



UNIVERSITÀ DEGLI STUDI DI SALERNO

FACOLTÀ' DI SCIENZE MM. FF. NN.

CORSO DI LAUREA TRIENNALE IN INFORMATICA

Corso di Ingegneria del Software

LinkedPeople

LinkedPeople Object Design Document

Versione 1.1

ANNO ACCADEMICO 2018/2019

Coordinatore del progetto:

Nome	Matricola
Andrea De Lucia	
Rita Francese	

Partecipanti:

Nome	Matricola
Bruno D'Agostino	0512103598

Revision History

Data	Versione	Descrizione	Autore
09/10/2018	1.0	Prima stesura del documento	Membri del team
20/10/2018	1.1	Aggiornamento e completamento	Membri del team

Indice

Indice	2
Introduzione	3
Object Design Trade-Offs	3
Linee Guida per la Documentazione delle Interfacce	3
Classi PHP	4
Pagine PHP	5
Script JavaScript	5
Fogli di Stile CSS	5
Database	5
Definizioni, Acronimi e Abbreviazioni	6
Riferimenti	6
Directories	6
Directory Principale	7
Directory Classes	8
Directory Handlers	9
Directory Config	10
Directory View	10
Directory Test	11
Interfacce delle Classi	11
Gestione Account	11
Gestione Posts	12
Gestione Messaggi	12
Gestione Notifiche	12
Glossario	12

Introduzione

Con la stesura del documento di analisi dei requisiti (RAD) e del documento di progettazione del sistema (SDD) abbiamo descritto in linea di massima ciò che sarà il nostro sistema, con le relative funzionalità, ma abbiamo tralasciato tutto ciò che riguarda l'aspetto implementativo.

Gli sviluppatori si trovano a dover risolvere un puzzle che ha alcuni pezzi mancanti.

Questi pezzi si trovano durante la fase di Object Design.

In questo documento colmeremo questa mancanza, andando a definire tutto ciò che riguarda la parte di implementazione come la definizione delle interfacce, le varie tipologie di oggetti, le operazioni, le firme dei sottosistemi definiti nel SDD, ecc..

Ha come obiettivo finale quello di produrre un modello di riferimento capace di definire ed integrare in modo concreto, definito e corretto tutte le specifiche e i requisiti definiti nei documenti precedenti.

Object Design Trade-Offs

1. **Throughput vs. Latenza:** Il nostro progetto utilizza il flusso di controllo basato sugli eventi, ovvero il Web Server assegna un nuovo thread per ogni richiesta consentendo la gestione parallela delle richieste. Ciò aumenta il throughput ma il rischio di sovraccarico del sistema diminuisce.
2. **Spazio di memoria vs. Tempo di risposta:** Ci deve essere, per quanto possibile visti i tempi limitati, il giusto mix di ottimizzazione tra l'uso delle risorse hardware (RAM, storage) e i tempi di risposta.
3. **Funzionalità vs. Usabilità:** Il nostro software include funzionalità che soddisfano le specifiche dei requisiti. Con il server MySQL come componente del database, PHP come oggetto business e come front-end, l'usabilità del sistema è pienamente raggiunta.
4. **Interfaccia vs. Usabilità:** L'interfaccia grafica del sistema è stata progettata in modo da essere intuitiva, semplice e chiara, fa uso di form e pulsanti disposti in modo tale da rendere facilmente fruibile l'utilizzo del sistema da parte dell'utente finale.
5. **Sicurezza vs. Efficienza:** La sicurezza è fra i requisiti più importanti del sistema. Tuttavia, dati i tempi di sviluppo molto limitati, ci limiteremo ad implementare sistemi di sicurezza basilari come l'autenticazione degli utenti mediante email e password.
6. **Chiarezza codice vs. Tempo sviluppo:** Il codice deve essere strutturato in maniera tale da essere facilmente compreso e mantenuto anche da persone estranee allo sviluppo (tester) per facilitarne il lavoro e lo sviluppo di nuove funzionalità. Questo però causa inevitabilmente un aumento nei tempi di sviluppo.

Linee Guida per la Documentazione delle Interfacce

Questa sezione fornirà una panoramica sulla Naming Convention per rendere il programma più comprensibile e rendere il codice più facile da leggere.

Tipo di identificativo	Regole di denominazione
Directory	Il prefisso del nome della directory univoco è sempre scritto in minuscolo.

