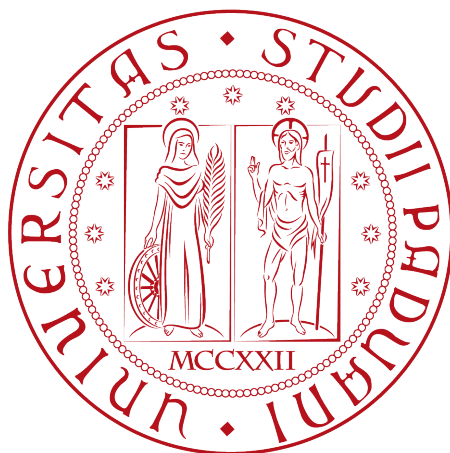


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA

CORSO DI LAUREA IN INFORMATICA



Refactoring di una piattaforma CMS: database MySQL, API REST ed interfaccia grafica

Tesi di laurea triennale

Relatore

Prof. Ombretta Gaggi

Laureando

Andrea Tombolato

ANNO ACCADEMICO 2015-2016

Indice

1	Il contesto aziendale	1
1.1	H-FARM	1
1.1.1	Primo pilastro: investimento	1
1.1.2	Secondo pilastro: industria	2
1.1.3	Terzo pilastro: istruzione	3
1.2	Digital Accademia	4
1.2.1	Formazione	4
1.2.2	Comunicazione interna	5
2	Il progetto di stage	9
2.1	Il contesto	9
2.2	Descrizione del progetto	11
2.3	Tecnologie utilizzate	12
2.3.1	Database	12
2.3.2	Framework	12
2.3.3	Versionamento	14
2.3.4	Gestione di progetto	14
3	Analisi dei requisiti	17
3.1	Il database	17
3.1.1	Descrizione dei requisiti	17
3.1.2	L'integrazione con Eloquent	19
3.2	Il CMS	20
3.2.1	Casi d'uso	20
3.3	Tracciamento dei requisiti	21
	Glossario	23
	Acronimi	25
	Bibliografia	27

Elenco delle figure

1.1	Logo H-FARM	1
1.2	Distribuzione temporale delle fasi previste dal programma H-CAMP	3
1.3	Programma H-CAMPUS rivolto a diverse fasce d'età	3
1.4	Logo Digital Accademia	4
1.5	Un post di tipo testo	6
1.6	Un post di tipo captcha	7
1.7	Un post di tipo team	7
2.1	Organizzazione della piattaforma prima del progetto di stage	11
2.2	Logo MySQL Workbench	12
2.3	Logo Laravel	13
2.4	Logo Bootstrap	14
2.5	Logo Bitbucket	14
2.6	Logo Git	14
2.7	Logo Basecamp	15
3.1	Use Case - UC0: Scenario principale	20

Capitolo 1

Il contesto aziendale

1.1 H-FARM

H-FARM nasce nel 2005 con l'obiettivo di aiutare i giovani imprenditori a lanciare le loro idee innovative e supportare la trasformazione delle aziende italiane in un'ottica digitale, il tutto ospitato in un'area verde di 90.000 m^2 all'interno del parco naturale del fiume Sile.

Il 13 Novembre 2015 H-FARM viene ammessa in Borsa, all'interno del segmento AIM (Alternative Investment Market).

Oggi H-FARM è una piattaforma innovativa che:

- Sostiene la creazione di nuovi modelli d'impresa attraverso investimenti in [startupG](#);
- Guida la trasformazione delle grandi aziende verso il digitale;
- Fornisce istruzione di alto livello nelle aree del digitale a studenti e professionisti.

Il modello d'impresa di H-FARM si basa su tre pilastri fortemente interconnessi tra loro: **investimento**, **industria** ed **istruzione**.



figura 1.1: Logo H-FARM

1.1.1 Primo pilastro: investimento

La chiave per sviluppare una visione d'investimento più ampia e creare nuovi progetti innovativi è il cambiamento. Vengono quindi prese in considerazione idee provenienti da tutta Europa, selezionando quelle più promettenti ed investendo su di esse.

La strategia d'investimento prevede:

- Investimenti diretti (da 100.000 € a 500.000 €) in [startup_G](#) italiane alle prime armi che abbiano ambizioni nazionali o globali nei campi SaaS e [B2B_G](#);
- Investimenti strategici in *InReach Ventures*: piattaforma di investimento che permette di accedere al fiorente ecosistema delle [startup_G](#) presenti in Europa. Ha sede a Londra ed è specializzata [finanziamento early stage_G](#) nel settore software;
- H-CAMP: programma di accelerazione della durata di 4 mesi che mira a sostenere giovani talenti nello sviluppo dei loro modelli d'impresa.

L'obiettivo principale è quello di scoprire ed investire nelle [startup_G](#) che si concentrano sulle eccellenze italiane: moda e design, cibo e vino, [IoT_G](#), viaggi e turismo.

H-CAMP

H-CAMP è un programma di accelerazione internazionale e “all-inclusive”, dedicato a quei talenti ed imprenditori che vogliono sviluppare le proprie idee. Ogni anno vengono lanciate due *call for ideas*: intervalli temporali entro i quali chiunque può fare domanda per partecipare ad H-CAMP.

H-CAMP fa parte del *Global Accelerator Network*, un'organizzazione internazionale che include tutti i più importanti acceleratori a livello mondiale. Questo permette alle [startup_G](#) che entrano a far parte di H-CAMP di poter accedere ad una serie di agevolazioni che vanno dal [mentoring_G](#) alle relazioni con investitori internazionali.

