

WEBClicker - Progetto HCI

Marco Loschiavo
Università degli studi di Firenze

Sommario

L'uso sempre più frequente di computer e smartphone ha reso necessario che le applicazioni siano facilmente utilizzabili dall'utente finale. La Human Computer Interaction fornisce importanti metodologie per la loro progettazione finalizzata ad un'efficiente interazione. In questo elaborato è stata realizzata una web application che fornisce un servizio in ambito scolastico. Il lavoro è stato suddiviso in 3 fasi principali: nella prima fase è stato effettuato uno studio preliminare per capire quali fossero i requisiti e le criticità del servizio, poi è stata realizzata l'app ed infine sono stati eseguiti dei test per avere una stima della performance e dell'usabilità.

Il codice del progetto è disponibile su: [Github](#).

Introduzione

WEBClicker è una applicazione web che fornisce un servizio aggiuntivo in ambito scolastico a professori e studenti. L'app offre ai professori, la possibilità di poter creare e proiettare delle domande riguardanti argomenti del corso spiegati durante le lezioni agli studenti, i quali attraverso la rete, utilizzando il proprio smartphone, tablet o pc possano dare una risposta. Alla fine della votazione il professore può mostrare alla classe i risultati, i quali vengono riportati su un grafico. Questo consente agli studenti di verificare di aver ben compreso la lezione

e allo stesso tempo mostra al professore il livello di apprendimento della classe.

Nelle sezioni successive verranno discusse le fasi realizzative dell'applicazione, partendo da un primo processo di Needfinding, dalla creazione di un Mock-up, e per finire alla effettiva implementazione dell'applicazione.

Needfinding

Nella prima fase di sviluppo dell'applicazione è stato seguito un processo di *Needfinding* ricavando le *Personas*, cioè gli utenti che utilizzeranno il nostro applicativo, e gli *Scenari*, ovvero i contesti nei quali le personas utilizzano il prototipo per soddisfare quelle che sono le loro necessità.

Nel nostro caso gli utenti che utilizzeranno il nostro applicativo saranno *Studenti* e *Professori*, i quali sono stati posizionati all'interno di alcuni contesti nell'ottica di ottenere i requisiti per l'applicazione, dunque sono stati costruiti quelli che sono gli scenari contestuali. Gli scenari costruiti sono delle storie nelle quali si descrive in dettaglio come le personas interagiscono con l'applicazione per risolvere i loro task.

Un esempio di scenario è riportato nella lista seguente:

- Il professore, venuto a conoscenza dell'applicazione, decide di iniziare ad utilizzarla, attivando così la sua area personale, nella quale può inserire, modificare, cancellare o visualizzare domande.

- Il professore, prima di iniziare una lezione, si prepara una serie di domande da fare alla classe.
- Il professore, durante la spiegazione, decide di mostrare alcune domande presenti nella sua lista.
- Ogni studente presente in aula, collegandosi all'applicazione con il proprio smartphone o pc risponde a tali domande.
- Il professore, una volta aver ricevuto tutte le risposte, mostra alla classe i risultati.
- Un professore deve avere la possibilità di copiare l'URL delle domande, in modo da poterlo inserire nelle slide che proietterà a lezione.
- Visualizzare la schermata di risposta alla domanda per gli studenti: quest'ultimi senza doversi autenticare, dovranno poter accedere ad una specifica pagina web dell'applicazione, in modo da poter rispondere alla domanda

Infine grazie alle descrizioni degli scenari sono stati ricavati i *Requisiti* dell'applicazione, ovvero quello che il prototipo deve includere per soddisfare le necessità degli utenti.

In dettaglio sono stati estratti i requisiti principali, i quali sono stati riportati nella lista seguente:

- Visualizzare la schermata di login dell'applicazione: Il professore deve poter inserire Username e Password per essere autenticato e reindirizzato nella propria pagina iniziale.
- Visualizzare la propria pagina personale, la quale mostrerà una lista con tutte le domande create. Dovrà essere possibile inserire nuove domande, ed eliminare e modificare quelle già presenti.
- Dalla propria pagina personale, il professore deve poter selezionare una domanda da lui creata e mostrarla alla classe, così da avviare il processo di votazione.