Classi, Pagine	I nomi delle classi devono cominciare con una lettera maiuscola e i nomi delle pagine devono cominciare con una lettera minuscola, e anche le parole seguenti all'interno del nome (se presenti) devono cominciare con una lettera maiuscola. I nomi di quest'ultime devono essere semplici, non lunghi e descrittivi.
Interfacce	I nomi delle interfacce devono cominciare con una lettera minuscola.
Metodi	<p>I nomi dei metodi devono cominciare con una lettera minuscola, e le parole seguenti con la lettera maiuscola. Il nome del metodo tipicamente consiste di un verbo che identifica un'azione, seguito dal nome di un oggetto; deve essere non lungo, semplice e descrittivo.</p> <p>I nomi dei metodi per l'accesso e la modifica delle variabili dovranno essere del tipo <code>getNomeVariabile()</code> e <code>setNomeVariabile()</code>.</p> <p>Le variabili dei metodi devono essere dichiarate appena prima del loro utilizzo e devono servire per un solo scopo, per facilitare la leggibilità del codice.</p>
Variabili, costanti	I nomi delle variabili devono cominciare con una lettera minuscola, e le parole seguenti devono essere precedute da un trattino basso con la lettera maiuscola. Per una migliore leggibilità del codice è preferibile che le variabili vengano dichiarate ad inizio blocco, solamente una per riga.

Classi PHP

Le classi PHP devono rispettare la seguente struttura:

- Una breve descrizione
- L'istruzione che permette di fare l'import degli oggetti che la classe utilizza
- Dichiarazione della classe pubblica
- Dichiarazioni di costanti (se presenti)
- Dichiarazioni di variabili di classe (se presenti)
- Dichiarazioni di variabili d'istanza
- Costruttore (se presente o necessario)
- Dichiarazione metodi

Poi:

- Si devono usare le tabulazioni per l'indentazione.

Pagine PHP

Le pagine PHP devono, quando eseguite, produrre, in ogni circostanza, un documento conforme allo standard HTML versione 5.

Le parti PHP delle pagine devono aderire alle convenzioni per la codifica in PHP, con le seguenti puntualizzazioni:

- Il tag di apertura (<?php) è seguito immediatamente dalla fine della riga;
- Il tag di chiusura (?>) si trova all'inizio di una riga;

Script JavaScript

Gli script JavaScript che svolgono funzioni distinte dal rendering della pagina dovrebbero essere collocati in file dedicati.

Fogli di Stile CSS

Tutti gli stili non inline devono essere collocati in fogli di stile separati.

Ogni foglio di stile deve essere iniziato da un commento analogo a quello presente nei file Java. Ogni regola CSS deve essere formattata come segue:

- I selettori della regola si trovano a livello 0 di indentazione, uno per riga;
- L'ultimo selettore della regola è seguito da parentesi graffa aperta ({});
- Le proprietà che costituiscono la regola sono listate una per riga e sono indentate rispetto ai selettori;
- La regola è terminata da una parentesi graffa chiusa (}), collocata da sola su una riga;

Database

I nomi delle tabelle devono seguire le seguenti regole:

- devono essere costituiti da sole lettere minuscole;
- il nome deve essere un sostantivo singolare tratto dal dominio del problema ed esplicativo del contenuto.

I nomi dei campi devono seguire le seguenti regole:

- devono essere costituiti da sole lettere minuscole;
- se il nome è costituito da più parole, queste sono rappresentate con le lettere minuscole e divise da un trattino basso (es. 'prima_seconda');
- il nome deve essere un sostantivo singolare tratto dal dominio del problema ed esplicativo del contenuto.

Definizioni, Acronimi e Abbreviazioni

Definizione	Descrizione
Naming Convention	Insieme di regole per la scelta della sequenza di caratteri da utilizzare per gli identificatori che denotano variabili, tipi, funzioni e altre entità nel codice sorgente e nella documentazione.
Throughput	Tasso massimo al quale qualcosa può essere elaborato.
Latenza	Tempo impiegato per la completa elaborazione di una operazione.

Acronimo / Abbreviazione	Descrizione
UI	User Interface
DB	DataBase
RAD	Requirements Analysis Document
SDD	System Design Document
PHP	Hypertext Preprocessor

Riferimenti

- B. Bruegge, A. H. Dutoit, Object Oriented Software Engineering-Using UML, Pattern and Java, Prentice Hall, 3rd edition, 2009.
- Documento RAD del progetto LinkedPeople.
- Documento SDD del progetto LinkedPeople.