Il programma è strutturato in 3 fasi, gestite da persone responsabili con competenze tali da poter garantire i migliori risultati ad ogni fase:

1. **Selezione delle [startup_G](#):** vengono scelti i progetti ritenuti più meritevoli tra quelli che hanno risposto alla call for ideas;
2. **Programma di accelerazione:** la fase di accelerazione è il cuore pulsante del progetto H-CAMP. In questa fase le [startup_G](#) selezionate lavorano a stretto contatto con un team di [mentor_G](#) ed esperti per sviluppare e consolidare la loro idea d'impresa. Durante questa fase, inoltre, ogni [startup_G](#) riceve 20.000 € per coprire le proprie spese iniziali;
3. **Demo day:** evento conclusivo del periodo di accelerazione durante il quale le [startup_G](#) accelerate presentano il proprio prodotto ad una platea di investitori, aziende, giornalisti e azionisti di H-FARM.

1.1.2 Secondo pilastro: industria

Le nuove tendenze tecnologiche dei consumatori obbligano le aziende a rivalutare le modalità con le quali investono sul personale, sui processi e sulle tecnologie. H-FARM



figura 1.2: Distribuzione temporale delle fasi previste dal programma H-CAMP

aiuta la aziende a trasformare i loro modelli d'impresa e le loro attività in armonia con i cambiamenti digitali in atto.

La strategia per l'industria propone:

- **Digital transformation:** formazione, innovazione e rinnovamento dei processi organizzativi per diventare protagonisti del cambiamento digitale in ambito aziendale;
- **H-ACK:** maratone di 24 ore focalizzate su specifici settori dell'industria. I partecipanti si dividono in team e lavorano per trovare soluzioni a problemi concreti esposti dalla compagnia promotrice dell'evento.

1.1.3 Terzo pilastro: istruzione

I tradizionali sistemi educativi stanno rapidamente diventando obsoleti e la domanda di metodi di apprendimento innovativi è in rapida crescita. H-FARM è il più grande ecosistema digitale attualmente presente in Italia, che unisce le best practice del mondo accademico con nuovi metodi di apprendimento presi in prestito dal mondo del business e delle [startup_G](#).

Per Maggio 2017 è prevista l'inaugurazione di una nuova area dedicata ad H-CAMPUS: il nuovo progetto formativo di H-FARM dedicato alla diffusione della cultura digitale attraverso un approccio innovativo. Studenti, professionisti e manager saranno accompagnati e resi protagonisti del processo di trasformazione digitale. H-CAMPUS nascerà all'interno di un grande spazio aperto che andrà ad ospitare una [International School_G](#) (dai 6 ai 18 anni) ed una università con più di 20 tra insegnanti e formatori.



figura 1.3: Programma H-CAMPUS rivolto a diverse fasce d'età

1.2 Digital Accademia

Digital Accademia è il centro italiano di riferimento per la diffusione della cultura digitale ed è situata all'interno dell'ecosistema H-FARM.

Questo è il luogo dove studenti, professionisti e innovatori possono conoscere il potenziale del digitale, integrarlo nei propri processi di sviluppo e guardare al futuro. Il tutto viene veicolato attraverso un approccio concreto, con il quale si mira alla formazione attraverso l'esperienza.

L'offerta di Digital Accademia si rivolge principalmente alle aziende e propone strumenti di gestione della conoscenza, strumenti di comunicazione e sessioni formative che consentano di rimanere al passo con il cambiamento costante. La cultura digitale offre possibilità di interazione impensabili fino a poco tempo fa e permette di veicolare contenuti in modo nuovo, coinvolgente, personale e non convenzionale.



figura 1.4: Logo Digital Accademia

1.2.1 Formazione

Diffondere la cultura digitale in azienda significa prima di tutto venire in contatto con le persone e con le loro competenze. L'innovazione del digitale non è un concetto astratto ma una prassi, per questo non bastano corsi di aggiornamento o sessioni di formazione in aula. La formazione deve essere, prima di tutto, un'esperienza che coinvolge.

Bootcamp

All'interno dell'offerta di formazione si colloca il progetto **Bootcamp**: un'esperienza full immersion di 3 giorni e 3 notti sulla cultura digitale, ospitata in Villa Annia: il quartier generale di Digital Accademia.

Le giornate alternano momenti di teoria a sessioni di pura pratica, mentre i pranzi, le cene e le sere sono momenti sempre conviviali arricchiti dalla presenza di ospiti ed esperti del settore di pertinenza per:

- Imparare a pensare digitale;
- Capire come funziona una buona storia;
- Scoprire le logiche dietro a un e-commerce di successo;
- Costruire presentazioni persuasive.

1.2.2 Comunicazione interna

Le esperienze di comunicazione interna più efficaci parlano ai dipendenti come si parlerebbe al pubblico di un evento di *entertainment*. L'avvento massiccio del digitale ha fatto sì che non ci sia più differenza tra comunicazione esterna ed interna all'azienda.

Plot

All'interno dell'offerta per la comunicazione interna si colloca il progetto **Plot**.

Questo progetto si basa sui principi degli [Alternate Reality Game_G](#) per coinvolgere e divertire le persone all'interno di una organizzazione. I dipendenti si trasformano in giocatori e partecipano alla risoluzione di enigmi come in un romanzo giallo interattivo: analizzando i file aziendali e individuando informazioni nel mondo reale devono riuscire ad arrivare alla soluzione.

Pensato inizialmente per Vodafone, Plot è un'esperienza avvincente progettata per coinvolgere gli impiegati di tutto il mondo in un gioco mirato a salvare una loro collega, Sugita, rapita da un'intelligenza artificiale da essa stessa progettata.

Sugita rivela ai partecipanti di aver lasciato alcune *backdoor* durante la fase di testing dell'intelligenza artificiale, ogni backdoor richiede una password che può essere trovata da qualche parte all'interno della intranet Vodafone. Per trovare le password, i giocatori (divisi in gruppi) devono studiare prodotti specifici dell'azienda, guardare tutorial e interagire tra loro e con i sistemi interni.

