Formulario di valutazione LPI

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

2 Analisi 4

2.1 Analisi del dominio 4

2.2 Analisi e specifica dei requisiti 4

2.3 Use case 7

2.4 Pianificazione 7

2.4.1 Sezione analisi 10

2.4.2 Sezione progettazione 10

2.4.3 Sezione implementazione 10

2.4.4 Sezione test 10

2.5 Analisi dei mezzi 10

2.5.1 Software 10

2.5.2 Hardware 11

3 Progettazione 11

3.1 Design dell’architettura del sistema 11

3.2 Design dei dati e database 12

3.2.1 Schema ER 12

3.2.2 Schema logico 13

3.2.3 Specifica dei dati 13

3.3 Design delle interfacce 15

3.3.1 Pagina di login 15

3.3.2 Pagina di registrazione 16

3.3.3 Pagina di reset della propria password 16

3.3.4 Pagina contenente lo storico 17

3.3.5 Pagina di aggiunta di un formulario 18

3.3.6 Pagina di aggiunta delle motivazioni 19

3.3.7 Pagina di resoconto finale 20

3.3.8 Pagina di amministrazione 21

4 Implementazione 22

5 Test 22

5.1 Protocollo di test 22

5.2 Risultati test 23

5.3 Mancanze/limitazioni conosciute 23

6 Consuntivo 23

7 Conclusioni 23

7.1 Sviluppi futuri 23

7.2 Considerazioni personali 23

8 Bibliografia 23

8.1 Bibliografia per articoli di riviste: 23

8.2 Bibliografia per libri 23

8.3 Sitografia 23

9 Allegati 24

# Introduzione

## 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

## 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 will modify the final evaluation which can then be saved in PDF and printed out.

## Scopo

Lo scopo di questo progetto è quello di creare 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.

# Analisi

## Analisi del dominio

Il prodotto dovrà essere un formulario per la valutazione degli LPI degli informatici in forma digitale, attualmente esiste unicamente la versione sia digitale che cartacea dei moduli ma sono 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 dai 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.

## 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 |

## Use case



Figure 1 Diagramma degli use case

La foto soprastante rappresenta gli 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.

## Pianificazione



Figure 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.

### Sezione analisi

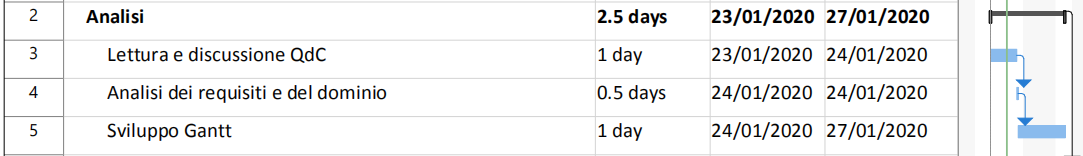


Figure 3 Diagramma Gantt, sezione analisi

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.

### Sezione progettazione

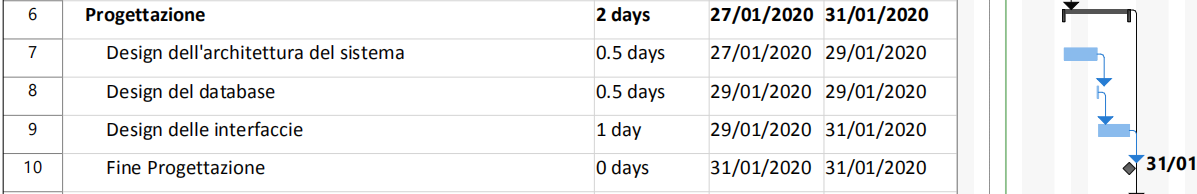


Figure 4 Diagramma Gantt, sezione progettazione

La seconda sezione del diagramma è la progettazione che comprende lo sviluppo teorico del sistema così da facilitarne l’implementazione effettiva. Questo sottocapitolo 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.