Directories

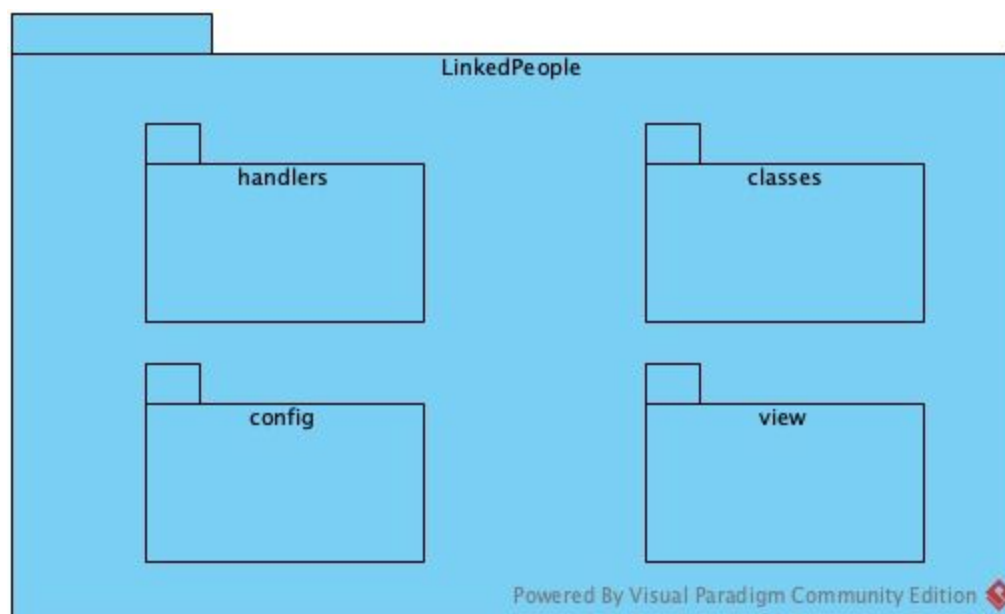
La gestione del nostro sistema è suddivisa in tre livelli (Three-Tier):

- Storage layer
- Interface layer
- Application Logic layer

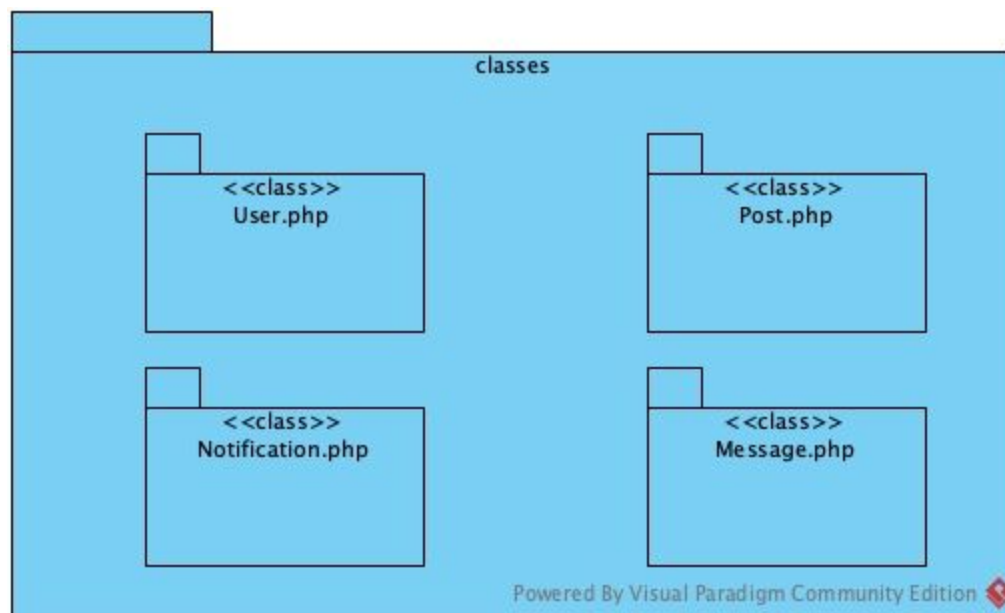
La directory principale LinkedPeople contiene sotto-directory che a loro volta inglobano classi atte alla gestione delle relative funzionalità.