Sono previsti dieci livelli e dieci settimane di gioco, durante le quali i team provenienti da tutto il mondo si sfidano per salvare Sugita.

Terminati i dieci livelli, è prevista un'ultima fase nella quale le migliori squadre da tutto il mondo si sfidano a Londra in un'intensa giornata di gioco.

La squadra vincitrice si aggiudica le 6.000 sterline messe in palio come ricompensa.

Il Plot sviluppato per Vodafone ha registrato un grande successo, con oltre 8.000 giocatori distribuiti in 21 paesi. Questo ha portato a riproporre il format per altri clienti, tra i quali TUI Group e Assicurazioni Generali.

La struttura Perchè abbia successo, il gioco non deve essere presentato come tale ma deve confondersi quanto più possibile con la realtà degli eventi. A questo scopo, la presentazione deve avvenire all'interno del contesto lavorativo. Ad esempio, nel caso di Vodafone, il meeting annuale dei dipendenti veniva interrotto da un video nel quale l'intelligenza artificiale comunicava di aver rapito Sugita e sfidava i colleghi a salvarla.

Dopo aver effettuato la registrazione presso il sistema, la piattaforma prevede una **fase di inviti** durante la quale ogni utente può invitare un certo numero di colleghi

a far parte della propria squadra.

Al termine di questa fase, un algoritmo si occupa di formare le squadre secondo le preferenze espresse dagli utenti. Idealmente, se l'utente A sceglie l'utente B e l'utente B accetta, gli utenti A e B vengono inseriti nella stessa squadra.

Una volta chiusa la fase di inviti e formate le squadre, inizia la **fase di gioco**. La fase di gioco si articola in varie **mission**, ognuna può essere pensata come un blog che, durante la settimana, viene popolato tramite **post** appartenenti a tre possibili tipologie:

- **Testo**: introduce, in modo molto generale, l'obiettivo della mission e può avere contenuti multimediali come immagini e video;
- **Captcha**: prevede una domanda rivolta singolarmente all'utente, atta a verificare che esso sia effettivamente un umano e non una macchina;
- **Team**: prevede una domanda rivolta alla squadra.

In questa fase, ogni settimana prevede due momenti chiave:

- Ogni **lunedì** viene pubblicata una nuova mission, alla quale sono associati inizialmente un post di tipo testo e un post di tipo captcha;
- Ogni **mercoledì** viene pubblicato un post di tipo team, associato alla mission del lunedì.

