovo modello definito nell'ordinanza del 2014. Grazie a questo prodotto potrà essere possibile l'aggiunta di motivazioni, abbinate ai punti di valutazione, da parte di un amministratore che verranno, in seguito, utilizzate dai docenti per giustificare le note assegnate. Il sito tornerà dunque utile ai responsabili delle valutazioni ma anche agli allievi che avranno delle spiegazioni più chiare.

I docenti potranno selezionare tramite dei filtri le motivazioni più adatte al caso ed esse verranno inserite nel documento e andranno a modificare la valutazione finale che potrà poi venire salvata in formato PDF e anche stampata.

2 Analisi

2.1 Analisi del dominio

Il prodotto dovrà essere un gestore di formulari, per la valutazione degli LPI degli informatici, in forma digitale, attualmente esiste unicamente la versione sia digitale che cartacea dei moduli ma sono entrambi da compilare a mano. Il sito ha quindi come obiettivo quello di velocizzare le operazioni di assegnazione delle note e di eliminare gli errori umani, le valutazioni finali verranno infatti calcolate automaticamente dal sistema che si occuperà anche di formattarle e inserirle in un PDF che potrà poi venire stampato e/o salvato. Il risultato finale generato per i docenti dovrà essere conforme allo standard deciso nell'ordinanza del 2014.

Gli utenti saranno docenti, formatori e periti di esame con buone conoscenze di informatica ma il sito dovrà comunque essere di facile utilizzo e intuitivo. Il bisogno principale degli utenti è di eliminare le incongruenze tra più valutazioni diverse utilizzando delle motivazioni standard che verranno inserite dagli amministratori del sistema.

2.2 Analisi e specifica dei requisiti

ID: REQ-01	
Nome	Creazione database
Priorità	1
Versione	1.0
Note	-
Sotto requisiti	
001	Bisognerà inserire in maniera hard-coded i punti di valutazione

ID: REQ-02	
Nome	Creazione interfaccia di registrazione
Priorità	1
Versione	1.0
Note	-
Sotto requisiti	
001	Bisognerà inserire i campi nome, cognome, email e telefono

ID: REQ-03	
Nome	Creazione interfaccia di login
Priorità	1
Versione	1.0
Note	Si dovrà avere un account per poter accedere

ID: REQ-04	
Nome	Creazione interfaccia di amministrazione
Priorità	1

Versione	1.0
Note	Per accedervi si dovranno avere i permessi di amministratore (REQ-03)
Sotto requisiti	
001	Si dovrà poter inserire/modificare/eliminare le motivazioni
002	Le motivazioni potranno essere abbinate a un solo punto
003	Si dovrà poter inserire/modificare/eliminare i docenti

ID: REQ-05	
Nome	Creazione interfaccia per i docenti
Priorità	1
Versione	1.0
Note	Per accedervi si dovrà aver creato un account
Sotto requisiti	
001	Si dovrà poter creare/modificare/eliminare dei formulari
002	Si dovrà poter visualizzare lo storico dei propri formulari

ID: REQ-06	
Nome	Creazione interfaccia di sviluppo del formulario
Priorità	1
Versione	1.0
Note	Bisognerà aver creato un nuovo formulario per accedervi
Sotto requisiti	
001	Si dovrà poter scegliere le motivazioni in base alla sezione, punto o parola chiave
002	Si dovranno poter inserire i dati dei periti/docenti/allievi coinvolti

ID: REQ-07	
Nome	Calcolo dei risultati
Priorità	1
Versione	1.0
Note	Bisognerà calcolare i risultati secondo il nuovo modello
Sotto requisiti	
001	Si dovranno calcolare i risultati in base alle motivazioni inserite

ID: REQ-08	
Nome	Creazione dei PDF
Priorità	1
Versione	1.0
Note	-
Sotto requisiti	
001	Si dovranno generare i PDF contenenti i risultati con il formato del nuovo modello
002	Si dovrà poter salvare i file

2.3 Use case

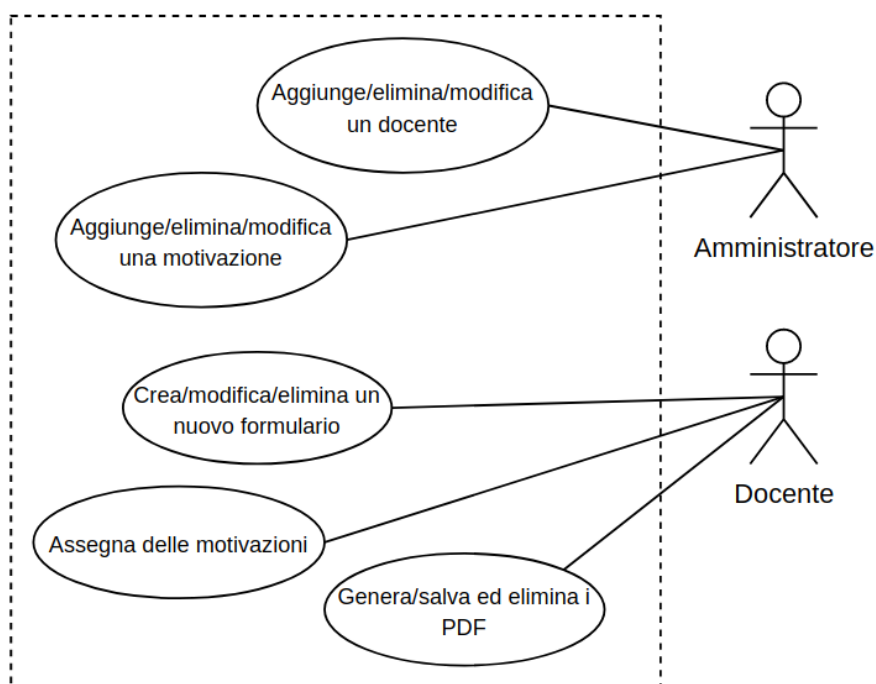


Figura 1 Diagramma degli use case

La foto soprastante rappresenta le use case del prodotto, come si può vedere vi saranno due attori che saranno rispettivamente gli amministratori e i docenti. Entrambi gli attori rappresentati nel diagramma avranno già creato un account valido ed effettuato il login con esso.

Gli admin del sistema potranno aggiungere, modificare ed eliminare gli utenti dei docenti oltre ad avere la possibilità di inserire, modificare ed eliminare delle motivazioni ed assegnarle ai rispettivi punti. I docenti potranno invece generare dei formulari a cui andranno ad assegnare delle motivazioni scelte fra quelle inserite in precedenza dagli amministratori ai vari punti di valutazione. Inoltre potranno, una volta terminata la valutazione, salvare il risultato in PDF.

2.4 Pianificazione

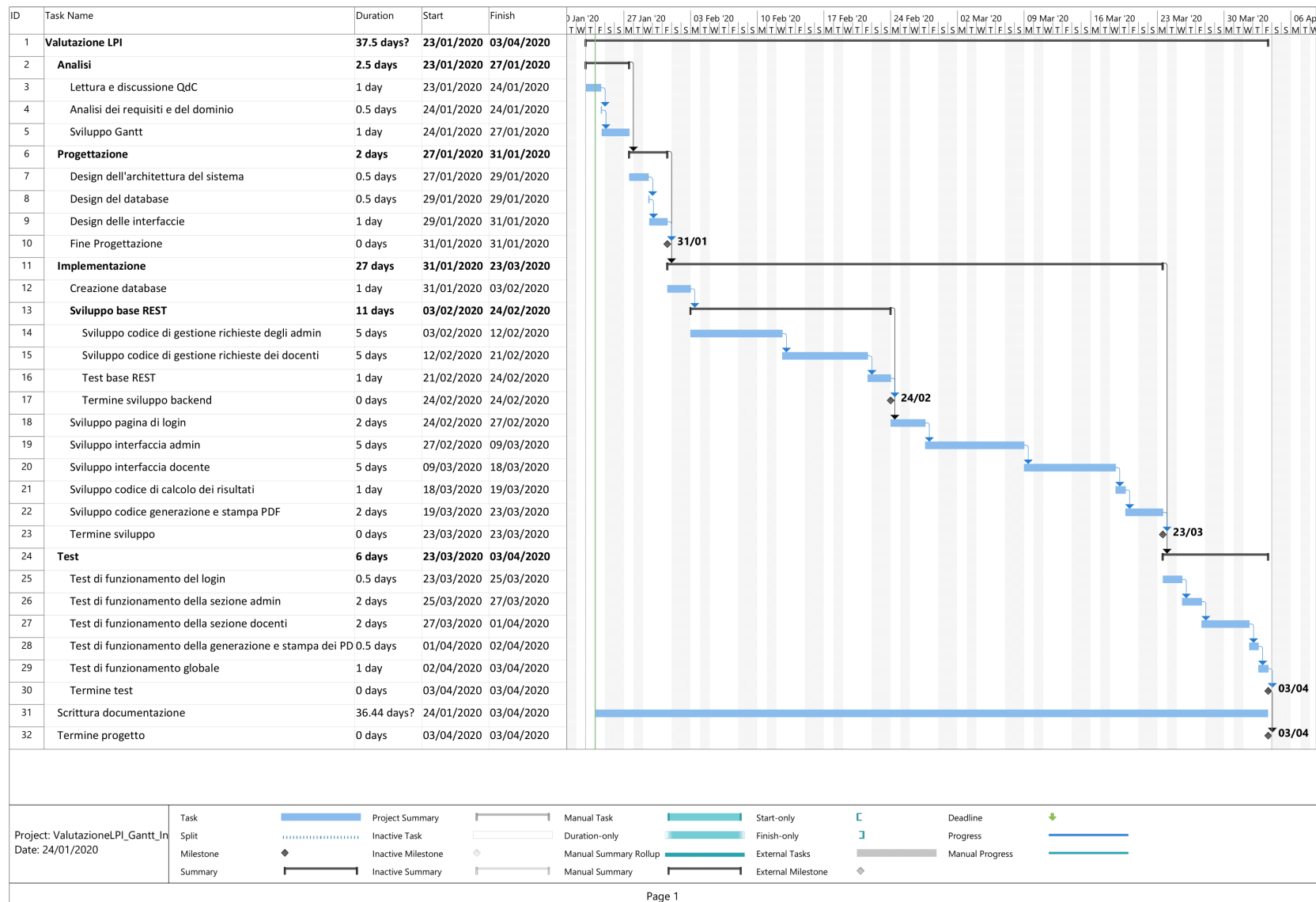


Figura 2 Diagramma di Gantt preventivo

Nella figura superiore si può vedere il diagramma di Gantt completo, esso rappresenta in maniera indicativa come dovrebbe andare il progetto. Esso è stato diviso in quattro grandi categorie che sono Analisi, Progettazione, Implementazione e Test, oltre ad esse vi è la documentazione che procede lungo tutto il progetto dato che verrà completata con l'avanzare delle altre attività.

2.4.1 Sezione analisi

2	Analisi	2.5 days	23/01/2020	27/01/2020	
3	Lettura e discussione QdC	1 day	23/01/2020	24/01/2020	
4	Analisi dei requisiti e del dominio	0.5 days	24/01/2020	24/01/2020	
5	Sviluppo Gantt	1 day	24/01/2020	27/01/2020	

Figura 3 Diagramma Gantt, sezione analisi

La sezione iniziale è dedicata alla parte di analisi che verrà effettuata ad inizio progetto e comprenderà tre capitoli fondamentali che sono: la lettura e discussione del quaderno dei compiti con il docente responsabile, l'analisi dei requisiti e del dominio e la creazione del diagramma di Gantt. Questi tre punti sono fondamentali per comprendere tutti gli agenti e particolari del progetto. Queste tre attività sono piuttosto veloci da eseguire ed ho quindi previsto di metterci tre giorni lavorativi per portarle a termine.

2.4.2 Sezione progettazione

6	Progettazione	2 days	27/01/2020	31/01/2020	
7	Design dell'architettura del sistema	0.5 days	27/01/2020	29/01/2020	
8	Design del database	0.5 days	29/01/2020	29/01/2020	
9	Design delle interfacce	1 day	29/01/2020	31/01/2020	
10	Fine Progettazione	0 days	31/01/2020	31/01/2020	

Figura 4 Diagramma Gantt, sezione progettazione

La seconda sezione del diagramma è la progettazione che comprende lo sviluppo teorico del sistema così da facilitarne l'implementazione effettiva. Questo sotto capitolo è stato suddiviso in tre attività che sono: il design dell'architettura del sistema, il design del database e il design delle interfacce. Inoltre esso termina con una milestone che rappresenta il termine della progettazione e l'inizio dello sviluppo e si situa al 31 di gennaio.

Le tre attività della sezione sono piuttosto veloci da completare ed ho quindi stimato di impiegarci mezza giornata lavorativa per le prime due e una intera per la terza.

2.4.3 Sezione implementazione

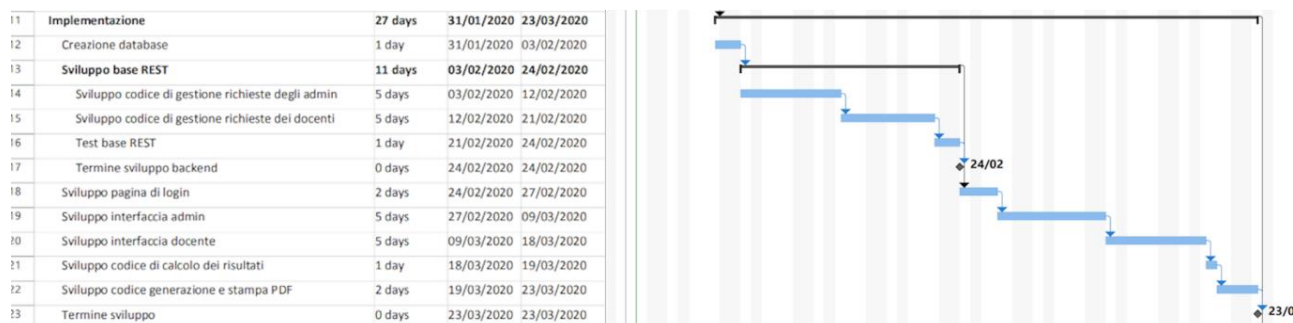


Figura 5 Diagramma Gantt, sezione implementazione

La terza sezione rappresenta l'implementazione, essa sarà la parte più lunga ed impegnativa del progetto e durerà circa un mese lavorativo. Essa inizia con lo sviluppo del database su MySQL che verrà seguito dalla creazione del codice di gestione delle richieste. Una volta che esso sarà stato testato si inizierà a sviluppare le pagine front-end e a collegarle con il back-end, infine si svilupperà la sezione di creazione dei PDF.