### Sezione implementazione

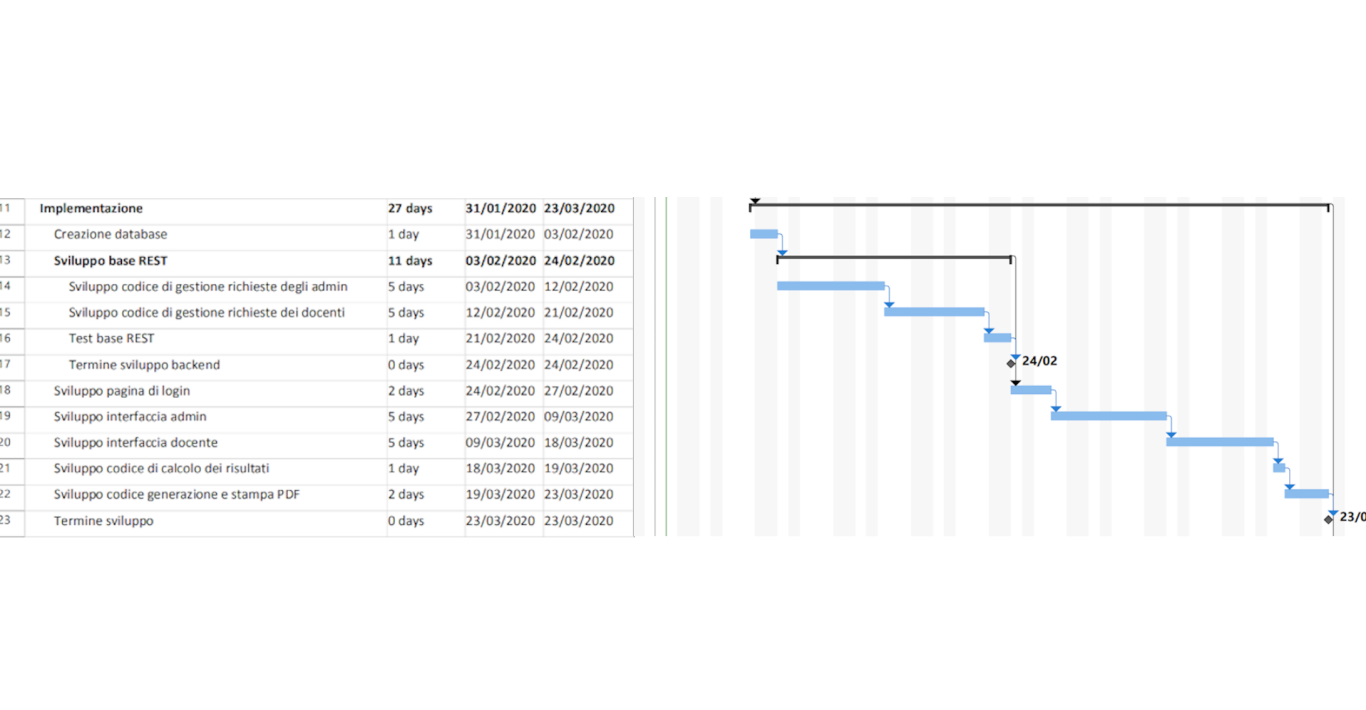


Figure 5 Diagramma Gantt, sezione implementazione

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.

### Sezione test



Figure 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.

## Analisi dei mezzi

### Software

SDK, librerie, tools utilizzati per la realizzazione del progetto e eventuali dipendenze.