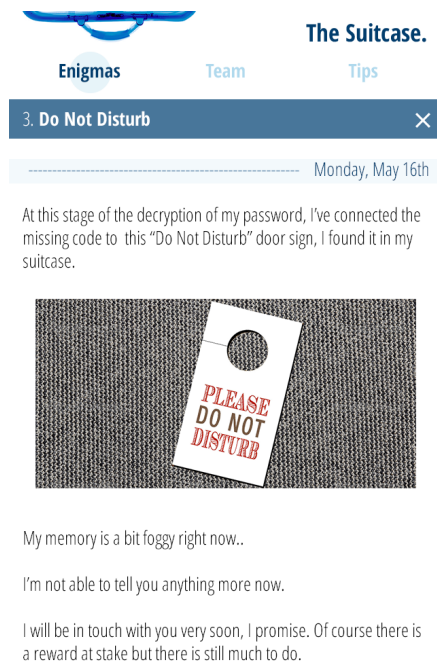


figura 1.5: Un post di tipo testo

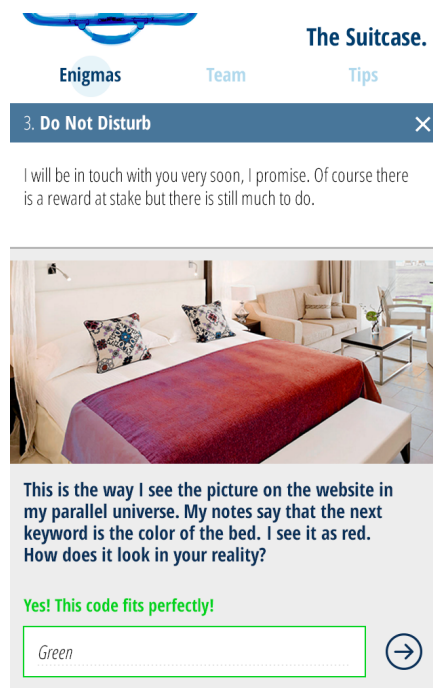


figura 1.6: Un post di tipo captcha

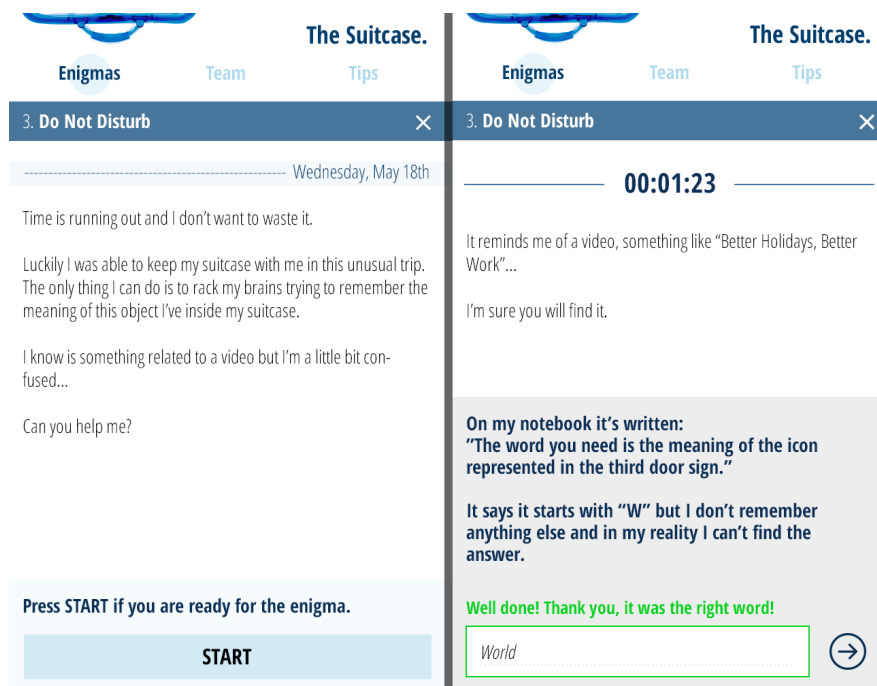


figura 1.7: Un post di tipo team

Per accedere al post di tipo team è necessario che almeno la metà dei componenti della squadra abbia risposto alla propria domanda di tipo captcha.

Le domande di tipo captcha sono uniche rispetto all'utente e alla squadra, con questo si intende che una specifica domanda di tipo captcha:

- non viene presentata allo stesso utente più di una volta;
- non viene presentata ai compagni di squadra.

La domanda associata al post di tipo team non è immediatamente visibile alla pubblicazione del post, ma può essere svelata da un componente della squadra. Una volta svelata la domanda di tipo team, viene avviato un cronometro che registra il tempo impiegato dalla squadra per rispondere.

Mentre le domande di tipo captcha sono pensate per richiedere uno sforzo minimo all'utente, quelle di tipo team hanno grado di difficoltà più alto, che aumenta col progredire del gioco.

Le domande sono pensate in modo da portare gli utenti ad affrontare specifici **touchpoint**: strumenti o concetti che l'azienda vuole far conoscere ai propri dipendenti.

Ad esempio, nel caso di Vodafone, Plot è stato progettato per perseguire i seguenti touchpoint:

- Far conoscere ai dipendenti di tutto il mondo una nuova area della intranet aziendale;
- Creare coesione tra i dipendenti;
- Attivare forme di coinvolgimento dirette attraverso la costruzione di un'esperienza;
- Trovare nuovi strumenti globali di comunicazione interna.

La squadra riceve un punteggio per ogni mission e, tale punteggio, è una funzione del tempo impiegato per rispondere alla domanda di tipo team.

Il punteggio totale di una squadra, che determina la posizione in classifica della stessa, è ricavato come somma dei punteggi ottenuti nelle varie mission.

Capitolo 2

Il progetto di stage

2.1 Il contesto

La piattaforma Plot è composta da due moduli Laravel distinti: un modulo di gioco ed un modulo **CMS_G**.

Il **modulo di gioco** incorpora la logica necessaria all'utente per interagire con la piattaforma. Questo modulo permette quindi agli utenti finali di:

- Autenticarsi;
- Invitare altri utenti a far parte della propria squadra;
- Accedere al contenuto delle mission;
- Accedere al contenuto dei post;
- Accedere e rispondere alle question che vengono loro proposte;
- Accedere alle informazioni riguardanti il proprio team;
- Consultare la classifica generale.

Il **modulo CMS_G** incorpora la logica necessaria agli amministratori per gestire il Plot. Questo modulo permette quindi agli amministratori di:

- Autenticarsi;
- Creare nuove mission;
- Eliminare mission esistenti;
- Modificare i dettagli di una particolare mission;
- Accedere all'elenco di tutte le mission;
- Creare nuovi post per una particolare mission;
- Eliminare post esistenti;
- Modificare i dettagli di un particolare post;

- Accedere all'elenco di tutti i post relativi ad una particolare mission;
- Creare nuove question per un particolare post;
- Eliminare question esistenti;
- Modificare i dettagli di una particolare question;
- Accedere all'elenco di tutte le question relative ad un particolare post;
- Accedere all'elenco degli utenti iscritti alla piattaforma;
- Eliminare utenti esistenti;
- Modificare i dettagli di un particolare utente;
- Accedere all'elenco dei compagni di team di un particolare utente;
- Accedere all'elenco dei team;
- Eliminare team esistenti;
- Modificare i dettagli di un particolare team;
- Accedere all'elenco dei componenti di un particolare team;
- Accedere alla classifica generale;
- Accedere all'elenco delle domande previste, corredate delle risposte corrette e del tipo di post al quale sono associate;
- Accedere all'elenco delle risposte fornite dagli utenti.

Entrambi i moduli Laravel utilizzano il medesimo database MySQL per la persistenza dei dati (Figura 2.1). Il database, pensato inizialmente per Vodafone, è stato adattato per diversi Plot, modificandolo di volta in volta. Le continue modifiche hanno portato ad avere un database inconsistente e poco manutenibile, a causa anche della scarsa documentazione.

Ogni modulo Laravel prevede un `front endG` collegato ad un relativo `back endG`. Le logiche che sottendono i due `back endG` sono, però, molto simili e questo porta ad avere codice duplicato e poco manutenibile.

Ulteriori problematiche nascono dal fatto che entrambi i `front endG` sono fortemente accoppiati al relativo `back endG`, questo implica una struttura poco modulare e difficilmente riusabile.

Nel prosieguo della trattazione, con l'appellativo "Plot 1.0" mi riferirò alla configurazione della piattaforma esistente prima dell'inizio del progetto di stage.