La sezione termina con una milestone che rappresenta il termine dell'implementazione e si situa il 23 marzo.

2.4.4 Sezione test



Figura 6 Diagramma Gantt, sezione test

L'ultima parte del progetto sarà riservata ai test di funzionamento del sistema inizialmente specifici per le pagine ed infine generali per verificare l'integrazione dei vari componenti. Essi occuperanno circa una settimana lavorativa ed al loro termine finirà pure il progetto come rappresentato dalla milestone situata al 3 aprile.

2.5 Analisi dei mezzi

2.5.1 Software

I software che sono stati utilizzati per questo progetto sono i seguenti:

- Microsoft Office Word 2016 Utilizzato per redigere la documentazione
- Microsoft Office Powerpoint 2016 Utilizzato per sviluppare la presentazione
- Microsoft Project Professional 2016 Utilizzato per creare il diagramma di Gantt
- Visual Studio Code 1.43.2 Utilizzato per scrivere il codice
- Google Chrome 77 Utilizzato per testare il progetto
- Mozilla Firefox 69 Utilizzato per testare il progetto
- MySQL Workbench 6.3.8 Utilizzato per gestire il database
- VmWare 15.5.2 Utilizzato per gestire la virtuale Windows con il pacchetto office

Mentre le librerie utilizzate sono le seguenti:

- Material Design Bootstrap 4.8.7 Utilizzata per la grafica del sito e dei controlli Javascript
- PHPMailer 6.1 Utilizzata per inviare le email
- Cookie JS 3.0.0 Utilizzata per gestire i cookies con Javascript
- TCPDF 6.2.2 Utilizzata per generare i file PDF
- Toastr 2.1.4 Utilizzata per generare le notifiche a schermo
- Firebase JWT 5.0.0 Utilizzata per gestire i token JSON

2.5.2 Hardware

L'hardware su cui verrà svolto il progetto e su cui potrà essere utilizzato in futuro è un normale PC, in questo caso un laptop Dell XPS 15 9570, su cui vi è installato un webserver funzionante che funzioni con il pattern MVC di PHP, quindi con il modulo di rewrite attivato e l'override delle cartelle consentito.

3 Progettazione

3.1 Design dell'architettura del sistema

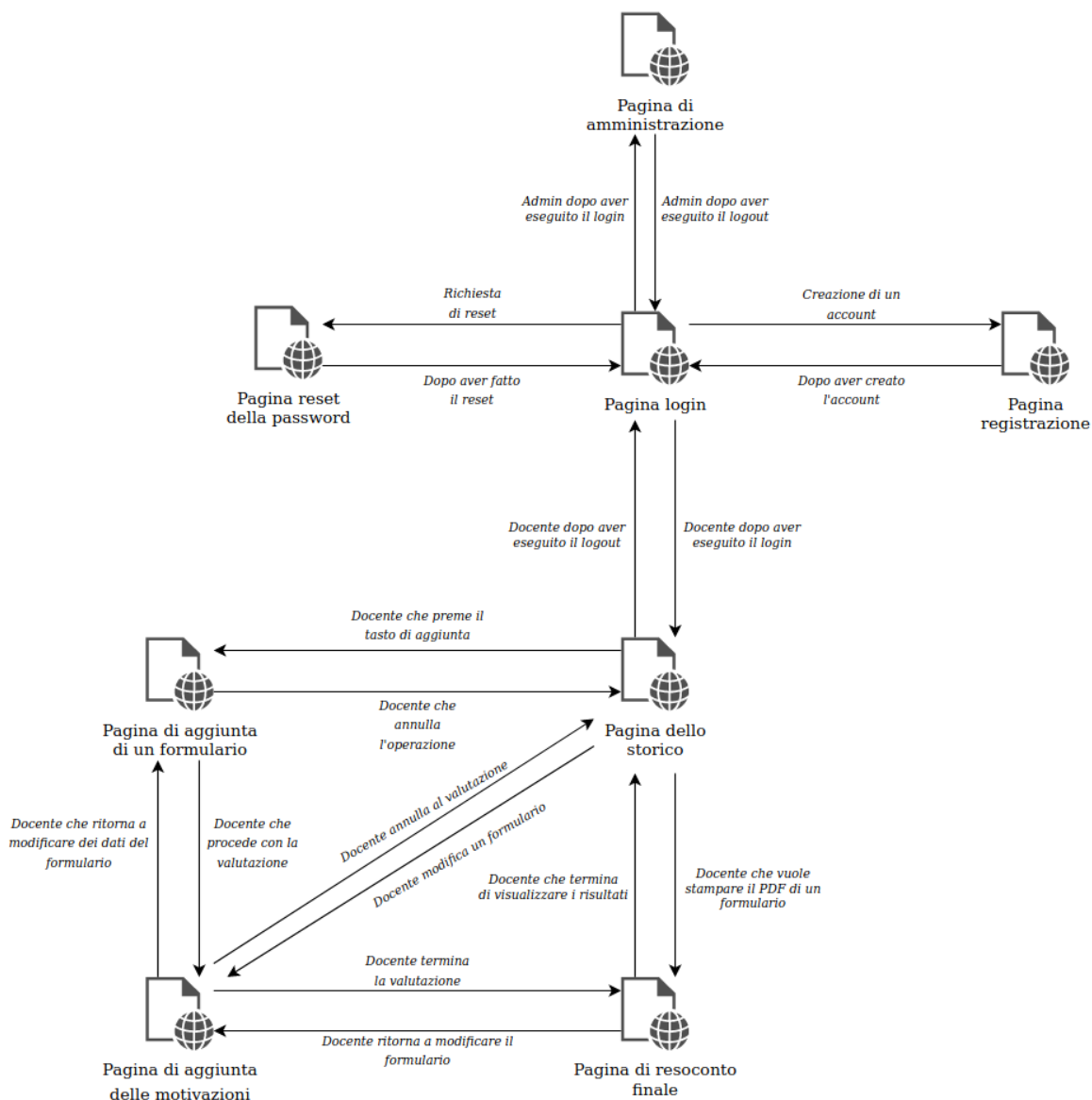


Figura 7 Diagramma dell'architettura del sistema

Il diagramma soprastante rappresenta le interazioni basilari e più importanti che saranno presenti nel prodotto, esso contiene ogni pagina con le operazioni che verranno eseguite maggiormente e la spiegazione di come gli utenti navigheranno tra esse. Questo non vuol dire che se nel diagramma le pagine non sono collegate esse non saranno accessibili fra loro ma che sarebbe un'operazione che non segue il flusso abituale del sito.

La pagina più centrale sarà quella di login dato che per utilizzare il prodotto bisognerà essere loggati, se l'utente non ha un account potrà accedere da essa alla sezione per registrarsi mentre se avrà dimenticato la propria password avrà la possibilità di reimpostarla tramite la pagina apposita. Una volta eseguito l'accesso

l'utente si troverà nella prima pagina predefinita per un account con i suoi permessi: se amministratore verrà reindirizzato alla pagina riservata agli admin altrimenti a quella contenente lo storico dei propri formulari.

Dalla pagina che mostra la lista delle valutazioni eseguite in passato si potrà modificare, creare o stampare un formulario. Se si sceglierà la prima opzione si verrà reindirizzati alla pagina in cui si potranno gestire le motivazioni in un formulario, premendo il tasto di aggiunta si verrà reindirizzati alla pagina di creazione, che porterà in seguito a quella di gestione delle motivazioni, mentre se si vorrà stampare un formulario si verrà trasportati nella pagina di resoconto della valutazione dove vi saranno i bottoni per la stampa.

3.2 Design dei dati e database

3.2.1 Schema ER

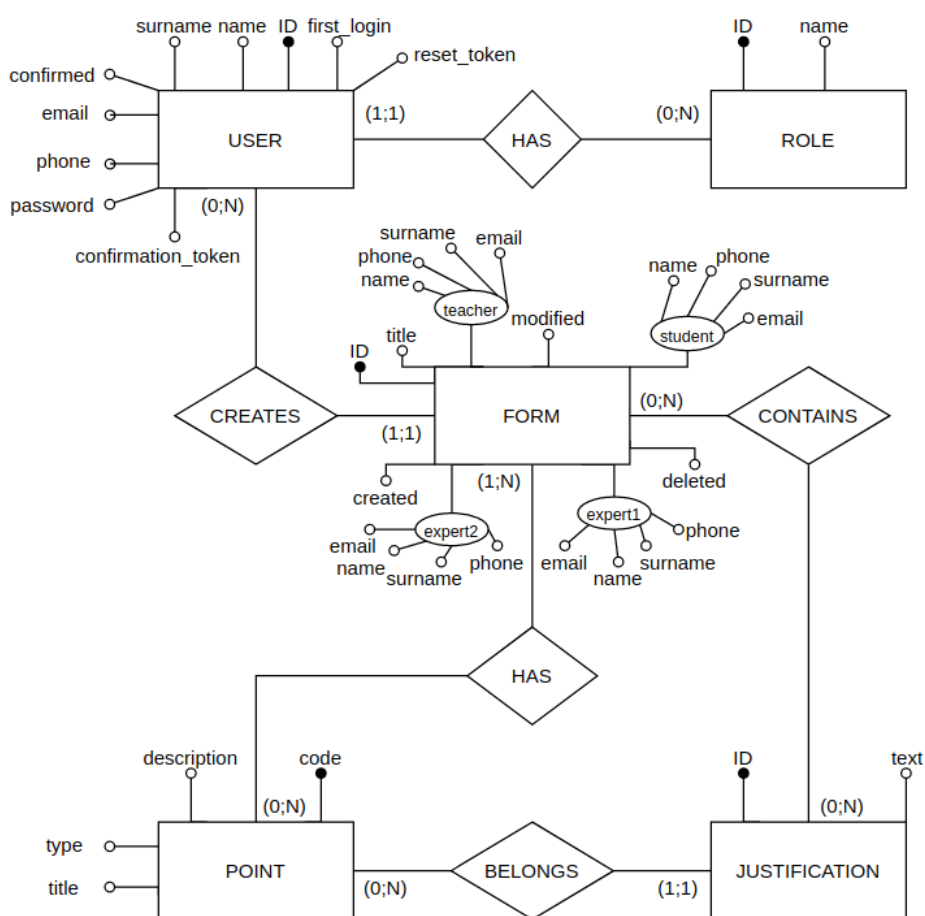


Figura 8 Diagramma ER del database

Il diagramma ER del database è piuttosto semplice e, come si può vedere dallo schema soprastante, comprende cinque tabelle che rappresentano rispettivamente: gli utenti del sistema, il ruolo che ricoprono, i punti di valutazione del LPI, le motivazioni a loro collegate e i formulari generati.

La prima tabella contiene le informazioni relative agli utenti che utilizzano il prodotto con le loro informazioni di base, essa prende il ruolo di ogni account dalla tabella `role`. Questo poteva essere sostituito con un attributo nella tabella `user` compilato manualmente alla creazione di ogni account ma avrebbe reso piuttosto complicato un futuro sviluppo del progetto. Il campo `confirmed` contiene un booleano che indicherà se l'utente avrà eseguito la conferma della propria email o meno. La tabella `point` contiene le informazioni dei

punti di valutazione presi dai criteri di valutazione LPI (estesi) e li identifica secondo il codice a loro assegnata nel file di provenienza, il campo non sarà quindi un intero ma una stringa. Ogni punto avrà un tipo che sarà o *generale* oppure *specifico* ed esso viene specificato nel campo *type* che verrà settato a *true* se il punto sarà uno dei sette che potranno aggiungere i docenti o *false* se sarà uno generico di base.

La tabella delle motivazioni conterrà soltanto un identificatore e il nome di essa, che è la stringa che verrà poi stampata nel file finale.

Infine la tabella più grande e complessa, quella relativa ai formulari, conterrà tutte le informazioni delle persone coinvolte nel LPI e quindi: studente, docente responsabile e i due periti. Oltre a ciò essa conterrà pure le date di creazione, ultima modifica e eliminazione così da rendere possibile la creazione dello storico di ogni docente.

3.2.2 Schema logico

ROLE (ID, name)

USER (ID, name, surname, email, phone, password, confirmed, first_login, confirmation_token*, reset_token*, id_role(FK)) *email, phone unique*

POINT (code, title, description*, type)

JUSTIFICATION (ID, text, id_point(FK))

FORM (ID, title, created, modified*, deleted*, student_name, student_surname, student_email, student_phone, teacher_name, teacher_surname, teacher_email, teacher_phone, expert1_name*, expert1_surname*, expert1_email*, expert1_phone*, expert2_name*, expert2_surname*, expert2_email*, expert2_phone*, id_user(FK))

HAS (id_form(FK), id_point(FK))

CONTAINS (id_form(FK), id_justification(FK))

Figura 9 Schema logico del database

Lo schema logico rende più chiari alcuni punti del database e di come andrà a diventare effettivamente una volta generato in MySQL. Dall'immagine si può capire che la tabella *user* riceverà la chiave del ruolo così come *justification* da *point* e *form* da *user*. Questo fa inoltre comprendere che la relazione situata fra i formulari e le motivazioni diverrà una tabella a sé che conterrà gli identificatori delle motivazioni con quelli dei formulari alla quale sono collegate, la stessa situazione è presente tra i punti di valutazione e i formulari.