- Visualizzare i risultati delle risposte: Il professore deve poter mostrare alla classe l'esito della votazione.

Mockup e Diagramma di Flusso

Nella fase che ha preceduto l'implementazione sono stati realizzati dei mockup che descrivono le viste dell'applicazione, in modo da avere una riproduzione dell'interfaccia finale. Un mockup è una rappresentazione statica del prodotto, realizzabile utilizzando diverse tecniche. Nel nostro caso è stata utilizzata la paper prototyping. In Figura 1 sono riportati i mockup dell'applicazione.

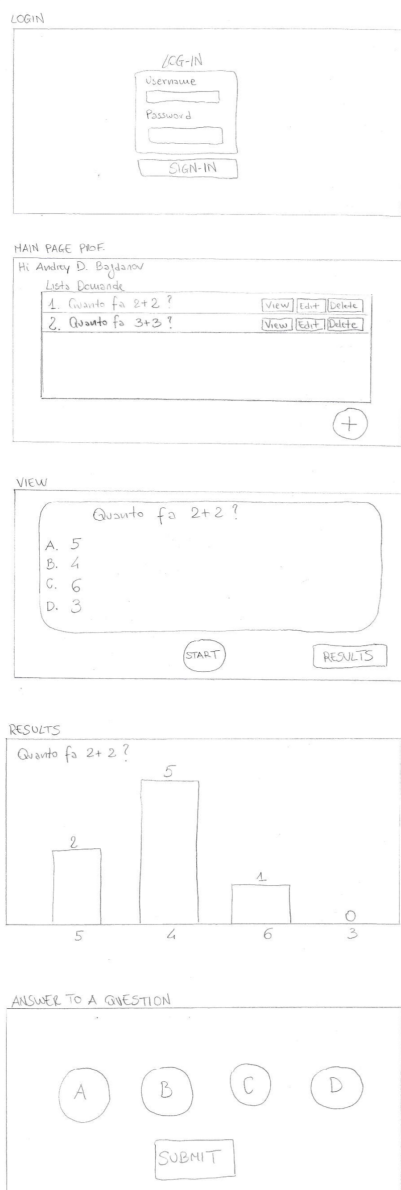


Figura 1: Mockup

Inoltre, si è reso opportuno un diagramma di flusso in modo da chiarire la sequenza del programma (vedi Figura 2).

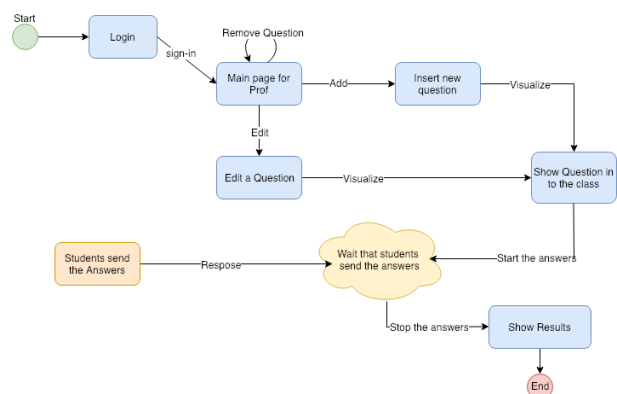


Figura 2: Diagramma di Flusso

Applicazione

Una volta estratti i requisiti dalla fase di needfinding, il lavoro si è concentrato sulla fase realizzativa dell'app.

Sono stati reliazzati sia il lato client, utilizzando tecnologie come HTML5, CSS, e JavaScript, oltre ai framework Bootstrap[1] e jQuery[2], sia il lato server adoperando il linguaggio PHP[3] per manipolare i dati e per interrogare il nostro Database.