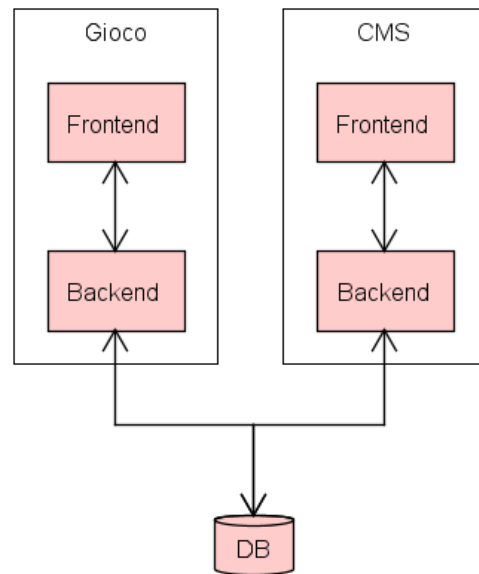


figura 2.1: Organizzazione della piattaforma prima del progetto di stage

2.2 Descrizione del progetto

Il progetto di stage prevede tre attività principali:

1. Refactoring del database che consente la persistenza dei dati utili alla piattaforma Plot;
2. Creazione di un set di API REST che permettano di interagire con il database (di cui al punto 1);
3. Refactoring del `front endG` relativo al `CMSG`.

La piattaforma Plot è in continua evoluzione e mira ad adattarsi, quanto più possibile, alle richieste dei clienti. Le attività di progetto devono quindi essere affrontate con l'obiettivo di massimizzare la modularizzazione, in modo da consentire una buona manutenibilità della piattaforma ed un basso accoppiamento tra le parti.

Dato il respiro internazionale e multinazionale dei potenziali clienti della piattaforma, deve essere implementato il concetto di **traducibilità**: ogni risorsa di testo importante ai fini del gioco deve poter essere localizzata in base alla lingua preferita dall'utente che ne usufruisce.

Mantenere uno storico delle azioni effettuate dai giocatori può risultare utile per individuare la causa di eventuali problemi inaspettati, si richiede quindi di prevedere un meccanismo di **logging** che registri e renda accessibili le richieste HTTP effettuate dai giocatori.

Il set di API REST ed il `front endG` del `CMSG` dovranno consentire di effettuare tutte le operazioni già disponibili dall'interfaccia grafica del `CMSG` di Plot 1.0

(elencate nella sezione "Il contesto"). In aggiunta alle operazioni già note, dovrà essere possibile:

- Accedere all'elenco degli amministratori della piattaforma;
- Eliminare gli amministratori;
- Modificare i dettagli di un particolare amministratore;
- Accedere all'elenco dei log generati dagli utenti del `front end`_G di gioco;
- Accedere all'elenco delle traduzioni disponibili per una data risorsa;
- Eliminare le traduzioni relative ad una data risorsa;
- Modificare i dettagli di un particolare traduzione.

Non è richiesta un'interfaccia grafica professionale per il modulo `CMS`_G.

2.3 Tecnologie utilizzate

2.3.1 Database

Per gestire il database è stato scelto **MySQL Workbench**: uno strumento di amministrazione visuale che permette di creare e modificare agevolmente lo schema di un database MySQL.

La scelta è ricaduta su questo strumento in quanto:

- Ufficialmente supportato e consigliato dalla comunità MySQL;
- Fornisce un procedimento assistito per la creazione dello schema, facilitando l'individuazione di errori.



figura 2.2: Logo MySQL Workbench

2.3.2 Framework

Back end

Il framework principale utilizzato per il progetto è **Laravel**: un framework PHP open-source, pensato per la creazione di applicazioni web basate sul pattern architetturale MVC (Model-View-Controller).

Il framework è stato scelto dall'azienda per i vantaggi che offre, tra i quali:

- Una documentazione ampia e facilmente consultabile in rete;
- Una sintassi pulita ed intuitiva;
- Facilità di implementazione della dependency injection;
- *Eloquent*: [ORM_G](#) integrato di Laravel;
- *Blade*: il template engine integrato di Laravel.



figura 2.3: Logo Laravel

Front end

Per il refactoring dell'interfaccia grafica del [CMS_G](#) è stato utilizzato **Bootstrap**: un framework HTML, CSS e JavaScript per la creazione di pagine web responsive e mobile-first.

Il framework è stato scelto perchè:

- Mette a disposizione tutti i componenti base e gli stili utili per la creazione di una pagina web (layout a griglia, tabelle, box modali, form e molto altro);
- Viene supportato da tutti i maggiori browser;
- Permette di creare interfacce grafiche consistenti, anche senza modificare le componenti fornite;
- Pensato per creare pagine responsive che si adattino anche ai dispositivi mobile;
- Include un set di icone pronte all'uso;
- Ha una buona documentazione ed un grande supporto dalla comunità;
- Esistono molti temi, template e plugin gratuiti.

Oltre ai molti pregi, questo framework porta con sé almeno due difetti:

- Produce codice HTML verboso che può risultare non perfettamente semantico;;
- Molti componenti non funzionano se JavaScript è disattivato.

In accordo con l'azienda, questi non sono stati ritenuti problemi bloccanti in quanto il [CMS_G](#) verrà ospitato su un server privato e sarà ad uso esclusivo degli amministratori, che avranno cura di utilizzare un browser che supporti JavaScript.



figura 2.4: Logo Bootstrap

2.3.3 Versionamento

Per il versionamento del codice è stato utilizzato **Bitbucket** con base **Git**, in quanto già in uso presso l'azienda. Bitbucket è un servizio di web-hosting per progetti software che utilizzano Git o Mercurial come sistemi di versionamento.