3.2.3 Specifica dei dati

Tabella ROLE

Attributo	Tipo	Descrizione
ID	int	Chiave primaria
name	varchar(50)	Il nome del ruolo

Tabella USER

Attributo	Tipo	Descrizione
ID	int	Chiave primaria
name	varchar(100)	Il nome dell'utente
surname	varchar(100)	Il cognome dell'utente
email	varchar(100)	L'email dell'utente, dovrà essere unica fra tutti i record (unique)
phone	varchar(50)	Il num. di telefono, dovrà essere unico fra tutti i record (unique)
password	varchar(255)	La password, verrà salvato l'hash di quella inserita dall'utente
confirmed	boolean	Booleano specificante se l'utente ha confermato la sua email
id_role	int	L'id del ruolo, se si eliminerà il ruolo accadrà anche all'utente
first_login	boolean	Se l'utente è stato creato da un admin e si collega la prima volta
confirmation_token	varchar(100)	Il token di conferma dell'email dell'account
reset_token	varchar(100)	Il token di reset della password

Tabella POINT

Attributo	Tipo	Descrizione
code	varchar(10)	Chiave primaria, riprende i codici dei criteri di valutazione
title	varchar(255)	Il titolo del punto di valutazione
description	longtext	La descrizione del punto di valutazione, può essere nulla
type	boolean	Il tipo di punto (se true: specifico altrimenti: generico)

Tabella JUSTIFICATION

Attributo	Tipo	Descrizione
ID	int	Chiave primaria
text	longtext	Il nome della giustificazione
id_point	varchar(100)	La chiave del punto, se si elimina il punto succede anche alla motivazione

Tabella FORM

Attributo	Tipo	Descrizione
ID	Int	Chiave primaria
title	varchar(255)	Il titolo del progetto che si andrà a valutare
created	datetime	La data di creazione del formulario
modified	datetime	La data di ultima modifica del formulario, può essere nulla
deleted	datetime	La data di eliminazione del formulario, può essere nulla
student_name	varchar(100)	Il nome dello studente
student_surname	varchar(100)	Il cognome dello studente
student_email	varchar(100)	L'email dello studente
student_phone	varchar(50)	Il numero di telefono dello studente
teacher_name	varchar(100)	Il nome del docente
teacher_surname	varchar(100)	Il cognome del docente
teacher_email	varchar(100)	L'email del docente
teacher_phone	varchar(50)	Il numero di telefono del docente
expert1_name	varchar(100)	Il nome del primo perito

expert1_surname	varchar(100)	Il cognome del primo perito
expert1_email	varchar(100)	L'email del primo perito
expert1_phone	varchar(50)	Il numero di telefono del primo perito
expert2_name	varchar(100)	Il nome del secondo perito
expert2_surname	varchar(100)	Il cognome del secondo perito
expert2_email	varchar(100)	L'email del secondo perito
expert2_phone	varchar(50)	Il numero di telefono del secondo perito
id_user	int	L'id dell'utente che ha creato il formulario

Tutti gli attributi dei periti possono essere lasciati nulli in caso il formulario sia di valutazione di un progetto di semestre e non di un LPI.

Tabella CONTAINS

Attributo	Tipo	Descrizione
id_justification	int	Chiave esterna della motivazione e chiave primaria
id_form	int	Chiave esterna del formulario e chiave primaria

Tabella HAS

Attributo	Tipo	Descrizione
id_point	varchar(10)	Chiave esterna del punto di valutazione e chiave primaria
id_form	int	Chiave esterna del formulario e chiave primaria

3.3 Design delle interfacce

3.3.1 Pagina di login

Valutazione LPI

Registrati

Email

Password

Accedi

Cambia password

Copyright SAMT 2020

Figura 10 Mockup della pagina di login

La pagina di login dovrà essere semplice e chiara, conterrà semplicemente un form di inserimento che consentirà all'utente di inserire la propria email e la relativa password. Sotto i due campi vi saranno due bottoni che consentiranno rispettivamente di accedere all'account o di modificare la propria password. Il form verrà validato sia in maniera interattiva tramite Javascript sia nel backend con PHP.

Per chi non avesse un account vi sarà un link in nel menu che consentirà di spostarsi alla pagina di registrazione.

3.3.2 Pagina di registrazione

Valutazione LPI		Accedi
Nome	<input type="text"/>	
Cognome	<input type="text"/>	
Telefono	<input type="text"/>	
Email	<input type="text"/>	
Password	<input type="password"/>	
<input type="button" value="Registrati"/>		
Copyright SAMT 2020		

Figura 11 Mockup della pagina di registrazione

La pagina di registrazione sarà piuttosto chiara e intuitiva e conterrà unicamente un form che consentirà di inserire tutti i campi richiesti per compilare il formulario di valutazione degli LPI più una password. Il form verrà validato sia in maniera interattiva tramite Javascript sia nel backend con PHP.

3.3.3 Pagina di reset della propria password

Valutazione LPI	Registrati	Accedi
<p>Reimposta la tua password</p> <p>Email <input type="text"/></p> <p><input type="button" value="Invia email"/></p>		
Copyright SAMT 2020		

Figura 12 Mockup della pagina di reset della password

La pagina di reset della password consentirà di inserire il proprio indirizzo email alla quale verrà inviato un messaggio contenente un link, seguendo tale link l'utente si troverà in una pagina contenente due campi d'inserimento che gli consentiranno di inserire una nuova password e ripeterla.

3.3.4 Pagina contenente lo storico

Valutazione LPI	Nome	Cognome															
<table border="1"> <thead> <tr> <th>Titolo</th> <th>Ultima modifica</th> <th>Creazione</th> </tr> </thead> <tbody> <tr> <td>Esempio</td> <td>20.01.2020</td> <td>20.01.2020</td> </tr> <tr> <td>Esempio</td> <td>10.01.2020</td> <td>01.01.2020</td> </tr> <tr> <td>Esempio</td> <td>24.01.2020</td> <td>20.12.2019</td> </tr> <tr> <td>Esempio</td> <td>03.02.2020</td> <td>20.01.2020</td> </tr> </tbody> </table> <p><input type="button" value="CREA"/></p>			Titolo	Ultima modifica	Creazione	Esempio	20.01.2020	20.01.2020	Esempio	10.01.2020	01.01.2020	Esempio	24.01.2020	20.12.2019	Esempio	03.02.2020	20.01.2020
Titolo	Ultima modifica	Creazione															
Esempio	20.01.2020	20.01.2020															
Esempio	10.01.2020	01.01.2020															
Esempio	24.01.2020	20.12.2019															
Esempio	03.02.2020	20.01.2020															
Copyright SAMT 2020																	

Figura 13 Mockup della pagina contenente lo storico

La pagina contenente lo storico sarà la prima che verrà visualizzata dai docenti quando accederanno al loro account. Essa mostrerà tutti i formulari da loro creati con le date di modifica, creazione e, se è il caso, eliminazione. Il docente avrà la possibilità di eliminarli, modificarli oppure crearne dei nuovi tramite il tasto crea.

3.3.5 Pagina di aggiunta di un formulario

Valutazione LPI		Nome Cognome
Nome e cognome candidato	<input type="text"/>	
Nome e cognome superiore	<input type="text"/>	
Nome e cognome perito 1	<input type="text"/>	
Nome e cognome perito 2	<input type="text"/>	
Ricerca criteri specifici		
<input type="text" value="Codice"/>	<input type="text" value="Nome"/>	<input type="button" value="Aggiungi"/>
<input type="button" value="Crea formulario"/>		
Copyright SAMT 2020		

Figura 14 Mockup della pagina di aggiunta di un formulario

La pagina di aggiunta di un formulario sarà abbastanza complessa e carica di elementi, essa conterrà due sezioni distinte per i vari tipi di informazioni richieste:

- Dati del progetto
- Criteri specifici

Nella prima sezione si dovranno inserire tutti i dati dell'allievo coinvolto, dei periti e del superiore responsabile nel progetto che saranno: nome, cognome, email e numero di telefono. Nella seconda vi sarà invece la sezione in cui il docente potrà selezionare i criteri di valutazione specifici che potranno essere ricercati tramite codice o nome.

3.3.6 Pagina di aggiunta delle motivazioni

Valutazione LPI

Nome Cognome

Formulario del progetto: <titolo progetto>

Ricerca motivazioni

Parola chiave

Sezione

Punto

Nome	Descrizione	Punto	Sezione
Esempio	Descrizione	A1	A
Esempio	Descrizione	B3	B
Esempio	Descrizione	C4	C

Aggiungi

Valuta

Copyright SAMT 2020

Figura 15 Mockup della pagina di aggiunta delle motivazioni ad un formulario

La pagina dove sarà possibile collegare delle motivazioni ai formulari conterrà una sezione centrale che consentirà di ricercare le motivazioni tramite tre metodi diversi che saranno:

- Ricerca in base ad una parola chiave
- Ricerca in base ad una delle tre sezioni di valutazione
- Ricerca in base ad un punto di valutazione

Ciò significa che scrivendo qualcosa nel campo di testo appariranno tutte le motivazioni che contengono quella/e parola/e, inserendo una delle tre sezioni si visualizzeranno solamente le motivazioni di essa così come per un punto.

Sotto a questa sezione ve ne sarà una più piccola che conterrà i bottoni per procedere e terminare l'aggiunta, ritornare alla pagina precedente e salvare il lavoro fatto fino a quel momento.

3.3.7 Pagina di resoconto finale

Valutazione LPI	Nome Cognome
<p>Formulario del progetto: <titolo progetto></p> <p>Riassunto valutazione</p> <p>Sezione A (Comp. profess.): 40/60</p> <p>Sezione B (Documentazione): 30/30</p> <p>Sezione A (Presentazione): 20/30</p> <p>Valutazione finale: 5.0</p> <p> <input type="button" value="Indietro"/> <input type="button" value="Salva"/> </p>	
Copyright SAMT 2020	

Figura 16 Mockup della pagina di resoconto

La pagina di resoconto finale conterrà i punteggi assegnati nelle tre sezioni con la possibilità di visualizzare quali motivazioni saranno parte di ogni sezione. Sotto di esse vi sarà la nota finale del LPI insieme alla possibilità di salvare il tutto in formato PDF con lo stile deciso nell'ordinanza del 2014.

3.3.8 Pagina di amministrazione

Valutazione LPI				Nome Cognome	
Gestione utenti					
Nome	Cognome	Email	Telefono		
Luca	Rossi	l.r@mail.ch	07900000	Modifica	
Marco	Blu	m.b@mail.ch	07900000	Modifica	
<input type="button" value="Aggiungi"/>					
Gestione motivazioni					
Titolo	Descrizione	Punto	Sezione		
Esempio	Desc.	A8	A	Modifica	
Esempio1	Desc.	A7	A	Modifica	
<input type="button" value="Aggiungi"/>					
Copyright SAMT 2020					

Figura 17 Mockup della pagina di amministrazione

La pagina riservata agli amministratori sarà suddivisa in due parti relative alle due categorie di informazioni da gestire: gli utenti e le motivazioni. Nella prima sezione vi sarà infatti una tabella mostrante tutti gli utenti del sistema con la possibilità di aggiungerne, modificarli e eliminarli. Nella seconda sezione vi sarà una tabella rappresentante tutte le motivazioni con la possibilità di eseguirvi le stesse operazioni disponibili per gli utenti.

4 Implementazione

4.1 Lumen

Il progetto è stato interamente sviluppato utilizzando il micro-framework in PHP chiamato Lumen, esso è un discendente di Laravel e quindi ne riporta i tratti più caratteristici ma ha delle possibilità e componenti in meno. Esso è stato definito dai suoi sviluppatori come *“Decisamente Laravel. Deliziosamente minimale”* ed è stato valutato come uno dei micro-framework più veloci esistenti posizionandosi davanti a Silex e Slim 3.

Lumen è un framework REST, ciò significa che si basa completamente su richieste HTTP. Questo consente di generare un codice PHP unico che possa essere utilizzato da un qualsiasi client che conosca il funzionamento delle API. Questa tecnica consente di ottenere una maggiore velocità e stabilità del prodotto in quanto ogni operazione è separata e comporta una richiesta differente.

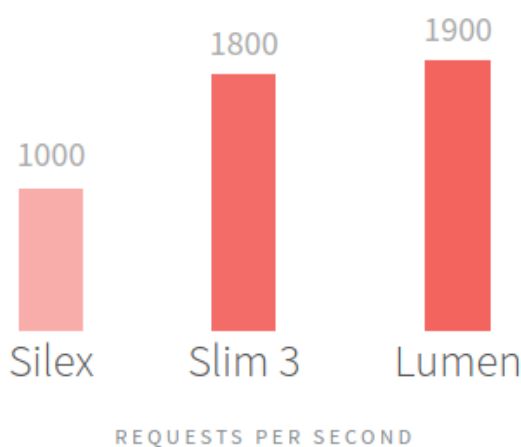


Figura 18 Lumen richieste soddisfatte per secondo

Lumen è pensato per fare delle piccole API come in questo caso e contiene già delle librerie che consentono di facilitare la scrittura del codice ad esempio `Facades` che dà la possibilità di eseguire query in maniera sicura e semplice sul database. Il vantaggio che esso ha su Laravel è che è molto più incentrato sulla velocità e quindi alcune cose sono gestite in maniera differente e facilitata per ridurre il tempo di risposta, ad esempio il routing è gestito dalla libreria `FastRoute` al posto di `Illuminate Routing`.

Il framework si basa su una struttura di lavoro del tipo MVC (`Models Views Controllers`), essi hanno i seguenti ruoli nel funzionamento dell'infrastruttura:

- **Models:** si occupano di caricare i dati dal database e di fornirli in un formato standard
- **Views:** sono le pagine web, si occupano di mostrare il risultato finale all'utente
- **Controllers:** si occupano di richiedere dei dati ai `models` e passarli alle `views`

Infine ci sono le `routes` che sono i mezzi tramite cui l'utente può eseguire le richieste al prodotto. A seguire uno schema del funzionamento di Laravel che è lo stesso di Lumen.

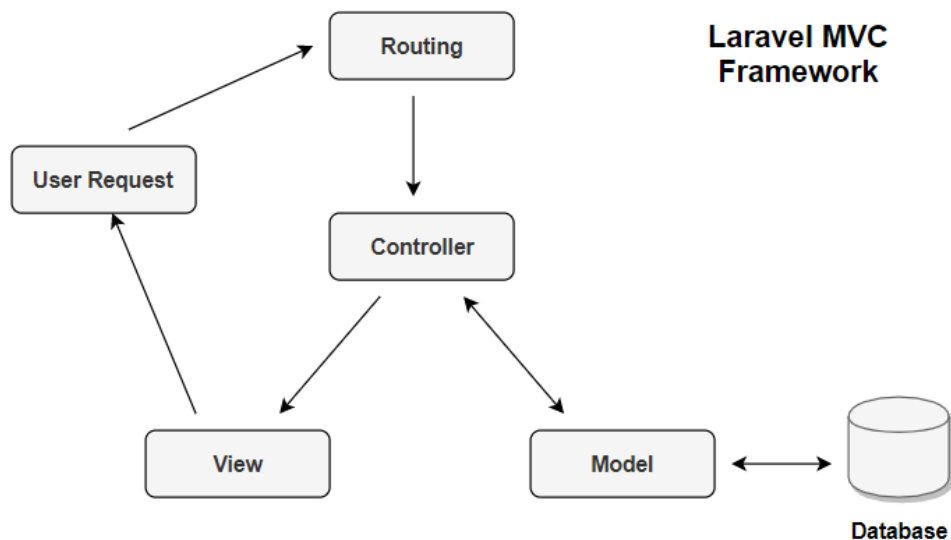


Figura 19 Schema di funzionamento MVC

4.1.1 Installazione

L'installazione di Lumen è molto veloce e semplice e si racchiude in un unico comando di composer che genera tutta la base del progetto:

```
composer create-project --prefer-dist laravel/lumen [nome_del_progetto]
```

Una volta fatto ciò basterà far partire il progetto tramite un webserver, il metodo più comodo e veloce è quello di usare il server integrato di PHP indicando quale è la base del progetto:

```
php -S localhost:8000 -t public
```

Fatto ciò il framework sarà funzionante e questo sarà visibile collegandosi al link scritto in precedenza.