Nome layer	Descrizione
Interface	Rappresenta l'interfaccia del sistema, offre la possibilità all'utente di interagire con quest'ultimo offrendo sia la possibilità di inviare che di visualizzare dati.
Application Logic	Ha il compito di elaborare i dati da inviare al client e permette, grazie a delle opportune richieste fatte al database tramite il Storage Layer, di accedere ai dati persistenti. Si occupa delle varie gestioni come: <ul style="list-style-type: none"> • Gestione Account • Gestione Posts • Gestione Messaggi • Gestione Notifiche
Storage	È il livello che gestisce direttamente i dati (utilizzando un DBMS), la logica e le regole dell'applicazione.

Directory Principale

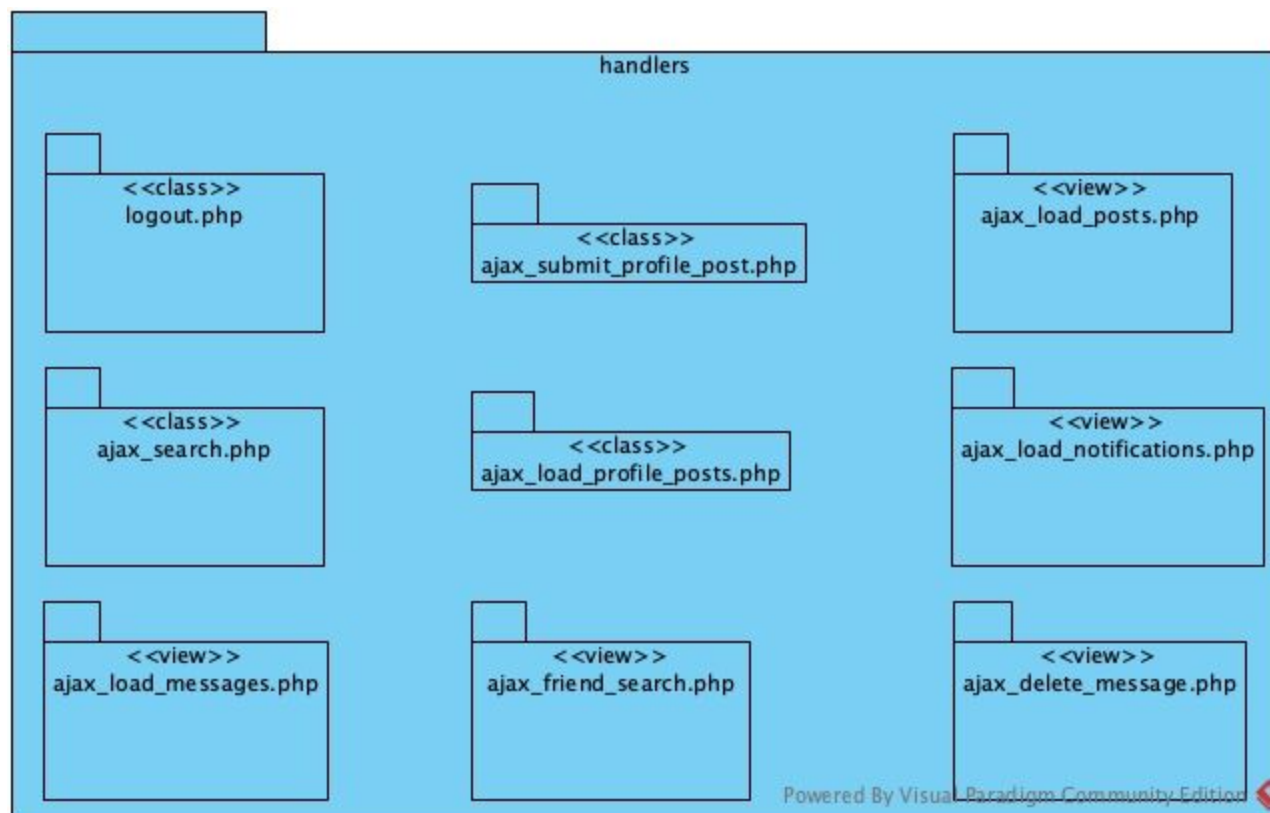


Directory Classes



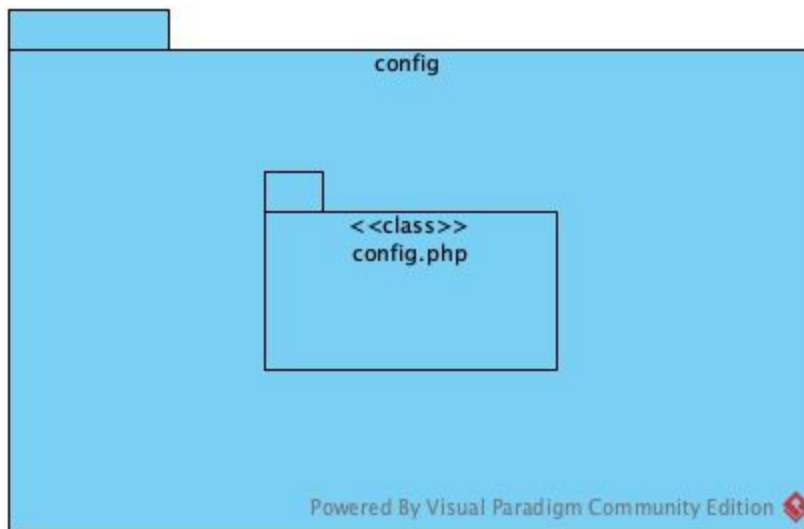
CLASSE	DESCRIZIONE
User.php	Descrive l'utente generico del sistema.
Post.php	Descrive il post pubblicato dall'utente.