Il punto di forza di Bitbucket è la possibilità di creare, gratuitamente, repository privati. Questo risulta fondamentale per un'azienda che voglia mantenere il codice dei propri progetti accessibile solo a determinate persone.



figura 2.5: Logo Bitbucket

Git è un software per il versionamento distribuito ed è stato scelto dall'azienda in quanto offre:

- Semplicità di utilizzo;
- Una buona documentazione disponibile online;
- La possibilità di lavorare anche offline, in quanto ogni sviluppatore possiede una copia locale del repository sul quale può effettuare commit anche in assenza di connessione.



figura 2.6: Logo Git

2.3.4 Gestione di progetto

Per la gestione di progetto è stato utilizzato **Basecamp**, in quanto già in uso presso l'azienda.

Basecamp è un servizio web per che permette di condividere e organizzare tutto ciò che è necessario per la gestione di un progetto: task, discussioni, deadline e file.



figura 2.7: Logo Basecamp

Capitolo 3

Analisi dei requisiti

Sin dalla sua concezione, la piattaforma Plot è stata pensata come un prodotto unico e fatto su misura per il cliente. Questo ha impedito la definizione di una struttura di base sulla quale innestare eventuali cambiamenti.

La prima attività del progetto è stata dunque la consultazione dello schema del database e del codice relativo al CMS di Plot 1.0. Successivamente sono state intervistate le persone coinvolte nello sviluppo di Plot 1.0: developer, responsabili dei contenuti ed utilizzatori del CMS in modo da capire cosa non li soddisfacesse nella struttura attuale e cosa potesse essere fatto per migliorarla. Da queste interviste sono stati sintetizzati i requisiti che il database ed il CMS avrebbero dovuto avere. Tali requisiti sono stati fondamentali per la successiva progettazione ma possono risultare utili anche all'azienda per valutare in modo più razionale limiti e possibilità della piattaforma.

3.1 Il database

3.1.1 Descrizione dei requisiti

Un **consumer** è un concetto astratto che rappresenta un generico utilizzatore della piattaforma Plot, esso è caratterizzato da nome, cognome, email e password. Un consumer può concretizzarsi nel concetto di admin o user.

Un **admin** rappresenta un amministratore del CMS, ha quindi l'autorizzazione di accedere al CMS ma non può accedere alla parte di gioco.

Uno **user** rappresenta un utente del gioco e può quindi accedere alla parte di gioco ma non al CMS.

Solitamente i consumer si autenticano presso la piattaforma tramite email e password ma può capitare che gli user utilizzino un servizio di autenticazione esterno tramite *token*.

Un **invitation** rappresenta un invito che può essere spedito o ricevuto da uno user durante la fase di inviti. Un invitation è caratterizzato dal mittente e dal destinatario, che dovranno essere entrambi user.

Terminata la fase di inviti, ogni user appartiene ad un **team** caratterizzato da un nome e da un'immagine rappresentativa dello stesso. I team vengono creati a

priori e sono associati agli user solo al termine della fase di inviti.

Qualsiasi richiesta HTTP effettuata da uno user nel front end di gioco genera un **log** contenente:

- Un riferimento allo user che ha effettuato la richiesta;
- Indirizzo IP del dispositivo che ha effettuato la richiesta;
- Il metodo HTTP utilizzato per la richiesta;
- L'URL al quale è stata inoltrata la richiesta;
- Lo user agent che ha effettuato la richiesta;
- Eventuali dati aggiuntivi trasportati dalla richiesta.

Un **language** rappresenta una lingua parlata ed è caratterizzato da un nome (es. Italiano) e da un codice identificativo (es. it).

Una **translation** rappresenta una traduzione disponibile per un certo contenuto in un particolare language. Ogni user può indicare il proprio language preferito e questa informazione può essere utilizzata per fornire i contenuti di gioco localizzati in base allo user.

Uno user può appartenere ad una **category**, il significato della category non è fissato a priori ma può essere istanziato in base alle necessità del progetto e del cliente che lo richiede. Il concetto di category è quindi astratto e viene utilizzato per partizionare gli user sulla base di una o più caratteristiche. Ad esempio, gli user possono essere categorizzati per regione di appartenenza o per posizione lavorativa. Una category ha un nome che la identifica.

Per questo progetto vengono istanziate le due category più usate nei Plot passati: region e business role.

Una **region** rappresenta l'area del mondo alla quale appartiene un certo user ed è caratterizzata da una timezone. Una **timezone** identifica un particolare fuso orario ed è caratterizzata dal nome (es. Central European Time), dal codice identificativo della timezone (es. CET) e dall'offset rispetto allo UTC (es. +1). Una region può quindi rappresentare un insieme di stati, un singolo stato o una porzione di esso, questo dipende dallo specifica istanza di Plot.

Un **business role** rappresenta il ruolo dello user all'interno dell'azienda o, più in generale, l'area lavorativa alla quale esso afferisce.

Un Plot è composto da una o più **mission**. Ogni mission è caratterizzata da un nome, un'immagine rappresentativa, una data d'inizio e una data di fine, oltre la quale la mission si considera chiusa. Una mission è composta da più **post**. Ogni post prevede un testo ed appartiene ad una delle seguenti tipologie:

- **Text post**;
- **Captcha post**: prevede una domanda indirizzata ai singoli user;
- **Team post**: prevede una domanda indirizzata ai team.

Una **question** è un concetto astratto che rappresenta una generica domanda, essa prevede un testo che riporta il corpo della domanda stessa. Una question può essere istanziata da una user question o da una team question.