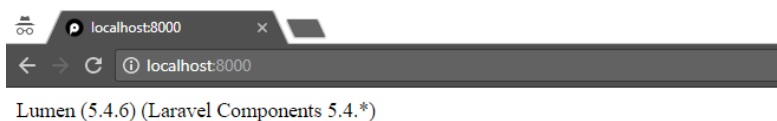


Figura 20 Lumen schermata iniziale

4.1.2 La struttura delle cartelle

Lumen alla sua installazione conta svariate cartelle, a seguire uno schema delle cartelle e files alla radice del progetto:

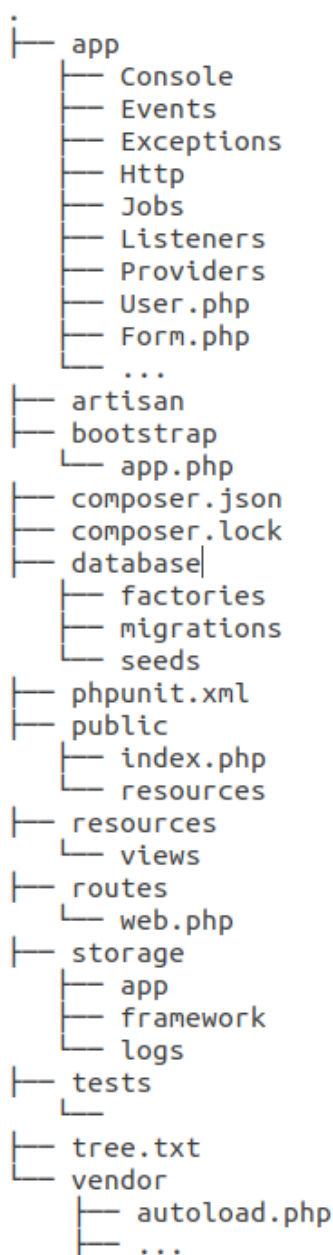


Figura 21 Schema di files e cartelle di Lumen

Le cartelle degne di nota sono quelle in cui verrà effettivamente scritto codice dallo sviluppatore, esse sono principalmente: `app` dove verranno inseriti i `models`, `Http/Controllers` dove verranno aggiunti i `controllers`, `resources/views` dove verranno salvate le `views`, `routes` che contiene il file dove andranno salvate le `routes` ed infine `Http/Middleware` dove verranno salvati i `middleware`. Un'altra cartella degna di nota è `public/resources` dove potranno venire salvate tutte le risorse esterne come librerie, estensioni, ...

4.1.3 Blade template

Oltre a ciò il micro-framework eredita da Laravel la possibilità di utilizzare i template blade, essi sono dei files contenenti codice HTML e/o Javascript che possono venire riutilizzati e richiamati per evitare la scrittura di codice già esistente. Consentono anche di eseguire delle operazioni con delle strutture comuni come gli `if` o i `for` che semplificano lo sviluppo di pagine modulari e dinamiche. Per utilizzare questo motore di templating basta creare i files delle views con l'estensione `.blade.php`.

Il vantaggio principale è l'ereditarietà che consente di richiamare una pagina in molte altre così da ridurre il codice, questo può essere utilizzato ad esempio per inserire sempre header e footer senza copiarne ed incollarne il codice in tutte le pagine.

4.1.4 Gestione delle dipendenze

Lumen si basa, per gestire le proprie dipendenze, su `Composer`, un software di gestione di dipendenze di PHP che consente di definire delle librerie da implementare nel file `composer.json`.

```
"require": {
    "php": "^7.2",
    "firebase/php-jwt": "^5.0",
    "laravel/lumen-framework": "^6.0",
    "phpmailer/phpmailer": "^6.1",
    "tecnickcom/tcpdf": "^6.3"
}
```

In questo caso si può notare come sono state implementate le librerie per utilizzare i `JWT`, il framework di base, `PHPMailer` per l'invio di email e `TCPDF` per la creazione dei PDF.

4.2 Gestione delle versioni

Le versioni del progetto sono state gestite costantemente tramite una repository di GitHub su cui venivano fatti dei push più volte al giorno. Affianco ad 'essa c'è stata, per gran parte del tempo, una repository di GitLab su un server interno della scuola sulla quale ho però smesso di eseguire dei push in quanto, dal 12.03.2020, le lezioni sono state eseguite da casa e non ho più avuto possibilità di collegarmi al server interno della scuola. Il link della cartella aggiornata costantemente è il seguente: <https://github.com/matteoforni/Valutazione-LPI>.

4.3 Database

Il database su cui si basa il progetto è stato sviluppato sulla base dello schema ER e di quello logico che si trovano rispettivamente al capitolo 3.2.1 e al capitolo 3.2.2. I punti fondamentali dello sviluppo di esso sono stati i collegamenti con le varie tabelle dato che essi implicano delle chiavi esterne che rendono necessari dei vincoli di integrità referenziale. Quest'ultimi sono stati implementati in tutte le tabelle che riportano una chiave esterna e consentono di gestire meglio i dati all'eliminazione di un record. Nella tabella `user` la chiave esterna `id_role` ha impostato che, all'eliminazione di un ruolo, nell'utente che lo possedeva, il campo viene impostato a `null` così da far continuare l'esistenza dell'utente, lo stesso comportamento avviene nella tabella `form` sul campo `id_user`.

```
foreign key(id_user) references user(id) on delete set null
```

Tutte le altre chiavi esterne hanno invece impostato che all'eliminazione del record padre esse vengono eliminate in cascata. Questo viene fatto nei casi in cui un campo non ha più senso di esistere se non ha l'attributo padre.

```
foreign key(id_form) references form(id) on delete cascade
```

L'accesso ad esso da parte del programma viene eseguito con un utente chiamato `valutazionelpi` che ha permessi solamente di scrittura, lettura, modifica e eliminazione sul database del progetto. Questo è stato fatto per evitare che, se qualcuno riuscisse ad aggirare il software, abbia pieno controllo di tutto il database.

```
create user 'valutazionelpi'@'localhost' identified by 'password';
GRANT SELECT, INSERT, UPDATE, DELETE ON valutazionelpi.* TO 'valutazionelpi'@
'localhost' ;
flush privileges;
```

4.4 Connessione con il database

Per effettuare la connessione con il database in Lumen è molto semplice e veloce, basta infatti inserire nel file `.env`, situato alla base della cartella del progetto, tutte le informazioni necessarie nei loro campi riservati.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=database_name
DB_USERNAME=username_mysql
DB_PASSWORD=password_mysql
```

Fatto ciò il framework penserà a tutto il resto e, se i dati inseriti sono corretti, la connessione verrà effettuata.

4.5 Model

Per comunicare con il database è necessario creare dei models, essi verranno salvati nella cartella `app` e conterranno poco codice dato che essi estenderanno per l'appunto la classe `Model` della libreria `Eloquent` che contiene tutti i metodi necessari. Per utilizzare `Eloquent` andrà però de commentata la riga 26 del file `bootstrap/app.php`.

```
$app->withEloquent();
```

Una volta fatto ciò basterà, nel model appena creato, estendere la classe e specificare alcuni attributi necessari al funzionamento. Bisognerà per l'appunto specificare se la tabella relativa al model utilizza i timestamps, il nome della tabella e quali colonne si potranno modificare, questo verrà fatto specificando degli attributi nella classe.

```
...
public $timestamps = false;

protected $table = 'point';

protected $fillable = [
    'code', 'title', 'description', 'type'
];
...
```

Di classi così ve ne sono otto di cui sette sono quelle relative alle sette tabelle del database del progetto mentre l'ultima è quella che gestisce l'invio delle mail di modifica della password o di notifica agli utenti. Quest'ultima è diversa dalle altre in quanto il suo compito non è quello di comunicare con il database, essa contiene un solo metodo che consente, passando il destinatario, il corpo del messaggio e il soggetto, di inviare un'email tramite il server SMTP di Gmail basandosi sulla libreria `phpmailer`.

Nella classe `Mailer` inizialmente si genera una nuova istanza di `PHPMailer` dalla quale si richiameranno tutte le funzioni necessarie per l'invio del messaggio che verrà, infine, inviato tramite il metodo `Send()`.

Gli utenti riceveranno il messaggio da parte di `valutazionelpi@gmail.com`, un indirizzo generato appositamente per il progetto.

```
$mail = new PHPMailer();
...
$mail->IsSMTP();
$mail->Host = "smtp.gmail.com";
...
$mail->Subject = $subject;
$mail->Body = $body;
$mail->AddAddress($email);

if(!$mail->Send()) {
    return false;
} else {
    return true;
}
```

4.6 Routes

La gestione delle routes, come detto in precedenza, è tutta fatta nel file `routes/web.php`. Esse rappresentano le richieste che possono essere fatte ai controllers del prodotto e, grazie al micro-framework, nel file vengono raggruppate per pagina, questo vuol dire che ogni pagina del sito avrà un suo gruppo di richieste fattibili e ben distinte.

La prima richiesta che viene effettuata, collegandosi al sito, è di tipo `get` e richiama la pagina di login.

```
$router->get('/', function () use ($router) {
    return view('login/index');
});
```

Le richieste per caricare le pagine del sito non necessitano di aver effettuato il login e sono raggruppate per un prefisso che specifica la pagina in uso come si può vedere nell'esempio seguente che mostra le routes della pagina di registrazione.

```
$router->group(['prefix' => 'register'], function () use ($router) {
    //route che mostra la pagina di registrazione chiamando URL/register
    $router->get('/', 'RegisterController@home');

    //route che richiama il metodo register quando si chiama URL/register/register
    $router->post('/register', 'RegisterController@register');
});
```

Tutti i gruppi di richieste che necessitano di privilegi speciali hanno collegato un middleware che verifica e valuta se far procedere l'utente o fermarlo. Ci sono quindi, per alcuni casi, due gruppi della stessa pagina: uno con middleware e uno senza.

```
$router->group(['middleware'=>'jwt.auth','prefix'=>'login'],function() use
($router){
    ...
});
```

4.7 Middleware

Il progetto conta un unico middleware, queste classi si occupano di filtrare gli accessi e le richieste in base, ad esempio, a delle autenticazioni. Esso si occupa di gestire le richieste che necessitano un login verificando che in esse vi sia impostato il token JWT, se esso è impostato la richiesta viene fatta passare, altrimenti viene bloccata e ritornato il codice 401 Unauthorized.

La classe riceve la richiesta http e carica il token dall'header, più precisamente dalla voce `Authorization`. Se esso è presente cerca di decodificarlo e se ci riesce manda avanti la richiesta, altrimenti ritorna un errore 401.

```
//carico il token
$token = $request->header('Authorization');
//se il token non è settato ritorno l'errore
if(!$token) {
    return response()->json(['error' => 'Token non impostato'], 401);
}
...
//cerco di decodificare il token
$credentials = JWT::decode($token, env('APP_KEY'), ['HS256']);
...
//mando avanti la richiesta
return $next($request);
```

4.8 Controllers

Le classi controller del progetto sono situate nella cartella `app/Http/Controllers` e servono a elaborare e passare i dati tra views e models e viceversa. Ogni gruppo di routes corrisponde ad un controller e quindi ve ne è uno per ogni sezione del sito: uno per il login, uno per la registrazione, uno per i docenti, uno per gli admin e uno per la creazione dei PDF.

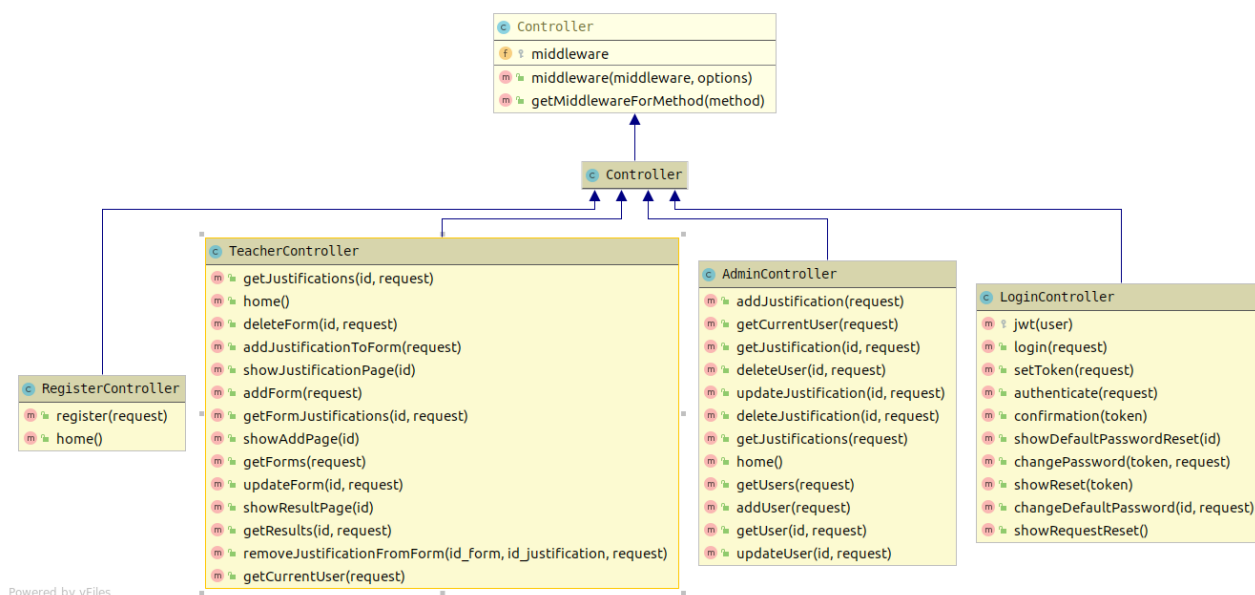


Figura 22 Schema UML della cartella Controllers

Come si può notare dallo schema soprastante tutti i controller vanno ad estendere una classe `Controller` globale che definisce una base standard per tutti gli altri. Nell'immagine non è presente il controller che si dedica a generare i PDF.

Tutti i metodi delle classi ritornano i dati in formato JSON e si aspettano anche di ottenerlo nello stesso formato.

4.8.1 Validazione dei dati

La validazione dei dati con Lumen è molto semplice ed avviene tramite la classe `Validator` presente nel framework. Essa consente, richiamando il metodo `make()` a cui va passata la richiesta, l'array di valori che dovrà contenere la richiesta con tutti i punti che devono soddisfare e i messaggi di errore, personalizzati, che verranno poi mostrati.