Notification.php	Descrive la notifica che si riceve dopo ogni evento generato dall'utente.
Message.php	Descrive il messaggio che l'utente invia.

Directory Handlers



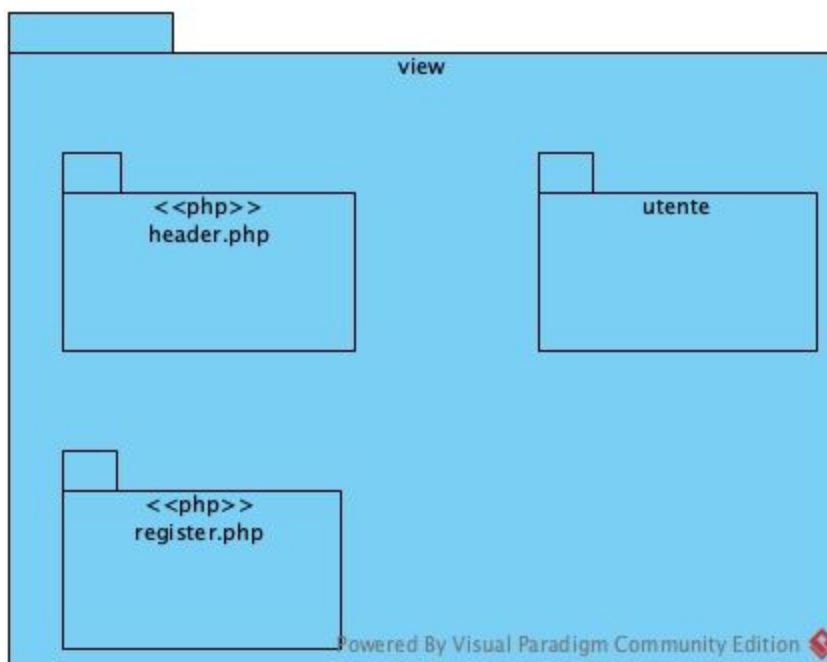
CLASSE	DESCRIZIONE
logout.php	Gestisce il processo di logout dell'utente.
ajax_search.php	Gestisce la ricerca nel sistema.
ajax_load_messages.php	Gestisce il caricamento dei messaggi.
ajax_submit_profile_post.php	Gestisce la pubblicazione di un post sulla bacheca di un utente.
ajax_load_profile_posts.php	Gestisce il caricamento dei posts sul proprio profilo.
ajax_friend_search.php	Gestisce la ricerca degli amici nel sistema.
ajax_load_posts.php	Gestisce il caricamento dei posts sulla newsfeed.
ajax_load_notifications.php	Gestisce il caricamento delle notifiche.
ajax_delete_message.php	Gestisce la cancellazione dei messaggi.

Directory Config



CLASSE	DESCRIZIONE
config.php	Gestisce la configurazione con il database.