### 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.

# Progettazione

## Design dell’architettura del sistema

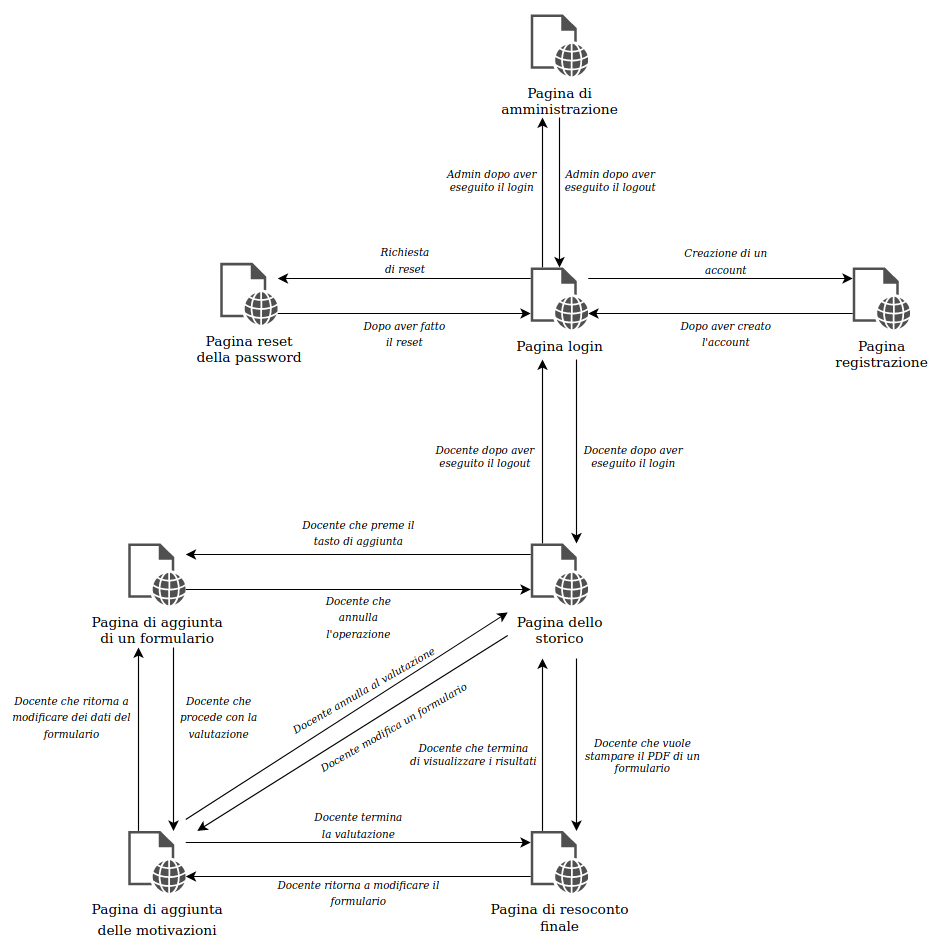


Figure 7 Diagramma dell'architettura del sistema

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 spegazione 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.

## Design dei dati e database

### Schema ER

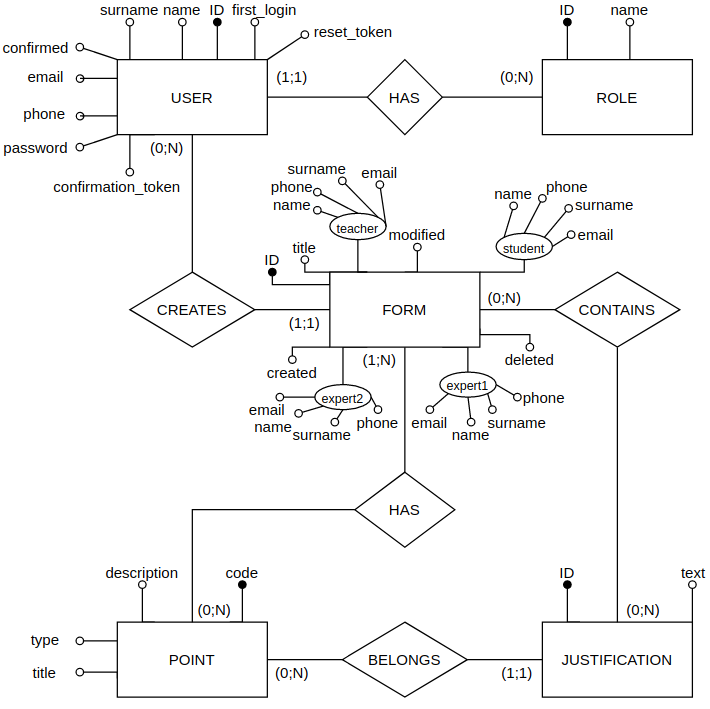


Figure 8 Diagramma ER del database

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.