```
//Personalizzo i messaggi di errore.
$messages = [
  'required' => "Il campo :attribute deve essere specificato",
  'min' => "Il campo :attribute deve essere di almeno :min caratteri",
  'max' => "Il campo :attribute deve essere di massimo :max caratteri",
  'email' => "Il campo :attribute deve essere un indirizzo email valido",
  'numeric' => "Il campo :attribute deve essere di tipo numerico",
  'same' => "Il campo :attribute deve valere 0",
  'unique' => "Il campo :attribute inserito è già in utilizzo",
];

//Eseguo la validazione dei dati.
$validation = Validator::make($request->all(), [
  'name' => ['required', 'min:2', 'max:100', 'regex:/[ A-Za-zÀ-ÖØ-öø-ÿ]+/'],
  'surname' => ['required', 'min:2', 'max:100', 'regex:/[ A-Za-zÀ-ÖØ-öø-ÿ]+/'],
  'phone' => ['required', 'min:9', 'regex:/^(0|0041|\+41)?[1-9\s]{0-9\s}{1,12}$/'],
  'email' => 'required|email|unique:user',
  'password' => 'required|min:8',
], $messages);
```

Come si può notare dalla schermata superiore vi sono due metodi per specificare i requisiti che deve rispettare un campo: il primo e più utilizzato è passando un'unica stringa con tutti i punti separati da una barra verticale | mentre il secondo è passando un array di stringhe, questo si utilizza unicamente se si necessita di applicare un'espressione regolare al campo.

Per verificare che sia tutto andato bene si può utilizzare il metodo `fails()` ritornando gli errori:

```
if($validation->fails()){
  //Se fallisce ritorno gli errori.
  return response()->json($validation->errors(), '422');
}
```

La validazione di Lumen verifica autonomamente che i dati non siano tentativi di attacchi e quindi elimina autonomamente le possibilità di SQL Injection, cross-site scripting e molti altri attacchi comuni.

4.8.2 Lettura di un elemento

Per leggere un elemento dal database ci sono svariate possibilità che rispecchiano quelle presenti in MySQL esse sono tutte traducibili con un metodo di Eloquent e in totale sono

- `all`
- `find`
- `first`
- `get`
- `findOrFail`
- `firstOrFail`

La prima è l'equivalente di una query `select *` e ritorna quindi tutti i campi contenuti in una tabella del database. Se non c'è nulla nella tabella ritorna `null`.

```
$users = User::all();
```

La funzione di Eloquent `find` consente invece di ritornare solamente un campo selezionato passando come parametro del metodo il suo identificatore. Se il campo con la chiave primaria corrispondente a quella passata non esiste verrà ritornato `null`.

```
$user = User::find($id);
```

I due metodi seguenti vengono solitamente utilizzati insieme ad una clausola `where` che consente di specificare dei filtri di ricerca. La funzione `first` ritorna sempre il primo campo ottenuto come risposta dal database e se non ve ne sono ritorna `null`. Mentre `get` ritorna tutti i dati generati dalla query e mantiene lo stesso comportamento della precedente se non vi sono dati. Nell'esempio seguente la prima riga ritornerà il primo utente con il nome cercato mentre la seconda ritornerà tutti quelli che si chiamano così.

```
$user = User::where('name', $name)->first();
$users = User::where('name', $name)->get();
```

Gli ultimi due metodi della lista sono esattamente come quelli appena spiegati ma quando non trovano nessun dato non ritornano `null` bensì un codice di errore HTTP, solitamente 404.

4.8.3 Aggiunta di un elemento

L'aggiunta di un elemento al database è relativamente semplice e veloce, le caratteristiche che accomunano tutte le operazioni di questo tipo indifferentemente dalla tabella alla quale si riferiscono sono una validazione dei dati come mostrato nel capitolo precedente, se necessario l'inserimento o la rimozione di dati aggiuntivi o che non verranno salvati nel database come ad esempio la ripetizione della password e il comando di aggiunta.

Grazie al micro-framework per inserire una riga nel database basta richiamare il metodo `create()` tramite il model relativo della tabella alla quale bisogna passare la richiesta ricevuta.

```
$user = User::create($request->all());
```

4.8.3.1 Aggiunta di un utente da parte di un amministratore

L'unica aggiunta diversa dalle altre è la creazione di un utente da parte dell'amministratore, essa infatti comprende che la password non venga inserita ma creata sul momento in maniera casuale e poi spedita per email all'indirizzo dell'account.

La creazione della stringa casuale viene fatta nel seguente modo:

```
$password = bin2hex(random_bytes(5));
$request['password'] = password_hash($password, PASSWORD_BCRYPT, $options);
```

Viene appunto creata una stringa di caratteri alfanumerici lunga 10 e in seguito criptata tramite il metodo `password_hash` e inserita nella richiesta. Fatto ciò l'utente potrà essere creato e notificato via email.

```
$user = User::create($request->all());
//Invio l'email di conferma della creazione
$body = "...";
$subject = "Nuovo account di 'Valutazione LPI'";
$mailer = new Mailer();
if(!$mailer->sendMail($user->email, $body, $subject)){
    response()->json(["Bad Gateway" => "Impossibile inviare l'email"], 502);
}

//Ritorno la risposta di successo.
return response()->json($user, 201);
```

4.8.4 Modifica di un elemento

La modifica di una riga del database comporta la sovrascrittura di essa e può essere fatta in due modi diversi.

4.8.4.1 Modifica totale

Con modifica totale si intende quando si vanno a sovrascrivere tutti i dati contenuti nel record attuale, anche se quelli nuovi sono uguali a quelli vecchi. Questo metodo viene utilizzato nel progetto quando vi sono dei form di modifica che consentono di cambiare tutti i dati, così da evitare inutili verifiche su quale campo abbia subito un cambiamento e quale no.

Per fare ciò con Lumen basta inviare nella richiesta tutti i dati, come se si volesse aggiungere il campo, validarli ed infine passarli al metodo `update` del model di riferimento della tabella.

```
$user->update($request->all());
```

4.8.4.2 Modifica parziale

Con modifica parziale si intende quando si va a modificare solamente un campo della riga salvata nel database, questo è utile se si sa che si sta modificando unicamente un valore ed evita quindi la sovrascrittura di tutti gli altri.

Per eseguire questa operazione in Lumen si deve solamente caricare il record dal database con uno dei metodi disponibili e modificare unicamente il campo scelto.

```
$user = User::find($id);
$user->name = $name;
$user->surname = $surname;
```

Infine bisognerà sempre ricordarsi di salvare per rendere effettive le modifiche.

```
$user->save();
```

4.8.5 Eliminazione

Per eliminare un record del database si dovrà seguire una procedura simile a quelle già mostrate in precedenza. Inizialmente va cercato il campo da eliminare tramite uno dei metodi di lettura dopodiché andrà richiamato il metodo `delete()` e il record verrà cancellato, se questo non esiste verrà ritornato errore.

```
Form::findOrFail($id)->delete();
```

4.8.6 Query strutturate

L'ORM utilizzato dal micro-framework, Eloquent, consente di eseguire gran parte delle richieste ad un database però ha dei difetti quando si fanno delle query complesse. Esse infatti sono complicate da creare in quanto la posizione di chiamata dei metodi influenza il funzionamento della richiesta.

Con Eloquent si possono per l'appunto fare richieste contenenti `union`, `join`, ... ma esse richiedono la chiamata di svariate funzioni che se non sono nell'ordine esatto non fanno ciò che viene richiesto. Inoltre richiedono più tempo per scriverle in quanto vanno prima pensate e sviluppate in SQL e poi tradotte in codice PHP.

Nel progetto per risparmiare tempo in queste situazioni è stato quindi utilizzato un secondo metodo messo a disposizione da Lumen che è il `QueryBuilder`, esso consente infatti di passare la richiesta in codice SQL e si occupa di filtrare i dati, passati come parametri, e ritornare il risultato nello stesso formato di Eloquent. A seguire la query che ritorna tutte le motivazioni di una sezione di valutazione.

```
$query = "select * from contains inner join justification on
contains.id_justification = justification.id where justification.id_point like 'A%'
and contains.id_form = " . $id;
$resultA = DB::select($query);
```

Una query come quella mostrata sopra sarebbe abbastanza complessa e dispendiosa di tempo da sviluppare e da testare tramite l'ORM.

4.8.7 Creazione dei PDF

Per creare il file PDF di valutazione è stata utilizzata la libreria TCPDF che è stata installata tramite composer. Essa consente di sviluppare dei file abbastanza complessi senza troppe difficoltà dato che gli si può passare il codice HTML e lo traduce autonomamente in PDF.

Anche essa non è però perfetta e in molti casi riporta dei problemi che sono da risolvere in maniera poco pulita, questo giustifica il fatto che ad esempio lo stile di certi elementi va bene esterno mentre in altri lo accetta solo inline.

Tutto il codice relativo ai PDF si trova nel file `app/Http/Controllers/PDFController.php`. Esso contiene la classe `PDFController` che estende `TCPDF`, essa va a richiamare i metodi della superclasse per generare l'intestazione e il piè di pagina.

```
/**
 * Funzione che consente di generare l'header del PDF
 */
public function Header() {
    //Mi posiziono a 9 mm dall'inizio del documento
    $this->SetY(9);
    //Imposto il font
    $this->SetFont('courier', 'I', 9.5);
    //Genero l'header
    $this->Cell(0, 9.5, 'Procedura di qualificazione: Informatica/o AFC (Ordinanza
2014) Griglia di valutazione', 0, false, 'C', 0, '', 0, false, 'M', 'M');
}

/**
 * Funzione che consente di generare il footer del PDF
 */
public function Footer() {
    //Mi posiziono a 15 mm dalla fine del documento
    $this->SetY(-15);
    //Imposto il font
    $this->SetFont('courier', 'I', 10);
    //Genero il footer
    $this->Cell(30, 10, 'Pagina '.$this->getAliasNumPage().'/'.$this-
>getAliasNbPages(), 0, false, 'C', 0, '', 0, false, 'T', 'M');
    $this->Cell(215, 10, 'Versione 2.0 (310119) - Ordinanza 2014', 0, false, 'C',
0, '', 0, false, 'T', 'M');
}
```

Dopodiché contiene tutti i metodi che vanno a caricare i dati che dovranno apparire all'interno del file. Il più importante è quello che va a caricare il formulario, prendendo l'id dall'URL passato, perché tutti gli altri si baseranno sui dati ritornati da lui.

```
public function getForm() {
    $parts = explode("/", $_SERVER['REQUEST_URI']);
    $this->form = Form::find(end($parts));
}
```

Dopodiché verranno caricati i dati di tutte le persone coinvolte nel LPI: lo studente, il docente e i due periti. I quattro metodi sono praticamente uguali, caricano semplicemente gli stessi dati di persone diverse.

```
public function getStudent() {
    $student = (object) [
        "name" => $this->form['student_name'] . " " . $this->form['student_surname'],
        "phone" => $this->form['student_phone'],
        "email" => $this->form['student_email']
    ];
    return $student;
}
```

Dopodiché verranno caricate le motivazioni collegate al formulario e calcolati i punti per ogni sezione (A, B e C).

La creazione del file inizierà istanziando un nuovo oggetto di tipo `PDFController` passandogli l'orientamento, l'unità di misura del file, altri dati necessari e poi impostandogli i margini delle pagine.

```
$pdf = new PDFController(PDF_PAGE_ORIENTATION, PDF_UNIT, PDF_PAGE_FORMAT, true,
'UTF-8', false);
//Imposto i margini
$pdf->SetMargins("20", "15", "20");
```

Poi viene richiamata la funzione che carica il formulario, se essa ritorna `null` non viene caricato alcun dato e mostrato il PDF vuoto altrimenti vengono richiamate tutte le funzioni mostrate in precedenza.

Una volta ottenuti, se necessari, tutti i dati si passa alla creazione delle pagine. La struttura sarà sempre la stessa per ogni pagina: si aggiungerà la pagina, si imposterà la distanza dall'inizio da mantenere, si genererà il contenuto HTML e lo si aggiungerà al PDF.

```
$pdf->AddPage('P', 'A4');
$pdf->SetY(10);

//Genero la prima pagina
$content = <<<EOF
...
<table border="0.5" cellspacing="0" cellpadding="2">
    <tr>
        <th><b> Superiore professionale</b></th>
        <th><b> Candidata/o</b></th>
    </tr>
    ...
</table>
...
EOF;
//Scrivo nel file la pagina
$pdf->writeHTML($content);
```

I dati verranno inseriti direttamente all'interno della stringa dato che le stringhe multilinea definite con `<<<EOF` lo consentono.

```
<tr>
    <td>$teacher->name</td>
    <td>$student->name</td>
</tr>
```

Dopo aver generato tutte le pagine verrà richiamato il metodo che indica il termine del file e verrà mostrato tramite il PDF Viewer del browser il file compilato.

