

PANDA ANALYTICS

INFORMATICA ED ELEMENTI DI PROGRAMMAZIONE I
A.A. 2018-2019



CORBIOLI VALENTINA, DE MARCHI
MAURIZIO, LORENZONI GAIA

OBIETTIVO

Lo scopo del progetto assegnatoci verteva sulla realizzazione di una web app. Dopo numerose discussioni, ci siamo posti come obiettivo quello di creare un'app che, dopo aver inserito i dovuti dati, permettesse di simulare il cash flow aziendale. Ciò che abbiamo realizzato, in realtà, simula quello degli anni passati, ma fornisce ugualmente delle previsioni circa l'amministrazione patrimoniale futura dell'azienda.

La web app, infatti, ha il suo fondamento nello stato patrimoniale dell'azienda. Gli utenti possono accertarsi circa l'andamento tramite l'inserimento dei dati degli anni precedenti. Per quanto riguarda queste informazioni, l'utente ha la possibilità di scegliere il valore temporale, quindi se si tratta di transazioni che avverranno nell'anno corrente o nei successivi. Il punto focale è perciò la rappresentazione come grafico per quanto riguarda il cash flow; tuttavia, i dati saranno visualizzati anche sotto forma di tabella, per quanto riguarda fonti e impieghi. Altri dati sono facilmente visualizzabili, come la disponibilità di liquidità, divise per categorie, e le scadenze di pagamenti, dalla più imminente alla più distante. Infine, verrà fornita una previsione per quanto riguarda il periodo più carico e denso per quanto riguarda i pagamenti.

STRUTTURA

Fulcro principale, nonché punto di partenza della app è lo stato patrimoniale dell'utente. Tramite l'inserimento dei dati degli anni passati è possibile visualizzare e controllare l'andamento della propria azienda. Inoltre, per ogni voce, l'utente ha la possibilità di definire il valore temporale e se si tratta di transazioni che avverranno nell'anno corrente o nei successivi. Sono presenti sia un grafico, per quanto riguarda il cash flow, che delle tabelle, per fonti e impieghi, per rappresentare al meglio i dati inseriti. La disponibilità liquida è visualizzata divisa in categorie, mentre le scadente sono ordinate dalla più imminente alla più distante. Infine, viene data una previsione dei periodi in cui saranno più concentrati pagamenti da effettuare.

SEZIONI DI CODICE PARTICOLARI

Creazione utente e autenticazione

Data la natura del progetto, è stata decisione comune il voler puntare sulla sicurezza dell'autenticazione.

Per quanto riguarda la creazione dell'utente si procede nel seguente modo:

- dopo aver inserito i dati, essi verranno controllati per contenuto (rimossi i caratteri di controllo tipo '\n' e vari, rimossi i tag html, gli spazi) e dimensione, nel caso della mail si controlla che sia ben formata;
- dopo aver validato l'input, si passa all' hashing della password: dato che salvarla in chiaro è inadeguato, si provvede a salvare l'hash nel db al suo posto. Tuttavia, anche questa soluzione può essere aggirata ricorrendo a lookup tables o rainbow tables; quindi, si è ricorso al salting: aggiungendo un salt unico, casuale e crittograficamente sicuro alla password prima di effettuare l'hashing si rendono le tecniche di cracking sopracitate inefficaci. Per effettuare l'hashing abbiamo usato una funzione standard (pbkdf2) con algoritmo di hashing SHA3-512 e 1000 iterazioni.

Come metodo di inserimento all'interno del progetto abbiamo deciso di usare il PDO, dato che permette l'utilizzo di prepared statements, i quali prevengono l'SQL INJECTION ed ottimizzano le prestazioni delle query ripetute. Tuttavia, i prepared statements inficiano negativamente sulle prestazioni quando si tratta di query ripetute poche volte in confronto ad una normale query per concatenazione, come ad esempio per la creazione utente. Per questo particolare caso abbiamo hexato con bin2hex i valori e li abbiamo concatenati. Così facendo si previene l'sql injection ed i valori verranno ripristinati in binario all'interno del DB tramite la funzione UNHEX di MySQL.

Per quanto concerne la gestione degli errori, in caso di input malformato viene comunicato all'utente come sistemarlo (come per la password o la mail); in caso di errore del server viene mostrato un errore generico, mentre l'errore più dettagliato viene salvato in un file di log .

Quando verrà effettuato il login invece, la password inserita dall'utente verrà sottoposta allo stesso processo enunciato durante la creazione utente ed i due hash verranno messi a confronto.

Struttura

Per semplificarci la vita con query e funzioni che avremmo dovuto rifare troppe volte, abbiamo adottato una struttura orientata agli oggetti, con una classe per le transazioni, una per gli utenti e una per i flow. Quella per le transazioni è astratta; ha due classi che ereditano da essa e sono quelle per la transazione in data precisa e non. Data la struttura del db, nel metodo della classe astratta, si gestisce la creazione della tupla in TRANSAZIONI. Tramite i metodi astratti che le classi concrete devono implementare, vengono gestite TBOH e TPRECISA.