### Schema logico

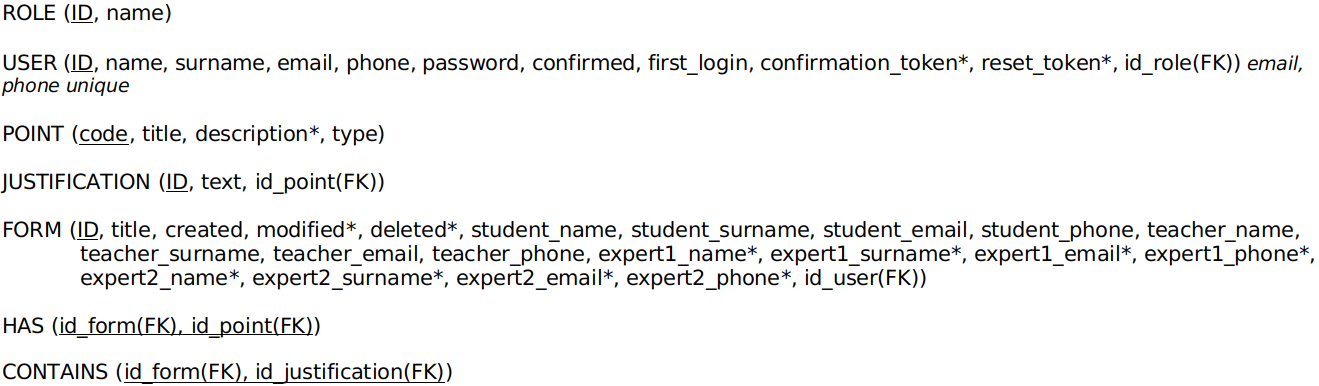


Figure 9 Schema logico del database

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 se 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.

### 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 |

## Design delle interfacce

### Pagina di login



Figure 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.

### Pagina di registrazione



Figure 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.

### Pagina di reset della propria password



Figure 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.

### Pagina contenente lo storico

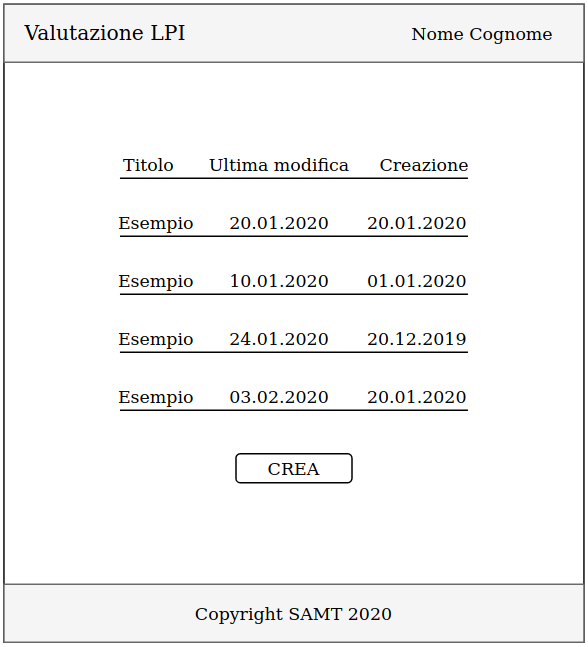


Figure 13 Mockup della pagina contenente lo storico

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.*

### Pagina di aggiunta di un formulario



Figure 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.