```
//Chiudo il file
$pdf->lastPage();

//Notifico al browser che invio un PDF
header("Content-Type: application/pdf");
//Stampo il PDF e chiudo l'output
$pdf->Output($title, "I");
exit();
```

Il file terminato apparirà molto simile al file originale creato dal cantone ma non esattamente uguale in quanto alcuni particolari non sono fattibili tramite la libreria utilizzata e dato che è stato ricreato ad occhio cercando di imitare al meglio i colori e le posizioni dei contenuti.

Procedura di qualificazione: Informatica/o AFC (Ordinanza 2014) Griglia di valutazione

1 INFORMAZIONI GENERALI

Superiore professionale	Candidata/o
Massimo Sartori	Matteo Forni
Telefono	Telefono
1239120391	0799119368
Email	Email
massimo.sartori@edu.ti.ch	matteo.forni@samtrevano.ch

Perito 1	Perito 2
Perito Primo	Perito Secondo
Telefono	Telefono
10293121212	10293912312
Email	Email
primo.perito@edu.ti.ch	secondo.perito@edu.ti.ch

2 PROCEDURA

Il seguente documento non deve assolutamente essere mostrato al candidato una volta attribuiti i punti.

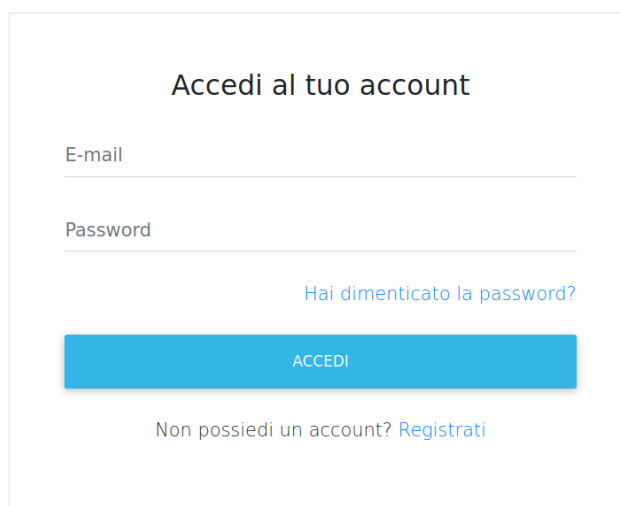
Documentazione

I periti d'esame trattano la documentazione in maniera confidenziale. La conservazione della documentazione è regolata dalla legge cantonale.

Figura 23 File PDF generato automaticamente

4.9 Views

4.9.1 Pagina di login

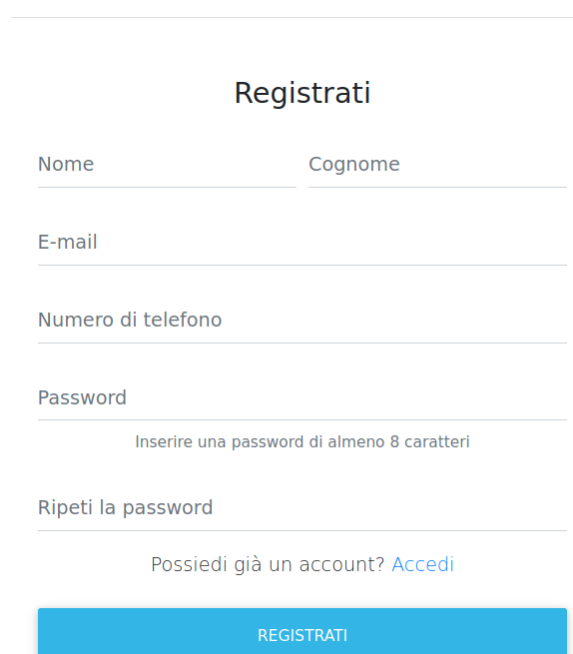


The login page features a light gray background with a white central box. At the top, the title 'Accedi al tuo account' is centered. Below it are two input fields: 'E-mail' and 'Password'. A blue link 'Hai dimenticato la password?' is positioned to the right of the password field. A prominent blue button labeled 'ACCEDI' is centered below the fields. At the bottom, a text prompt 'Non possiedi un account?' is followed by a blue link 'Registrati'.

Figura 24 Pagina di login

Questa è la pagina di login che consente agli utenti di accedere al proprio account, se sono dei docenti verranno mandati alla pagina dello storico (4.9.4) altrimenti, se sono admin, a quella di amministrazione. Premendo su “*Hai dimenticato la password*” si verrà inviati alla pagina che consente di modificarla.

4.9.2 Pagina di registrazione



The registration page has a light gray background with a white central box. The title 'Registrati' is centered at the top. Below it are several input fields: 'Nome' and 'Cognome' (split), 'E-mail', 'Numero di telefono', 'Password', and 'Ripeti la password'. A text hint 'Inserire una password di almeno 8 caratteri' is shown below the password field. A blue link 'Possiedi già un account? Accedi' is placed below the repeat password field. A large blue button labeled 'REGISTRATI' is centered at the bottom.

Figura 25 Pagina di registrazione

La pagina di registrazione contiene unicamente un form di inserimento dei propri dati, una volta compilati tutti i campi, se i dati saranno stati inseriti correttamente, l'account verrà creato e l'utente rimandato al login.

4.9.3 Pagina di amministrazione

Gestione motivazioni

Cerca motivazioni

Id	Text	Id point	Modifica	Elimina
1	Alcune attività del Gantt non rispettate	A1		
2	Il Gantt non mostra le ore di lavoro	A1		
3	Descritto con poca precisione e sommariamente	B1		
4	Diario/i consegnato/i in ritardo	B2		
6	Errori con il funzionamento di Powerpoint	C2		

[Pagina precedente](#)
1
[2](#)
[3](#)
[4](#)
[Prossima pagina](#)

AGGIUNGI MOTIVAZIONE

Figura 26 Pagina di amministrazione, sezione motivazioni

La pagina di amministrazione è suddivisa in due sezioni, la prima consente di gestire le motivazioni contenute nel database. Esse potranno essere visualizzate in una tabella con la possibilità di modificarle, tramite un modale, o eliminarle. Se si vorrà inserire una nuova motivazione si potrà premere sul bottone sotto alla tabella che aprirà un modale di aggiunta.

Gestione utenti

Cerca utenti

Id	Nome	Cognome	Email	Telefono	Ruolo	Modifica	Elimina
1	Matteo	Forni	matteo.forni@samtrevano.ch	0799119368	admin		
2	Matteo	Forni	matte20012001@gmail.com	76567865456	teacher		
9	Matteo	Forni	matteoforni@hotmail.com	123534522334	teacher		

[Pagina precedente](#)
1
[Prossima pagina](#)

AGGIUNGI UTENTE

Figura 27 Pagina di amministrazione, sezione utenti

La seconda sezione conterrà una tabella che mostrerà gli utenti salvati nel database con, come in precedenza, la possibilità di eliminarli o modificarli tramite un modale. Sarà inoltre possibile aggiungerne di nuovi tramite il modale che verrà aperto premendo il bottone di aggiunta.

4.9.4 Pagina dello storico

Storico dei formulari

Search:

Id	Titolo	Creato	Email allievo	Email docente	Motivazioni	Risultati	Modifica	Elimina
7	Valutazione LPI	2020-03-11	matteo.forni@samtrevano.ch	massimo.sartori@edu.ti.ch	+	📄	✎	🗑
8	Gestione acquario	2020-03-11	mattia.ruberto@samtrevano.ch	luca.peduzzi@edu.ti.ch	+	📄	✎	🗑

Pagina precedente **1** Prossima pagina

AGGIUNGI FORMULARIO

VISUALIZZA TEMPLATE

Figura 28 Pagina dello storico

La pagina dello storico è la prima che visualizzano i docenti dopo essersi connessi al loro account, essa contiene una tabella che mostra tutti i formulari creati da un docente sia che siano i suoi sia che li abbia creati per un altro docente. Tramite la tabella avrà la possibilità di aggiungere motivazioni al formulario, visualizzarne i risultati, modificarlo o eliminarlo.

Sotto la tabella avrà due bottoni che consentono rispettivamente di aggiungere un formulario o di vedere il template del PDF.

4.9.5 Pagina di aggiunta

Aggiungi un formulario

Titolo del progetto

Informazioni sullo studente *

Nome dello studente

Cognome dello studente

E-mail dello studente

Numero di telefono dello studente

Informazioni sul docente responsabile *

Nome del docente

Cognome del docente

E-mail del docente

Numero di telefono del docente

Figura 29 Pagina di aggiunta

La pagina di aggiunta consente ad un docente, per l'appunto, di aggiungere dei formulari di valutazione. Essa conterrà unicamente un form di inserimento dove potrà immettere tutti i dati necessari: il titolo del lavoro e le generalità dello studente, del docente responsabile e dei due periti. Tutti i campi relativi ai periti sono opzionali in quanto il progetto valutato potrebbe essere uno di preparazione all'esame e quindi senza periti. Dopo tutti i dati delle persone coinvolte vi è una sezione dedicata a scegliere i punti specifici di valutazione.

La pagina viene utilizzata in due modalità: come aggiunta di formulari o come modifica di un formulario. Infatti essa ha due routes con cui è possibile richiamarla: la prima senza nessun parametro mentre la seconda, per eseguire una modifica, con l'identificatore del formulario da caricare. Questo viene fatto tramite un percorso opzionale di Lumen, se si mette una parte di URL tra parentesi quadre essa diventa opzionale.