Implementazione

Il principale strumento utilizzato per lo sviluppo dell'applicazione è stato PHPStorm, un ambiente di sviluppo integrato (IDE) sviluppato dalla JetBrains, che fornisce numerosi strumenti per una vasta gamma di linguaggi di programmazione. L'implementazione è strutturata secondo il pattern architetturale MVC (Model View Controller) così da rispettare il principio della *separation of concern*. Questo permette di suddividere un'applicazione interattiva in tre componenti:

- Model: contiene i dati principali e le relative funzionalità.
- View: è come i dati sono rappresentati all'interno dell'app.
- Controller: gestisce l'input e fa da intermediario tra vista e modello.

Possiamo pensare alle interfacce grafiche (HTML, CSS) come le viste della nostra applicazione, mentre come modello tutta la parte inerente la gestione dei dati e del database. Il controller invece è la parte della nostra app che gestisce le interazioni dell'utente, ed in base a queste va a modificare il modello. Un'esempio di ciò potrebbe essere un utente che preme il pulsante di Reset (la vista permette di visualizzare all'utente tale bottone). A questo punto il compito viene passato al controller, nel nostro caso implementato utilizzando codice Javascript, il quale mediante tecnologia AJAX[4], va ad effettuare operazioni sul Database coinvolgendo il modello.

Interfaccia e Funzionalità

Recandoci alla pagina iniziale dell'applicazione, viene mostrata la sezione di Login (Figura 3), nella quale un docente, immettendo i propri dati personali, viene reindirizzato nella propria pagina personale (Figura 4).

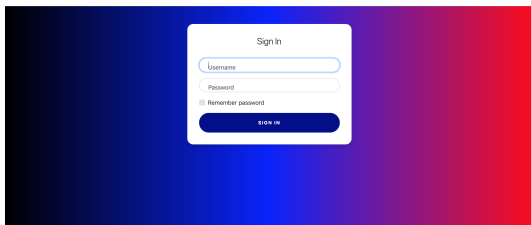


Figura 3: Interfaccia di Login

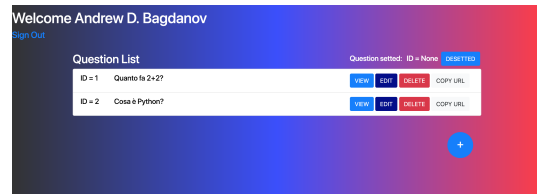


Figura 4: Interfaccia principale del docente

Adesso il professore cliccando sul bottone $+$ potrà inserire una nuova domanda che andrà ad aggiungersi alla lista di quelle create precedentemente. Sarà consentito, tramite i due bottoni $+$ e $-$, inserire domande contenenti un minimo di 2 ed un massimo di 5 possibili risposte. L'interfaccia che consente di creare una nuova domanda è mostrata in Figura 5.

L'applicazione offre anche la possibilità di rimuovere e modificare delle domande presenti nella lista, mediante i bottoni *delete* e *edit*. L'interfaccia che permette di modificare un domanda è la stessa di quella in Figura 5.

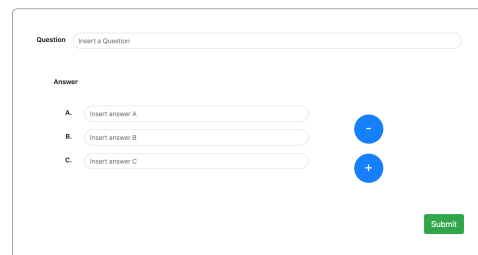


Figura 5: Interfaccia di inserimento nuova domanda

Premendo il pulsante *view* relativo alla domanda di nostro interesse, essa, con le relative risposte, viene impaginata in modo (facendo uso di un proiettore) da poter essere mostrata alla classe (Figura 6). Da qui sarà possibile, attraverso i bottoni di *start*

e di *stop* far iniziare e finire il sondaggio, e attraverso il pulsante *results* mostrare i risultati.