La struttura del sito, invece, è stata fatta con file separati, mettendo ad esempio le funzioni indispensabili come l'error reporting, l'inizio della sessione e l'home url in un file denominato core.php che viene richiamato in ogni pagina. Il sito modifica dinamicamente la pagina corrente in base all'azione che si vuole compiere (cambiando ovviamente il titolo della pagina per evitare ambiguità e smarrimento), così da avere meno pagine da controllare e rifare da capo. Inoltre, è presente una sezione admin dove è possibile visualizzare, modificare ed eliminare i dati dei vari utenti.

Verifica tramite mail

L'idea iniziale era di mettere una verifica tramite mail per la registrazione dell'utente ed il cambio password. Durante la registrazione, il sito avrebbe richiamato le funzioni presenti su libs/php/utils.php per l'invio mail: partendo da sendEmailViaPhpMail avrebbe inviato una mail all'indirizzo fornito, per verificare il proprio account tramite un token generato dalla funzione getToken. Stesso procedimento per il reset della password tramite access code inviato via mail.

Sfortunatamente, non avendo un provider per il server mail e dovendo trasportare il sito su altermvista, alla fine abbiamo optato per la modifica della password nella sezione impostazioni.

DATABASE

Modello logico

Abbiamo progettato il database nel seguente modo: un utente potrà gestire uno o più flow, i quali però potranno essere gestiti da un solo utente (1->n). Ciascun flow poi conterrà i vari tipi di transazioni, suddivisi in sottoclassi.

Modello relazionale

Data l'assenza di attributi multipli o composti, la trasformazione da logico a relazionale è stata pressoché indolore. Dato l'elevatissimo numero di sottoclassi per ciascuna transazione era impensabile fare una relazione per ognuna di loro; abbiamo quindi optato per fare una sottoclasse per le transazione a data precisa e una per quelle a data imprecisa, discernendo poi la macroarea e la sottoclasse tramite codici.

Realizzazione

Il codice per la macroarea è semplicemente un campo booleano (fonti->0 impieghi->1); per identificare la sottoclasse, invece, abbiamo usato un codice a 3 numeri che identifica la diramazione, che verrà poi convertita in modo leggibile con un'apposita funzione in php. Le transazioni in periodo preciso e incerto sono state tradotte in tabelle contenenti il loro attributo specifico e la chiave esterna che rimanda agli attributi della tabella padre.

Per ottimizzare le query abbiamo inserito un unique-index per ogni tabella; per ottimizzare i procedimenti ricorrenti, invece, abbiamo realizzato delle stored procedures da richiamare al posto delle query. Dato che l'update dei dati è sempre un momento critico, abbiamo inserito all'interno della stored procedure una transazione che in caso di SQLEXCEPTION individuato da un CONTINUE HANDLER effettua un ROLLBACK a prima di effettuare l'update; se non si verificano eccezioni, esegue un COMMIT.

ALGORITMO

Il sito si basa sui dati dello stato patrimoniale degli anni precedenti per calcolare il flow. In particolare frutta due voci: denaro e valori in cassa e depositi bancari e postali che si trovano in Stato patrimoniale -> Impieghi -> Attivo corrente -> disponibilità liquide

Schema stato patrimoniale <https://mm.tt/1252126606?t=GzGFko6WEO>

Sottraendo ai valori di queste due voci dell'ultimo anno i valori dell'anno precedente si ottiene la variazione del cash flow

Esempio con numeri a caso:

	Anno corrente (es. 2019)	Anno precedente (es. 2018)	Variazione
Denaro e valori in cassa	116.000,00	154.800,00	- 38.800,00
Depositi bancari e postali	9.295,00	12.450,00	- 3.115,00
Totale	125.295,00 (f)	167.250,00 (a)	-41.955,00

167.250,00 è la disponibilità monetaria iniziale (a) per il 2019.

125.295,00 è la disponibilità netta finale (f).

La variazione è il cash flow. In questo caso è negativa.

Il motivo per cui usiamo questi due valori è che si tratta della somma di tutte le altre sottovoci e per il calcolo del cash flow sono sufficienti.

Il resto dei dati servono principalmente nella sezione dell'anno corrente, la quale permette di fare previsioni di entrate e uscite monetarie in base ai valori inseriti dall'utente. Per semplicità del db, abbiamo mantenuto la stessa struttura sia per inserire gli stati patrimoniali che i dati per l'anno corrente.

La differenza sta nei valori temporali attribuiti ai due diversi dati, infatti per l'anno corrente l'utente deve attribuire ad ogni cifra numerica una scadenza (entro un anno/più anni, entro tot giorni/mesi, entro la data xx/xx/xxxx oppure nessuna scadenza), mentre per gli stati patrimoniali degli anni precedenti non sono necessarie scadenze ma occorre specificare l'anno a cui si riferiscono i dati.