```
$router->get('/form/show/add[/]{id}', 'TeacherController@showAddPage');
```

4.9.6 Pagina di aggiunta delle motivazioni

Aggiunta di motivazioni al formulario

Cerca motivazioni

Id	Text	Id point
11	Nessuna descrizione reale di test	A1
16	Manca la parte di condivisione sui social	A1
10	Qualche descrizione in più per il GANTT (aggiuntive)	A2
13	In qualche occasione il candidato ha chiesto aiuto per la risoluzione di alcune problematiche	A2
17	Alcune variabili e commenti sono in inglese / italiano	A2

Pagina precedente 1 **2** 3 4 Prossima pagina

Motivazioni presenti

ID	Titolo
1	Alcune attività del Gantt non rispettate
2	Il Gantt non mostra le ore di lavoro
6	Errori con il funzionamento di Powerpoint
7	Problema con il punto specifico 135
10	Qualche descrizione in più per il GANTT (aggiuntive)

Pagina cedente **1** 2 3 Prossima pagina

Figura 30 Pagina di aggiunta delle motivazioni

La pagina di aggiunta delle motivazioni ad un formulario contiene due tabelle: la prima, sulla sinistra, dove sono presenti tutte le motivazioni dei punti di base più quelle dei punti specifici selezionati per il formulario e la seconda dove vi sono quelle che sono già state collegate con il formulario.

Nella prima tabella le motivazioni già utilizzate diventano disabilitate, come si può vedere nell'immagine, finché esse non verranno rimosse dalla tabella di destra. Ad un formulario si potranno aggiungere al massimo tre motivazioni per ogni punto di valutazione.

4.9.7 Pagina dei risultati




Figura 31 Pagina dei risultati

La pagina dei risultati mostra quattro note differenti: le tre valutazioni delle sezioni e quella globale. Per ogni sezione si potranno visualizzare le motivazioni collegate grazie ad un tasto apposito posto sotto i risultati. Al centro della pagina vi è un bottone che consente di aprire, in un'altra pagina, il PDF compilato e pronto per essere stampato o scaricato, la gestione del file online è lasciata al viewer utilizzato dal browser.


5 Test

5.1 Protocollo di test

Test Case:	TC-01	Nome:	Verifica del funzionamento della connessione con il database
Riferimento:	REQ-01		
Descrizione:	Controllare che il sito riesca a collegarsi al database creato e che non ritorni errori		
Prerequisiti:	<ul style="list-style-type: none"> Aver creato il database in MySQL Aver scritto i dati di connessione corretti nel file .env 		
Procedura:	<ol style="list-style-type: none"> Aprire un browser alla pagina principale del sito Verificare che la pagina si apra senza mostrare messaggi di errore 		
Risultati attesi:	Dovrebbe aprirsi la pagina di login senza che vengano mostrati messaggi di errore a schermo.		

Test Case:	TC-02	Nome:	Verifica del funzionamento del form di registrazione e dell'invio dell'email di conferma
Riferimento:	REQ-02		
Descrizione:	Controllare che la registrazione di un utente vada a buon fine e che venga inviata l'email di conferma		
Prerequisiti:	-		
Procedura:	<ol style="list-style-type: none"> 1. Aprire un browser alla pagina principale del sito 2. Spostarsi alla pagina di registrazione tramite il menu o il link nel form di login 3. Riempire il form con dati validi (es.: "Matteo", "Forni", "email@esempio.com", "+41 79 111 222 33", "Password&1", "Password&1") 4. Premere il tasto "Registrati" 5. Recarsi nell'inbox dell'email inserita durante la registrazione e verificare che sia arrivato un messaggio da "valutazione.lpi@gmail.com" contenente un link per la conferma dell'indirizzo 		
Risultati attesi:	<p>Si dovrebbe ottenere un messaggio di errore se i dati inseriti non sono validi mentre se tutto è corretto si dovrebbe venire reindirizzati alla pagina di login. Nella inbox dell'email utilizzata nella registrazione dovrebbe essere arrivato il seguente messaggio:</p> <div style="text-align: center;">  valutazione.lpi@gmail.com a me ▼ </div> <p>Conferma la tua email</p> <p>Hai creato con successo il tuo account di 'Valutazione LPI', conferma la tua email per completare l'operazione Premi qui per confermare l'email</p>		

Test Case:	TC-03	Nome:	Verifica dei controlli e dei messaggi di errore del form di registrazione
Riferimento:	REQ-02		
Descrizione:	Bisognerà verificare che nella pagina di registrazione i controlli funzionino correttamente, quindi che non si possano registrare più utenti con la stessa email, che non si possa registrare un utente con dei numeri nel nome, ecc. Dovranno uscire dei messaggi di errore in caso l'utente inserisca dei dati errati.		
Prerequisiti:	<ul style="list-style-type: none"> • Deve esserci un utente, di cui si conosce l'email, già presente nel database 		
Procedura:	<ol style="list-style-type: none"> 1. Aprire un browser alla pagina principale del sito 2. Spostarsi alla pagina di registrazione tramite il menu o il link nel form di login 3. Riempire il form con dati non validi o duplicati, nel caso dell'email 4. Premere il tasto "Registrati" 5. Ripetere i punti 3 e 4 con diversi campi sbagliati (es. numero di telefono con lettere, email senza "@", nome con numeri, password che non coincidono ed eventualmente campi vuoti). 		
Risultati attesi:	Si dovrebbero vedere dei messaggi di errore differenti a dipendenza di cosa si ha inserito di invalido e la registrazione non dovrà venire effettuata.		

Test Case:	TC-04	Nome:	Verifica del funzionamento del form di login e dell'link di conferma della email
Riferimento:	REQ-03		
Descrizione:	Controllare che il login di un utente funzioni correttamente e che il link ricevuto alla registrazione confermi l'account		
Prerequisiti:	<ul style="list-style-type: none"> Aver creato l'account tramite la pagina di registrazione Aver ricevuto l'email di conferma 		
Procedura:	<ol style="list-style-type: none"> 1. Aprire un browser alla pagina principale del sito 2. Inserire nel form i dati di accesso (email e password) dell'account corretti 3. Controllare che, dopo l'accesso, appaia una notifica che ricorda di confermare l'email  <ol style="list-style-type: none"> 4. Aprire l'inbox e più precisamente l'email di conferma dell'account 5. Premere il link 6. Effettuare nuovamente il login 		
Risultati attesi:	Si dovrebbe ottenere un messaggio di errore se i dati inseriti non sono validi mentre se tutto è corretto si dovrebbe venire reindirizzati alla pagina contenente lo storico dei formulari. Accedendo la prima volta dovrebbe apparire la notifica, dopo aver premuto il link essa non dovrebbe più apparire e l'account nel database dovrebbe avere l'attributo dell'utente <code>confirmed</code> impostato a 1.		

Test Case:	TC-05	Nome:	Verifica del funzionamento della sezione delle motivazioni nella pagina admin
Riferimento:	REQ-04		
Descrizione:	Controllare che la sezione dedicata alle motivazioni funzioni correttamente e che consenta di eseguire tutte le operazioni necessarie su di esse		
Prerequisiti:	<ul style="list-style-type: none"> Avere un account valido e con privilegi di amministratore Ci devono essere delle motivazioni nel database 		
Procedura:	<ol style="list-style-type: none"> 1. Aprire un browser alla pagina principale del sito 2. Inserire nel form i dati di accesso (email e password) dell'account corretti 3. Inserire delle parole chiavi nel campo di testo sopra la tabella 4. Eliminare una motivazione tramite il tasto nella tabella 5. Modificare una motivazione tramite il tasto nella tabella 6. Inserire dati incorretti (testo uguale ad una già presente) 7. Premere il bottone di modifica 8. Inserire dati corretti 9. Premere il bottone di modifica 10. Premere il bottone di aggiunta "Aggiungi motivazione" 11. Inserire dati incorretti (testo uguale ad una già presente) 12. Premere il tasto "Aggiungi motivazione" 13. Creare una nuova motivazione con dati corretti 14. Premere il tasto "Aggiungi motivazione" 		
Risultati attesi:	Inserendo parole o lettere nell'input soprastante la tabella i dati mostrati in essa dovranno essere filtrati. Dovrebbero venire mostrati unicamente quelli che contengono la parola. All'eliminazione dovrebbe apparire, in alto a destra, una notifica di conferma se tutto è andato		

	<p>a buon fine o di errore se non è stato possibile eliminare il campo.</p> <p>Nella modifica si dovrebbe ottenere un messaggio di errore se i dati inseriti non sono validi mentre se tutto è corretto si dovrebbe visualizzare il cambiamento nella tabella.</p> <p>Creando una nuova motivazione dovrebbero apparire, se sia hanno inserito dati errati, dei messaggi di errore altrimenti si dovrebbe vedere l'aggiunta presente nella tabella.</p>
--	---

Test Case:	TC-06	Nome:	Verifica del funzionamento della sezione degli utenti nella pagina admin
Riferimento:	REQ-04		
Descrizione:	Controllare che la sezione dedicata agli utenti funzioni correttamente e che consenta di eseguire tutte le operazioni necessarie su di essi		
Prerequisiti:	<ul style="list-style-type: none"> Avere un account valido e con privilegi di amministratore Ci devono essere degli utenti nel database 		
Procedura:	<ol style="list-style-type: none"> 1. Aprire un browser alla pagina principale del sito 2. Inserire nel form i dati di accesso (email e password) dell'account corretti 3. Inserire delle parole chiavi nel campo di testo sopra la tabella 4. Eliminare un utente tramite il tasto nella tabella 5. Modificare un utente tramite il tasto nella tabella 6. Inserire dati incorretti (controlli uguali alla registrazione) 7. Premere il bottone di modifica 8. Inserire dati corretti 9. Premere il bottone di modifica 10. Verificare l'inbox dell'utente 11. Premere il bottone di aggiunta "Aggiungi utente" 12. Inserire dati incorretti (testo uguale ad una già presente) 13. Premere il tasto "Registra l'utente" 14. Creare un nuovo utente con dati corretti 15. Premere il tasto "Registra l'utente" 16. Verificare l'inbox dell'utente 		
Risultati attesi:	<p>Inserendo parole o lettere nell'input soprastante la tabella i dati mostrati in essa dovranno essere filtrati. Dovrebbero venire mostrati unicamente quelli che contengono la parola.</p> <p>All'eliminazione dovrebbe apparire, in alto a destra, una notifica di conferma se tutto è andato a buon fine o di errore se non è stato possibile eliminare il campo.</p> <p>Nella modifica si dovrebbe ottenere un messaggio di errore se i dati inseriti non sono validi mentre se tutto è corretto si dovrebbe visualizzare il cambiamento nella tabella. Inoltre l'utente dovrebbe ricevere un'email con tutti i dati aggiornati</p> <p>Creando una nuovo utente dovrebbero apparire, se sia hanno inserito dati errati, dei messaggi di errore altrimenti si dovrebbe vedere l'aggiunta presente nella tabella ed esso dovrebbe ricevere un'email contenente la sua password da usare al primo login.</p>		

Test Case:	TC-07	Nome:	Verifica del funzionamento della pagina iniziale dei docenti
Riferimento:	REQ-05		
Descrizione:	Controllare che la prima pagina visualizzata dai docenti dopo il login, che mostra lo storico, funzioni correttamente		
Prerequisiti:	<ul style="list-style-type: none"> Avere un account valido e con privilegi di docente Ci devono essere dei formulari collegati ad esso 		
Procedura:	<ol style="list-style-type: none"> 1. Aprire un browser alla pagina principale del sito 2. Inserire nel form i dati di accesso (email e password) dell'account corretti 3. Inserire delle parole chiavi nel campo di testo sopra la tabella 4. Eliminare un formulario tramite il tasto nella tabella 5. Premere sul tasto della tabella "Modifica" 6. Tornare alla pagina iniziale tramite il menu 7. Premere sul tasto della tabella "Motivazioni" 8. Tornare alla pagina iniziale tramite il menu 9. Premere sul tasto della tabella "Risultati" 10. Tornare alla pagina iniziale tramite il menu 11. Premere il bottone "visualizza template" 12. Premere il bottone "aggiungi formulario" 		
Risultati attesi:	<p>Inserendo parole o lettere nell'input soprastante la tabella i dati mostrati in essa dovranno essere filtrati. Dovrebbero venire mostrati unicamente quelli che contengono la parola.</p> <p>All'eliminazione dovrebbe apparire, in alto a destra, una notifica di conferma se tutto è andato a buon fine o di errore se non è stato possibile eliminare il campo.</p> <p>Premendo i tasti nella tabella "Motivazioni", "Risultati" e "Modifica" si dovrebbe venire reindirizzati alla pagina relativa: la pagina di aggiunta delle motivazioni al formulario, la pagina che mostra i risultati e quella che consente di modificare i dati del formulario.</p> <p>Il bottone "visualizza template" dovrebbe aprire in una nuova pagina il template di base su cui verranno basati i PDF di valutazione mentre tramite il tasto "aggiungi formulario" si dovrebbe venire reindirizzati alla pagina di aggiunta.</p>		

Test Case:	TC-08	Nome:	Verifica del funzionamento della pagina di aggiunta
Riferimento:	REQ-06		
Descrizione:	Controllare che la pagina di aggiunta di un formulario da parte di un docente funzioni correttamente		
Prerequisiti:	<ul style="list-style-type: none"> Avere un account valido e con privilegi di docente Ci dovranno essere almeno sette punti specifici nel database 		
Procedura:	<ol style="list-style-type: none"> 1. Aprire un browser alla pagina principale del sito 2. Inserire nel form i dati di accesso (email e password) dell'account corretti 3. Premere il bottone "aggiungi formulario" 4. Riempire il form con dati validi (es.: "Valutazione di formulari LPI", "Matteo", "Forni", "email@esempio.com", "+41 79 111 222 33", ...) 5. Scegliere sette punti specifici 6. Premere il tasto "Aggiungi" 		
Risultati attesi:	Il bottone "aggiungi formulario" dovrebbe mostrare la pagina di aggiunta, inserendo i dati nel form si dovrebbe ottenere un messaggio di errore, se ne sono stati inseriti di non validi,		

mentre se tutto è corretto si dovrebbe venire reindirizzati alla pagina di aggiunta delle motivazioni ad un formulario.

Test Case:	TC-09	Nome:	Verifica dei controlli e dei messaggi di errore del form di aggiunta
Riferimento:	REQ-06		
Descrizione:	Bisognerà verificare che nella pagina di aggiunta di un formulario i controlli funzionino correttamente, quindi che non si possano che non si possa inserire un utente con dei numeri nel nome, che i numeri di telefono siano corretti ecc. Dovranno uscire dei messaggi di errore in caso l'utente inserisca dei dati errati.		
Prerequisiti:	<ul style="list-style-type: none"> Avere un account valido e con privilegi di docente 		
Procedura:	<ol style="list-style-type: none"> 1. Aprire un browser alla pagina principale del sito 2. Inserire nel form i dati di accesso (email e password) dell'account corretti 3. Spostarsi nella pagina di aggiunta 4. Riempire il form con dati non validi 5. Premere il tasto "Aggiungi" 6. Ripetere i punti 4 e 5 con diversi campi sbagliati (es. numero di telefono con lettere, email senza "@", nome con numeri ed eventualmente campi vuoti). 		