Una **user question** rappresenta una domanda indirizzata ai singoli user. Le user question sono caratterizzate dalle category alle quali sono rivolte. Se il concetto di category è utilizzato dalla particolare istanza di Plot, è possibile assegnare domande diverse a user appartenenti a diverse category. Viceversa, se il concetto di category non è utilizzato, ogni domanda può potenzialmente essere assegnata ad ogni user. Una user question è visibile alla pubblicazione del captcha post al quale appartiene. Una **team question** rappresenta una domanda indirizzata ai team. Le team question prevedono un tempo limite entro il quale devono ricevere risposta. Qualsiasi componente del team può fornire una o più risposte ad una specifica domanda. Una team question non è immediatamente visibile alla pubblicazione del team post al quale appartiene, per svelarla è necessario che un componente del team dichiari di volerla visualizzare, a quel punto la domanda è visualizzabile da tutti i membri del team. Nel momento in cui uno user decide di svelare una domanda, viene inizializzato un timer che decreta il tempo impiegato dal team per fornire una risposta corretta. Una funzione del tempo impiegato andrà a decretare il punteggio totalizzato dal team nella specifica mission. Il punteggio totale di un team è dato dalla somma dei punteggi totalizzati dal team nelle varie mission e, sulla base di questo, viene stilata la classifica dei team.

Una **correct answer** rappresenta una risposta corretta prevista per una specifica question, ogni question deve prevedere almeno una risposta corretta. Ogni user può fornire più risposte alle question proposte, una risposta ad una question è considerata corretta se e solo se il suo contenuto compare tra le correct answer previste per la specifica question.

Uno user può visualizzare un post più volte e, lo stesso post, può essere visualizzato da user diversi.

Un admin può decidere se mostrare (o mantenere nascosta) agli user una mission o un post.

Deve essere possibile fornire una translation per i contenuti testuali di mission, post, question ed in generale tutti gli elementi di gioco con i quali entra in contatto lo user.

3.1.2 L'integrazione con Eloquent

Per poter sfruttare appieno le potenzialità di Eloquent, l'ORM fornito con Laravel, è necessario rispettare alcune convenzioni:

- Lo schema di ogni tabella appartenente al database deve prevedere le colonne `created_at` e `updated_at`. Il valore relativo a queste colonne viene automaticamente inserito dall'ORM ogniqualvolta venga creato o modificato un record;

- Lo schema di ogni tabella appartenente al database deve prevedere una colonna relativa alla primary key e tale colonna deve chiamarsi `id`;
- I nomi di tabelle ed attributi devono essere in minuscolo e seguire la convenzione snake case;
- I nomi delle tabelle devono essere pensati al plurale. Ad esempio, la tabella contenente i captcha post dovrà essere chiamata `captcha_posts`;
- Gli attributi che fungono da foreign key devono essere nominati posponendo "id" al nome singolare della tabella alla quale si riferiscono. Ad esempio, una foreign key che si riferisca alla tabella users dovrà essere chiamata `user_id`.

3.2 Il CMS

3.2.1 Casi d'uso

Per lo studio dei casi di utilizzo del prodotto sono stati creati dei diagrammi. I diagrammi dei casi d'uso (in inglese *Use Case Diagram*) sono diagrammi di tipo [Unified Modeling Language \(UML\)](#) dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso. Essendo il progetto finalizzato alla creazione di un tool per l'automazione di un processo, le interazioni da parte dell'utilizzatore devono essere ovviamente ridotte allo stretto necessario. Per questo motivo i diagrammi d'uso risultano semplici e in numero ridotto.

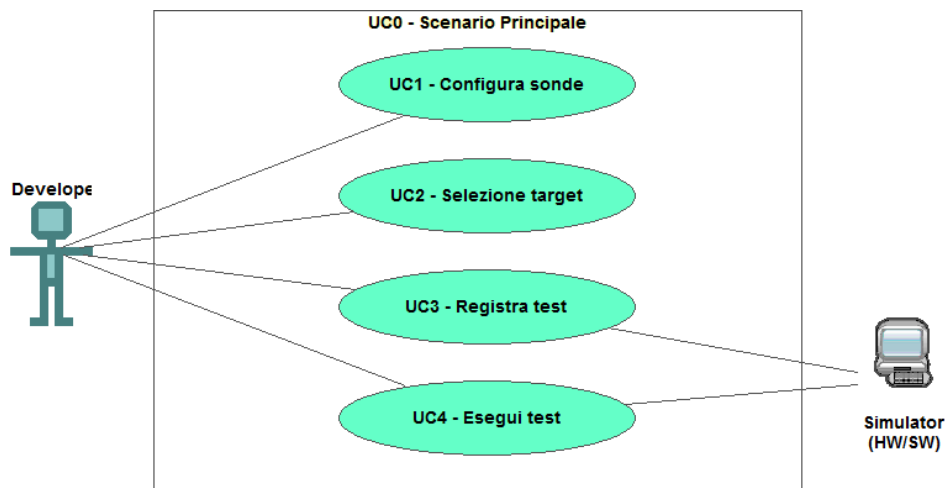


figura 3.1: Use Case - UC0: Scenario principale

UC0: Scenario principale

Attori Principali: Sviluppatore applicativi.

Precondizioni: Lo sviluppatore è entrato nel plug-in di simulazione all'interno dell'IDE.

Descrizione: La finestra di simulazione mette a disposizione i comandi per configurare, registrare o eseguire un test.

Postcondizioni: Il sistema è pronto per permettere una nuova interazione.

3.3 Tracciamento dei requisiti

Da un'attenta analisi dei requisiti e degli use case effettuata sul progetto è stata stilata la tabella che traccia i requisiti in rapporto agli use case.

Sono stati individuati diversi tipi di requisiti e si è quindi fatto utilizzo di un codice identificativo per distinguerli.