Questi due tipi di dati portano a due grafici distinti sulla home:

- Il primo mostra l'andamento del cash flow negli anni passati, grazie ai dati degli stati patrimoniali inseriti
- Il secondo mostra le previsioni di soldi in entrata e in uscita, in base al valore temporale inserito accanto a ciascun dato riferito all'anno corrente.

Questo servirà per dare un pronostico chiaro di quali saranno i periodi con maggiori spese. Oltre a questo sarà possibile visualizzare, in una tabella sottostante ai due grafici, le scadenze in ordine dalla più imminente alla più avanti nel tempo (sempre utilizzando i valori temporali dei dati dell'anno corrente).

USE CASE

Nome caso d'uso:	Registrazione utente
Breve descrizione:	Caso d'uso per accedere per la prima volta al sistema tramite registrazione.
Descrizione flusso:	<ul style="list-style-type: none">• Il caso d'uso inizia quando l'utente vorrà registrarsi al sistema.• L'utente deve creare un profilo fornendo email, username e password.• L'utente deve completare il profilo tramite compilazione di un form, con i dati della propria azienda (nome, data di fondazione, se si tratta del primo anno di esercizio).
Post-condizioni:	Le informazioni sono state aggiornate ed è possibile accedere alla home.

Nome caso d'uso:	Visualizzare sulla home debiti a scadenza
Breve descrizione:	Caso d'uso per inserire debiti e visualizzare nella home la loro scadenza.
Pre-condizioni:	L'utente ha già effettuato il login.
Descrizione flusso:	<ul style="list-style-type: none">• Il caso d'uso inizia con l'utente già registrato che effettua il login.• Dalla home, preme su "Aggiungi anno corrente", in cui verrà portato alla sezione "Capitale di debito".• L'utente può inserire la cifra di un debito, in base alla scadenza dello stesso (ex: a breve scadenza, fondi rischi e oneri).• L'utente inserisce quindi il periodo temporale, oppure una data specifica per la scadenza.• L'utente salva.
Post-condizioni:	Nella home compare nella sezione delle transazioni imminenti i debiti che l'utente ha inserito.

Nome caso d'uso:	Inserire stato patrimoniale di una nuova azienda
Breve descrizione:	Caso d'uso per creare un nuovo flow in un profilo già registrato, ed inserire uno stato patrimoniale.
Pre-condizioni:	L'utente ha già effettuato il login.
Descrizione flusso:	<ul style="list-style-type: none"> • Quando l'utente effettua il login, vengono visualizzate le sue aziende (nome, data di fondazione, possibilità di eliminarle). • Per aggiungere un nuovo flow, l'utente preme su "Crea azienda". • L'utente inserisce nome, data di fondazione e disponibilità monetaria iniziale. • Creata la nuova azienda, potrà accedervi premendoci sopra, e sarà portato alla home, in cui sono visualizzati il cash flow e il grafico dell'anno corrente. • L'utente preme allora su "Aggiungi S.P." • L'utente compila allora i campi riguardanti il capitale di debito, capitale proprio, attivo corrente e attivo immobilizzato. • L'utente salva.
Post-condizioni:	Nella sezione "Consulta e modifica S.P." è possibile ritrovare le informazioni aggiunte in precedenza.

Nome caso d'uso:	Visualizzare il flow di un utente da parte dell'admin
Breve descrizione:	Caso d'uso che riguarda l'admin per visualizzare i flow di un altro utente.
Pre-condizioni:	L'utente deve già avere un account registrato come admin nel database.
Descrizione flusso:	<ul style="list-style-type: none"> • L'admin accede alla sezione "Users", dove può visualizzare le informazioni degli altri utenti (username, email, access level, status e flows) • L'admin preme sul link "Flows" • L'admin è ora in grado di visualizzare i flow di quell'utente.
Post-condizioni:	Se all'admin aggrada, può apportare modifiche ai flow, o eliminarli.

STARTING LEVEL

L'idea per il progetto ci ha coinvolti ed affascinati, ma questo ci ha portati, purtroppo, a sottostimare il problema. Il calcolo del cash flow, infatti, è materia di economia aziendale di terzo livello, ben al di fuori dalla nostra portata. Nonostante il nostro impegno, che speriamo abbia portato a qualche risultato, non siamo però riusciti a realizzare tutto al meglio, in base alle nostre aspettative.

PARTI IMPEGNATIVE

Per quanto riguarda gli aspetti che più ci hanno messo alla prova, sicuramente da annoverare c'è tutta la parte di ricerca delle informazioni necessarie per calcolare cash flow e stato patrimoniale.

Inoltre, ci siamo dovuti adoperare con la parte di css, per rendere il sito visivamente ed esteticamente più piacevole, sebbene non sia stata un'impresa semplice. Anche la realizzazione del grafico in Javascript è stata ostica, visto che non venivano segnalati gli errori di accesso alle property.

Infine, altro punto da segnalare concerne lo storico. Per modificarlo, non è possibile usufruire del Get, come nella sezione admin, e bisogna perciò creare dinamicamente un array da passare tramite Post. Tutto ciò è stato piuttosto complesso da realizzare e concepire.