Risultati attesi:	Si dovrebbero vedere dei messaggi di errore differenti a dipendenza di cosa si ha inserito di invalido e l'aggiunta non dovrà venire effettuata.		

Test Case:	TC-10	Nome:	Verifica del funzionamento dell'aggiunta delle motivazioni ad un formulario
Riferimento:	REQ-06		
Descrizione:	Controllare che la pagina che consente di aggiungere motivazioni ad un formulario funzioni correttamente		
Prerequisiti:	<ul style="list-style-type: none"> Avere un account valido e con privilegi di docente Ci devono essere delle motivazioni nel database 		
Procedura:	<ol style="list-style-type: none"> 1. Aprire un browser alla pagina principale del sito 2. Inserire nel form i dati di accesso (email e password) dell'account corretti 3. Premere il tasto "Motivazioni" nella tabella oppure creare correttamente un nuovo formulario 4. Inserire delle parole chiavi nel campo di testo sopra la tabella 5. Aggiungere una motivazione premendovi sopra 6. Rimuovere una motivazione dalla tabella "Motivazioni presenti" premendoci sopra 7. Premere il tasto "Risultati" 		
Risultati attesi:	<p>Inserendo parole o lettere nell'input soprastante la tabella i dati mostrati in essa dovranno essere filtrati. Dovrebbero venire mostrati unicamente quelli che contengono la parola.</p> <p>Alla pressione su di una motivazione nella tabella "Aggiunta di motivazioni al formulario" dovrebbe apparire un modale che chiederà conferma dell'aggiunta e, confermando l'operazione, nella tabella di sinistra essa dovrebbe diventare disabilitata e dovrebbe apparire in quella di destra.</p> <p>Alla pressione su di una motivazione nella tabella "Motivazioni presenti" dovrebbe apparire un modale che chiederà conferma della rimozione e, confermando l'operazione, essa dovrebbe venire tolta dalla tabella di destra e riabilitata in quella di sinistra.</p> <p>Premendo sul tasto "Risultati" si dovrebbe aprire la pagina dei risultati.</p>		

Test Case:	TC-11	Nome:	Verifica del calcolo dei risultati
Riferimento:	REQ-07		
Descrizione:	Controllare che il calcolo dei risultati funzioni correttamente		
Prerequisiti:	<ul style="list-style-type: none"> Avere un account valido e con privilegi di docente Ci devono essere delle motivazioni nel database 		
Procedura:	<ol style="list-style-type: none"> Aprire un browser alla pagina principale del sito Inserire nel form i dati di accesso (email e password) dell'account corretti Premere il tasto "Motivazioni" nella tabella oppure creare correttamente un nuovo formulario Aggiungere delle motivazioni della sezione A premendovi sopra Aggiungere delle motivazioni della sezione B premendovi sopra Aggiungere delle motivazioni della sezione C premendovi sopra Premere il tasto "Risultati" Premere, per ogni sezione, "mostra motivazioni" 		
Risultati attesi:	<p>Aprendo la pagina dei risultati e selezionando "Mostra motivazioni" dovrebbero venire mostrate tutte le motivazioni inserite per la sezione selezionata. Le valutazioni delle sezioni mostrate sopra dovrebbero avere una nota calcolata con la seguente formula:</p> $(\text{punti totali} - \text{numero motivazioni aggiunte}) / \text{punti totali} * 5 + 1$ <p>Mentre la nota globale dovrebbe rispecchiare la seguente formula:</p> $\text{nota sezione A} * 0.5 + \text{nota sezione B} * 0.25 + \text{nota sezione C} * 0.25$		

Test Case:	TC-12	Nome:	Verifica creazione PDF
Riferimento:	REQ-08		
Descrizione:	Controllare che il PDF contenga i dati corretti nella posizione corretta		
Prerequisiti:	<ul style="list-style-type: none"> Avere un account valido e con privilegi di docente Deve esistere il formulario 		
Procedura:	<ol style="list-style-type: none"> Aprire un browser alla pagina principale del sito Inserire nel form i dati di accesso (email e password) dell'account corretti Premere il tasto "Risultati" nella tabella Premere il tasto "Visualizza PDF" 		
Risultati attesi:	<p>Premendo sul tasto "Visualizza PDF" si dovrebbe aprire, in una nuova pagina, il PDF di valutazione del progetto LPI. In esso dovrebbero venire riportati correttamente tutti i dati dello studente e del docente e, se ve ne sono, dei periti. Se sono state aggiunte delle motivazioni al formulario esse dovrebbero venire mostrate in una delle tre tabelle al punto a cui si riferiscono.</p> <p>Nell'ultima pagina dovrà essere riportata la nota correttamente.</p>		

5.2 Risultati test

Test Case	Risultato	Commento	Data
TC-01	OK	Si riesce ad aprire correttamente il sito senza ottenere messaggi di errore	01.04.2020
TC-02	OK	<p>L'account è stato creato correttamente come si può vedere nel database.</p> <p>11 Matteo Forni matte20012001@gmail.com 3456789012</p> <p>Inoltre l'email di conferma è arrivata.</p> 	01.04.2020
TC-03	OK	<p>Inserendo campi sbagliati appaiono sia le notifiche d'errore sia frontend che backend.</p>   	01.04.2020
TC-04	OK	<p>È possibile connettersi con un account esistente. Connettendosi la prima volta appare una notifica di verificare l'email.</p>  <p>Una volta fatto, al login successivo, essa non appare più.</p>	01.04.2020
TC-05	OK	Inserendo parole nel campo di testo i dati nella tabella variano a seconda di ciò che si ha scritto. La modifica funziona così come l'aggiunta. Entrambe le operazioni, se fatte in maniera scorretta, generano messaggi di errore.	01.04.2020
TC-06	OK	<p>Inserendo parole nel campo di testo i dati nella tabella variano a seconda di ciò che si ha scritto. La modifica funziona così come l'aggiunta. Modificando o aggiungendo un utente esso riceve un messaggio email.</p> <p>Abbiamo modificato il tuo account</p> <p>Abbiamo dovuto effettuare delle modifiche al tuo account, qui sotto troverai i tuoi nuovi dati:</p> <ul style="list-style-type: none"> Nome: Matteo Cognome: Forni Email: matte20012001@gmail.com Telefono: +413456789012 Ruolo: teacher <p>Entrambe le operazioni, se fatte in maniera scorretta, generano messaggi di errore.</p>	01.04.2020
TC-07	OK	Inserendo parole nel campo di testo i dati nella tabella variano a seconda di ciò che si ha scritto. L'eliminazione funziona correttamente così come tutti i collegamenti della tabella. Inoltre i bottoni sottostanti ad essa fanno ciò che devono e il template si apre correttamente.	01.04.2020
TC-08	OK	La pagina genera correttamente i formulari.	01.04.2020
TC-09	OK	Il form ritorna messaggi di errore sia frontend che backend se si inseriscono dati errati.	01.04.2020
TC-10	OK	Inserendo parole nel campo di testo i dati nella tabella variano a seconda di ciò che si ha scritto. L'aggiunta funziona correttamente e disabilita i campi aggiunti. Rimuovendo una motivazione essi si	01.04.2020

		riabilitano. Il collegamento con la pagina dei risultati funziona correttamente.	
TC-11	OK	Le motivazioni vengono caricate e suddivise per sezione correttamente e le note calcolate con la formula esatta.	01.04.2020
TC-12	OK	Il file PDF viene generato correttamente e i dati inseriti al posto giusto.	01.04.2020

5.3 Mancanze/limitazioni conosciute

Il progetto presenta alcune limitazioni che sono dovute alla struttura di come è stato creato e ai ruoli degli utenti, infatti i docenti non possono aggiungere le motivazioni per conto loro e quindi devono per forza cercarne una più o meno adeguata nella lista creata dagli amministratori. Questo sarebbe risolvibile in due modi: fare sì che i docenti possano, tramite una pagina apposita, creare le loro motivazioni oppure implementare un sistema di ticket dove un insegnante possa richiedere ad un amministratore di creare la motivazione necessaria.

6 Consuntivo

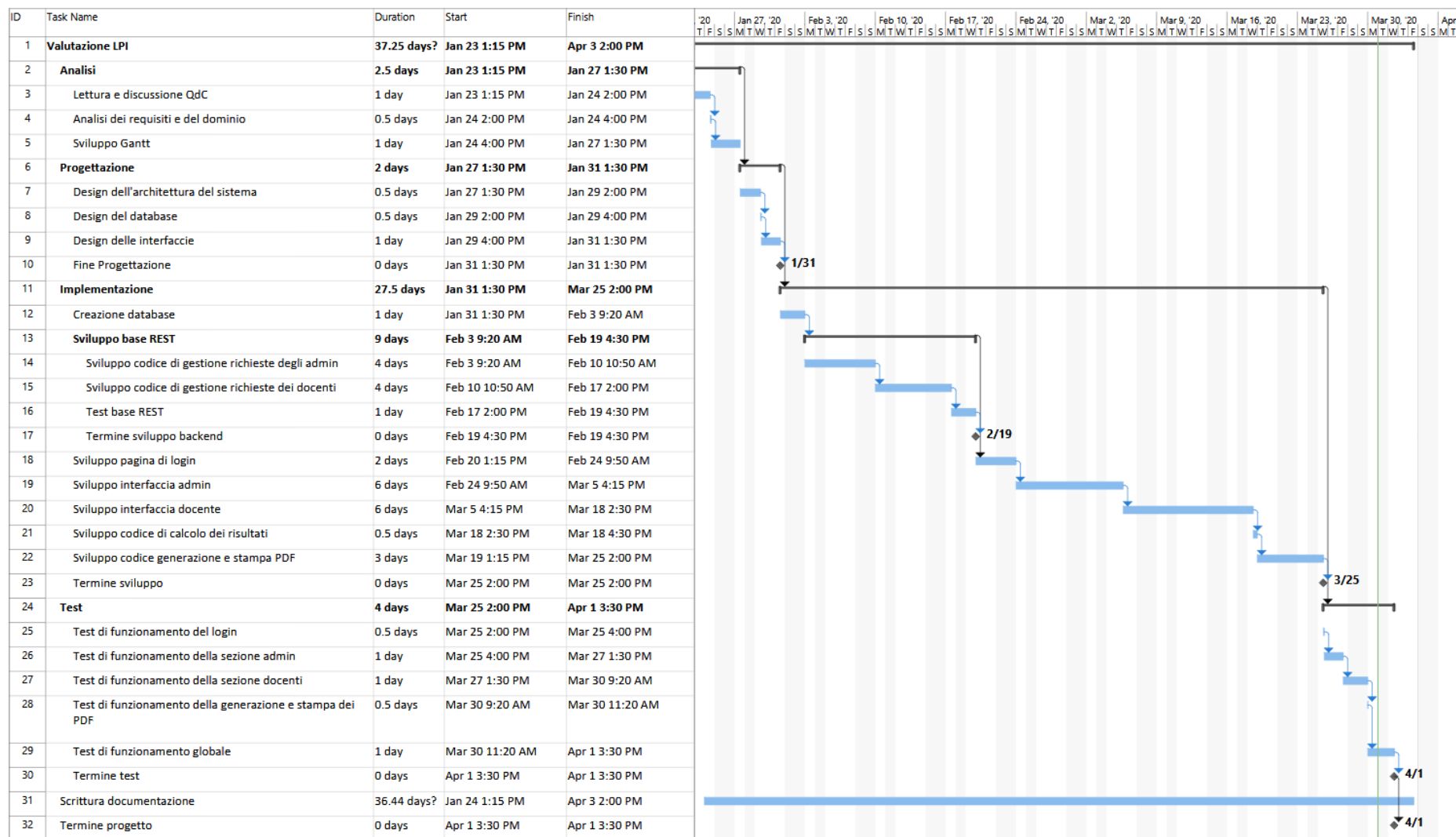


Figura 32 Diagramma di Gantt consuntivo

Dalla pianificazione originale non vi sono stati grandi cambiamenti, le principali differenze che si possono notare nel diagramma di Gantt consuntivo sono che lo sviluppo del codice backend ha preso meno tempo del previsto, ma questo è stato compensato dal fatto che il frontend ha occupato più di ciò che era stato calcolato. Inoltre altre attività quali: lo sviluppo codice di calcolo dei risultati e i test della sezione admin e docenti hanno occupato leggermente meno tempo di quello stabilito in precedenza. Al contrario la creazione dei PDF si è dilungata più del previsto.

7 Conclusioni

La soluzione implementata potrà tornare utile ed essere utilizzata dai docenti della scuola di Arti e Mestieri di Trevano nella correzione dei progetti di semestre delle terze e quarte così come da tutti i formatori del cantone. Questo è dovuto soprattutto ad una mancanza di mezzi digitali per svolgere questo compito.

I risultati ottenuti sono soddisfacenti e il progetto potrà essere portato avanti rendendolo più interessante, completo e utilizzato.

7.1 Sviluppi futuri

Un'aggiunta a cui ho pensato e che potrebbe venire effettuata in futuro è uno storico per gli allievi così che essi possano ritrovare facilmente le loro vecchie valutazioni per ricordarsi gli errori effettuati e correggerli nei progetti futuri. Questo implicherebbe la registrazione anche da parte degli allievi ma questo consentirebbe ad un docente di scegliere un allievo e tutti i suoi dati verrebbero inseriti nel formulario autonomamente.

Inoltre si potrebbe aggiungere la possibilità, ai docenti, di scrivere a mano almeno le motivazioni delle domande della presentazione in quanto esse sono molto variabili.

7.2 Considerazioni personali

Il progetto mi ha consentito di solidificare parecchie conoscenze già apprese in passato ma utilizzate di rado così come di migliorarmi in un discorso più globale dato che ho dovuto integrare parecchie nozioni di ambiti e materie diverse. Questo è stato il secondo progetto di preparazione all'esame finale e sono soddisfatto dei risultati ottenuti che mi hanno, anche, consentito di migliorare le mie abilità di gestione del tempo e delle attività da svolgere. Grazie a ciò ho preso ulteriore conoscenza dei miei punti deboli su cui dovrò lavorare di più in futuro.

Inoltre ho avuto la possibilità di conoscere un nuovo framework che mi ha consentito, una volta imparato il suo funzionamento, di eseguire questo lavoro in maniera più ordinata e modulare. Questa è stata una mia scelta in quanto volevo utilizzare qualcosa di nuovo e sono soddisfatto dei risultati ottenuti.

8 Bibliografia

8.1 Sitografia

- <https://github.com/CodeSeven/toastr>, Toastr notification Library, 21.02.2020
- <https://github.com/firebase/php-jwt>, PHP JSON Web Token Library, 26.02.2020
- <https://laracasts.com/discuss/channels/general-discussion/how-to-host-a-laravel-made-web-with-xampp>, Deploy Lumen on XAMPP, 23.03.2020
- <https://medium.com/@Dotunji/building-a-rest-api-with-lumen-403b67fec4d6>, Lumen build REST API, 12.02.2020
- <https://www.cloudways.com/blog/prevent-xss-exploits-using-laravel-validation-and-sanitization/>, Laravel Validation, 26.03.2020
- <https://lumen.laravel.com/docs/6.x>, Lumen 6 documentation, 27.01.2020
- <https://laravel.com/docs/6.x>, Laravel 6 documentation, 29.01.2020
- <https://laracasts.com/discuss/>, Laracasts discussion, 31.01.2020
- <https://php.net>, PHP documentation, 31.01.2020
- <https://api.jquery.com>, JQuery documentation, 03.02.2020
- <https://fontawesome.com/>, Fontawesome, 03.02.2020
- <https://stackoverflow.com/questions/>, Stackoverflow, 06.02.2020
- <https://tcpdf.org/>, TCPDF Library, 12.02.2020
- <https://github.com/PHPMailer/PHPMailer>, PHP Mailer Library, 13.02.2020
- <https://github.com/js-cookie/js-cookie>, Cookie JS Library, 17.02.2020
- <https://www.w3schools.com/>, W3school, 19.02.2020

9 Glossario

Vocabolo	Definizione
API	Insieme di funzioni e protocolli utilizzabili in un software
Browser	Programma che consente la navigazione in internet (es. Google, Firefox, ...)
Composer	Software di gestione di dipendenze di PHP
Controllers	Classi che uniscono i models con i controllers
Footer	Parte finale di una pagina web
Form	Sezione di una pagina in cui è possibile inserire dei dati
Framework	Struttura/architettura sulla quale sviluppare un software
Git	Software di gestione delle versioni
Header	Parte superiore di una pagina web, solitamente contiene il menu
Inbox	Cartella contenente i messaggi ricevuti di un indirizzo di posta elettronica
Integrità referenziale	Mantenimento dell'affidabilità dei dati di un database
JSON	Formato per lo scambio di dati
JWT	(JSON Web Token) Token di verifica basato su JSON
LPI	Lavoro Pratico Individuale
Middleware	Classi che consentono di filtrare gli accessi e le richieste
Modale	Finestra che appare a schermo
Models	Classi di comunicazione con il database
ORM	Tecnica di conversione di query in programmazione ad oggetti
PDF	Tipo di file non modificabile
PHP	Linguaggio di programmazione orientato alle pagine web dinamiche
Query	Richiesta al database
REST	Architettura basata su HTTP che prevede richieste e risposte secondo tale standard
Template	Base di un file su cui si apporteranno modifiche
Timestamp	Tipo di dato che rappresenta un data
Token	Insieme di caratteri alfanumerici utilizzato per identificare qualcuno
Views	Parte grafica che vedrà l'utente finale

10 Indice delle figure

<i>Figura 1 Diagramma degli use case</i>	9
<i>Figura 2 Diagramma di Gantt preventivo</i>	11
<i>Figura 3 Diagramma Gantt, sezione analisi</i>	12
<i>Figura 4 Diagramma Gantt, sezione progettazione</i>	12
<i>Figura 5 Diagramma Gantt, sezione implementazione</i>	13
<i>Figura 6 Diagramma Gantt, sezione test</i>	13
<i>Figura 7 Diagramma dell'architettura del sistema</i>	15
<i>Figura 8 Diagramma ER del database</i>	16
<i>Figura 9 Schema logico del database</i>	17
<i>Figura 10 Mockup della pagina di login</i>	19
<i>Figura 11 Mockup della pagina di registrazione</i>	20
<i>Figura 12 Mockup della pagina di reset della password</i>	21
<i>Figura 13 Mockup della pagina contenente lo storico</i>	21
<i>Figura 14 Mockup della pagina di aggiunta di un formulario</i>	22
<i>Figura 15 Mockup della pagina di aggiunta delle motivazioni ad un formulario</i>	23
<i>Figura 16 Mockup della pagina di resoconto</i>	24
<i>Figura 17 Mockup della pagina di amministrazione</i>	25
<i>Figura 18 Lumen richieste soddisfatte per secondo</i>	26
<i>Figura 19 Schema di funzionamento MVC</i>	27
<i>Figura 20 Lumen schermata iniziale</i>	27
<i>Figura 21 Schema di files e cartelle di Lumen</i>	28
<i>Figura 22 Schema UML della cartella Controllers</i>	32
<i>Figura 23 File PDF generato automaticamente</i>	38
<i>Figura 24 Pagina di login</i>	39
<i>Figura 25 Pagina di registrazione</i>	39
<i>Figura 26 Pagina di amministrazione, sezione motivazioni</i>	40
<i>Figura 27 Pagina di amministrazione, sezione utenti</i>	40
<i>Figura 28 Pagina dello storico</i>	41
<i>Figura 29 Pagina di aggiunta</i>	42
<i>Figura 30 Pagina di aggiunta delle motivazioni</i>	43
<i>Figura 31 Pagina dei risultati</i>	44
<i>Figura 32 Diagramma di Gantt consuntivo</i>	53

11 Allegati

Elenco degli allegati:

- Abstract
- Diari di lavoro
- Istruzioni di installazione del prodotto
- Quaderno dei Compiti
- Prodotto