Il codice dei requisiti è così strutturato $R(F/Q/V)(N/D/O)$ dove:

R = requisito

F = funzionale

Q = qualitativo

V = di vincolo

N = obbligatorio (necessario)

D = desiderabile

Z = opzionale

Nelle tabelle 3.1, 3.2 e 3.3 sono riassunti i requisiti e il loro tracciamento con gli use case delineati in fase di analisi.

tabella 3.1: Tabella del tracciamento dei requisiti funzionali

Requisito	Descrizione	Use Case
RFN-1	L'interfaccia permette di configurare il tipo di sonde del test	UC1

tabella 3.2: Tabella del tracciamento dei requisiti qualitativi

Requisito	Descrizione	Use Case
RQD-1	Le prestazioni del simulatore hardware deve garantire la giusta esecuzione dei test e non la generazione di falsi negativi	-

tabella 3.3: Tabella del tracciamento dei requisiti di vincolo

Requisito	Descrizione	Use Case
RVO-1	La libreria per l'esecuzione dei test automatici deve essere riutilizzabile	-

Glossario

Alternate Reality Game spesso riferito anche con l'acronimo “ARG”, è una situazione di gioco che collega internet al mondo reale. Solitamente si sviluppa attraverso numerosi strumenti web (blog, email, mini-siti) e presenta al giocatore una storia misteriosa con indizi che puntano al mondo reale (per esempio a monumenti o a veri e propri oggetti nascosti in determinate località). [5](#)

Back end termine che denota l'insieme delle applicazioni, relative ad una piattaforma web, con le quali l'utente non interagisce direttamente ma che sono essenziali al funzionamento del sistema. [10](#)

B2B acronimo di “*business-to-business*”. Locuzione utilizzata per descrivere le transazioni commerciali elettroniche tra imprese, distinguendole da quelle che intercorrono tra le imprese e altri gruppi, come quelle tra una ditta e i clienti individuali (B2C, acronimo di “*business-to-consumer*”) oppure quelle tra una impresa e il governo (B2G, acronimo di “*business-to-government*”). [2](#)

CMS acronimo di “*Content Management System*”. Con questo termine si indica uno strumento software, installato su un server web, il cui obiettivo è facilitare la gestione dei contenuti di siti web, soprattutto a coloro che non hanno conoscenze specifiche sulla programmazione web. [9](#), [11–13](#)

Finanziamento early stage investimento atto a sostenere le fasi iniziali di un'azienda o di un business. [2](#)

Front end termine che denota la parte, di una piattaforma web, visibile all'utente e con la quale esso può interagire. Ci si riferisce a questo concetto anche con il termine “*interfaccia utente*”. [10–12](#)

International School prevede un percorso didattico che si articola seguendo il modello anglosassone ed è certificato dall' *International Baccalaureate Organization*. Al termine del percorso, con il conseguimento dell' *IB-Diploma Programme* gli studenti ottengono un diploma equipollente alla Maturità italiana con un anno di anticipo rispetto al percorso italiano. Hanno accesso a tutte le Università italiane e ai più prestigiosi atenei del mondo. [3](#)

IoT acronimo di “*internet of things*”. Neologismo riferito ad una evoluzione dell'uso della rete: gli oggetti si rendono riconoscibili e reagiscono di conseguenza grazie al fatto di poter comunicare dati su se stessi e accedere ad informazioni

aggregate da parte di altri. Le sveglie suonano prima in caso di traffico, le scarpe da ginnastica trasmettono tempi, velocità e distanza per gareggiare in tempo reale con persone dall'altra parte del globo, i contenitori delle medicine avvisano i familiari se si dimentica di prendere il farmaco. Tutti gli oggetti possono acquisire un ruolo attivo grazie al collegamento alla Rete. [2](#)

Mentor nell'ambito del mentoring, è il soggetto con più esperienza. Deve avere capacità relazionali, saper condurre colloqui e porre domande sagge, deve saper gestire le fasi del processo di mentoring. [2](#)

Mentoring metodologia di formazione che fa riferimento a una relazione uno a uno tra un soggetto con più esperienza (mentor) e uno con meno esperienza (*junior*, *mentee*, *protégé*), cioè un allievo, al fine di far sviluppare a quest'ultimo competenze in ambito formativo, lavorativo e sociale e di sviluppare autostima, a livello educativo-scolastico. [2](#)

ORM acronimo di “*Object-Relational Mapping*”. Con questo termine ci si riferisce ad una tecnica di programmazione che favorisce l'integrazione tra sistemi software orientati agli oggetti e sistemi . Più nello specifico, un prodotto ORM fornisce accesso alla persistenza dei dati attraverso un'interfaccia ad oggetti, astruendo così dai dettagli dello specifico [RDBMS](#) utilizzato. [13](#)

RDBMS acronimo di “*Relational Database Management System*”. Con questo termine ci si riferisce ad un sistema per la gestione di basi di dati relazionali. [24](#)

Startup in economia, con questo termine, si indica una nuova impresa nelle forme di un'organizzazione temporanea o una società di capitali in cerca di un business model ripetibile e scalabile. La scalabilità è un elemento cardine di questa tipologia di impresa. L'avvio di un'attività imprenditoriale non scalabile, come l'apertura di un ristorante, non coincide dunque con la creazione di una startup ma, piuttosto, di una società tradizionale. [1–3](#)

UML in ingegneria del software *UML*, *Unified Modeling Language* (ing. linguaggio di modellazione unificato) è un linguaggio di modellazione e specifica basato sul paradigma object-oriented. L'*UML* svolge un'importantissima funzione di “lingua franca” nella comunità della progettazione e programmazione a oggetti. Gran parte della letteratura di settore usa tale linguaggio per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile a un vasto pubblico. [25](#)

Acronimi

UML [Unified Modeling Language](#). 20

Bibliografia

Riferimenti bibliografici

Erich Gamma Ralph Johnson, Richard Helm e John Vlissides. *Design Patterns*. Addison-Wesley, 1994.

Paolo Atzeni Stefano Ceri, Stefano Paraboschi e Riccardo Torlone. *Basi di Dati*. McGraw Hill, 2014.

Siti Web consultati

Laravel Documentation. URL: <https://laravel.com/docs/>.