### Pagina di aggiunta delle motivazioni



Figure 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.

### Pagina di resoconto finale



Figure 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.

### Pagina di amministrazione



Figure 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.

# Implementazione

## 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*.

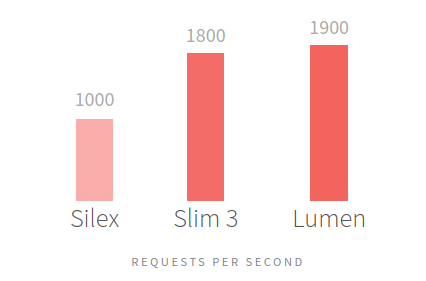


Figure 18 Lumen richieste soddisfatte per secondo

Lumen è pensato per fare delle piccole API come in questo caso e contiene già delle librerieche consentono di facilitare la scrittura del codice ad esempio *Facades* che da 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.*

### 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.



Figure 19 Lumen schermata iniziale

### La struttura delle cartelle

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

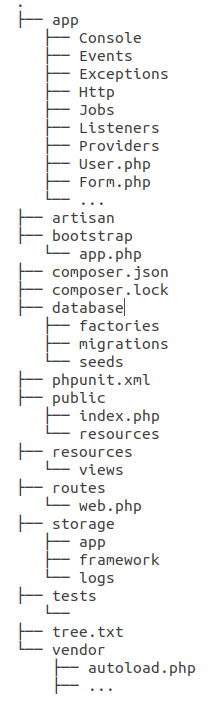


Figure 20 Schema di files e cartelle di Lumen

Le cartelle degne di nota sono quelle in cui verrà effettivamente scritto codice dallo sviluppatore, esse sono principalemente: 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, ...

### 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’ereditariatà 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.

### 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"

}

## Gestione delle versioni

Le versioni del progetto sono state gesite costantemente tramite una repository di GitHub su cui venivano fatti dei push più volte al giorno. Affianco ad’essa c’é 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>.

## 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.

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 dal database del progetto.

# Test

## Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l’adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-012 | **Nome:** | Import a card with KIC, KID and KIK keys, but not shown with the GUI |
| **Descrizione:** | Import a card with KIC, KID and KIK keys with no obfuscation, but not shown with the GUI | | |
| **Prerequisiti:** | Store on local PC: Profile\_1.2.001.xml (appendix n\_n) and Cards\_1.2.001.txt (appendix n\_n).  PIN (OTA\_VIEW\_PIN\_PUK\_KEY) and ADM (OTA\_VIEW\_ADM\_KEY) user right not set. | | |
| **Procedura:** | 1. Go to “Cards manager” menu,  in main page click “Import Profiles” link, Select the “1.2.001.xml” file, Import the Profile 2. Go to “Cards manager” menu,  in main page click “Import Cards” link, Select the “1.2.001.txt” file, Delete the cards,  Select the “1.2.001.txt” file, Import the cards 3. Research the “41795924770” Card, Click the imsi card link Check the card details 4. Execute the SQL: SELECT imsi, dir, keyset, cntr, rawtohex(kickey), rawtohex(kidkey), rawtohex(kikkey), rawtohex(chv), rawtohex(dap)FROM otacardkey a where imsi='340041795924770' ORDER BY keyset; | | |
| **Risultati attesi:** | Keys visible in the DB (OtaCardKey) but not visible in the GUI (Card details) | | |

## Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l’errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l’errore con eventuali ipotesi di correzione.

## Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

# Consuntivo

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

# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc

# Bibliografia

## Sitografia

* <https://stackoverflow.com/questions/,> *Stackoverflow*, 06.02.2020
* <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

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Codici sorgente/documentazione macchine virtuali
* Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
* Documentazione di prodotti di terzi
* Eventuali guide utente / Manuali di utilizzo
* Mandato e/o Qdc
* Prodotto
* …