Directory View



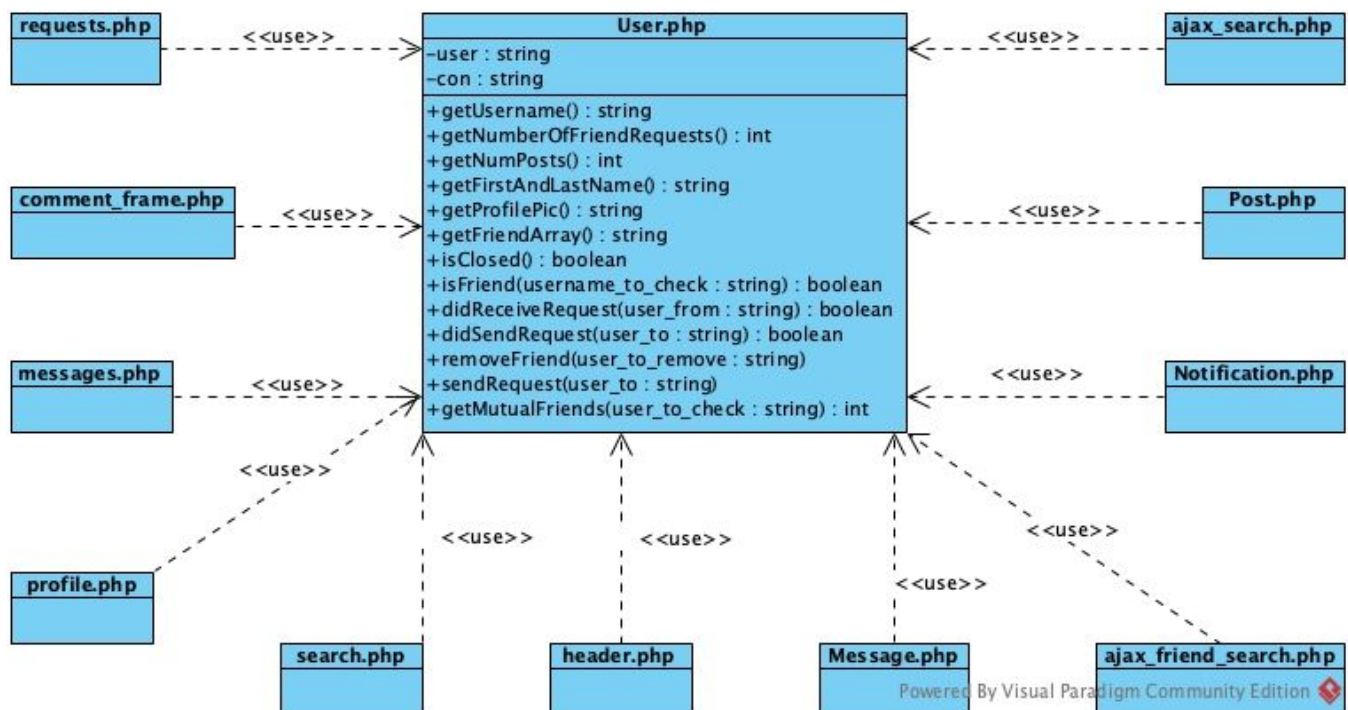
CLASSE	DESCRIZIONE
header.php	Pagina che si occupa dell'header del sito web, comunque a quasi tutte le pagine del sito.

register.php	Pagina che rappresenta la homepage del sito e la parte relativa all'autenticazione/registrazione dell'utente.
utente	Pacchetto che comprende tutte le view per l'utente loggato sulla piattaforma.

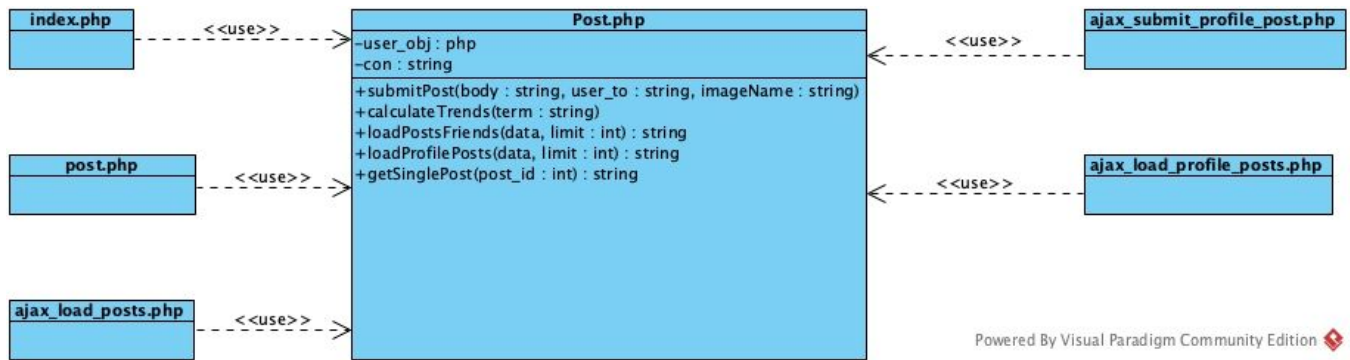
Directory Test

Interfacce delle Classi

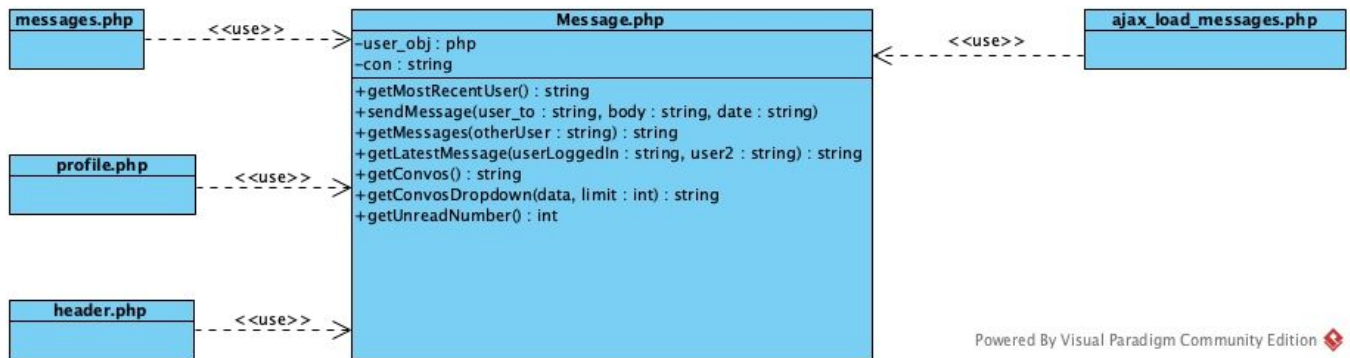
Gestione Account



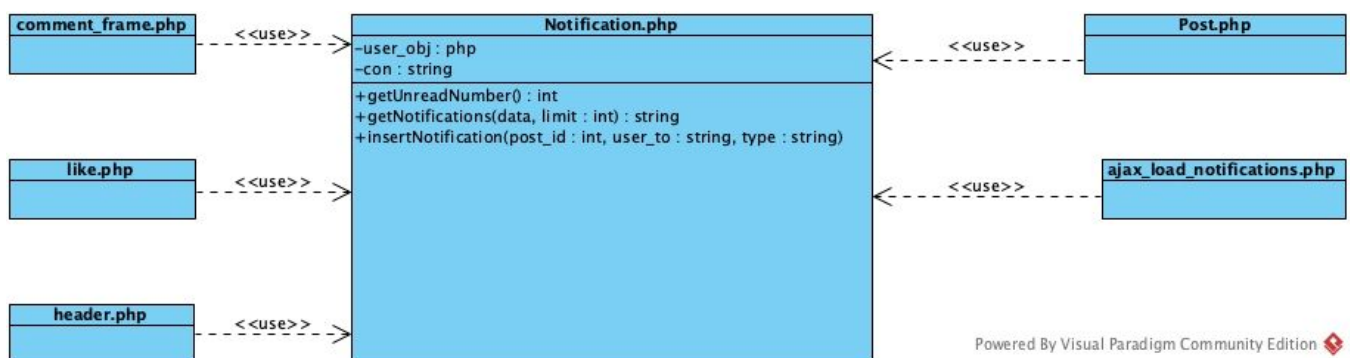
Gestione Posts



Gestione Messaggi



Gestione Notifiche



Glossario

TERMINE	DEFINIZIONE
API	Application Programming Interface Si indica ogni insieme di procedure disponibili al

	programmatore, di solito raggruppate a formare un set di strumenti specifici per l'espletamento di un determinato compito.
MySQLi	(MySQL Improved) è un driver di database relazionale utilizzato nel linguaggio di scripting PHP per fornire un'interfaccia con i database MySQL.
Test Suite	Raccolta di casi di test che devono essere utilizzati per testare un programma software per dimostrare che ha un determinato insieme di comportamenti.