Figura 6: Interfaccia di presentazione domanda

Il docente dovrà fornire l'URL che consentirà agli studenti di poter rispondere a tale domanda. L'URL sarà sempre la stessa, quindi non sarà necessario fornirla ogni volta.

Gli studenti tramite smartphone o pc potranno a questo punto rispondere alla domanda posta. Essi visualizzeranno un'interfaccia come quella riportata in Figura 7.

Figura 7: Interfaccia di risposta alla domanda

Non appena tutti gli studenti hanno risposto al quesito, il docente può mostrare alla classe i risultati (Figura 8)

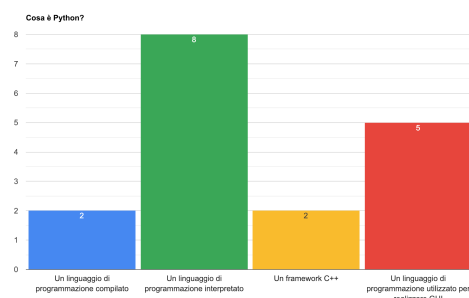


Figura 8: Grafico dei risultati mostrati alla classe

Usability Test

Durante la fase finale del progetto è stato svolto un usability test, per capire quanto il prototipo costruito fosse user-friendly per gli utenti del servizio. Dunque l'obiettivo del test era quello di riuscire a valutare alcuni aspetti dell'applicazione come la sua intuitività, la sua facilità d'uso e il suo aspetto.

I candidati sono stati acquisiti in parte all'interno del dipartimento MICC dell'università di Firenze ed in parte all'interno dell'istituto Santa Marta. Per motivi pratici, il test è stato fatto svolgere solamente agli studenti (ad eccezione di uno), chiedendo ad una parte di loro di impersonificarsi solo nella figura di docente. In totale le persone partecipanti al test sono state 14, divise in 7 professori e 7 studenti, ai quali sono stati assegnati alcuni task.

Per quanto riguarda i docenti sono stati assegnati i seguenti task:

- Accede alla loro pagina personale inserendo le credenziali fornite
- Creare una serie di domande (almeno 2)

- Modificare una domanda precedentemente creata
- Cancellare una domanda precedentemente creata
- Far visualizzare alla classe la domanda e consentire agli studenti di poter rispondere
- Mostrare i risultati

Mentre per quanto riguarda gli studenti sono stati assegnati i seguenti task:

- Visualizzare la domanda proposta dal docente
- Rispondere a tale domanda collegandosi all'indirizzo fornito
- Visualizzare i risultati al test

Ad ogni candidato, una volta eseguito il test, è stato chiesto di compilare un questionario con delle domande specifico a seconda del ruolo assunto dal tester. Sostanzialmente sono stati creati due diversi questionari, uno da far compilare al docente e una allo studente.

Le domande sottoposte sono di tipo SEQ con una scala che va da 1 a 7 dove 1 rappresenta fortemente disaccordo e 7 fortemente d'accordo; per richiamare l'attenzione del candidato alcune domande sono poste in modo inverso. In tabella 1 e 2 è riportato un quadro generale dei risultati ottenuti dal test.

I risultati ottenuti mostrano un buon funzionamento ed una buona facilità d'uso dell'applicazione. Dai dati emerge che il 57% degli studenti avrebbero voluto nell'interfaccia di risposta anche le possibili opzioni. Ciò inizialmente non fu stato fatto, in quanto ritenuto dopo una prima analisi essere ridondante con le opzioni di risposta proiettate. Viene invece reputato inutile ripetere

anche la domanda come si può vedere dalla N.6 del questionario posto agli studenti. Per quanto riguarda la parte dell'app dedicata al docente sono emersi problemi per quanto riguarda la navigabilità dell'applicazione, risultata non intuitiva, in quanto non presenti pulsanti di back e di ritorno alla home page. Il 71% dei docenti ha reputata la pagina principale piacevole, anche se la valutazione media non è stata eccessivamente elevata.

Tra i suggerimenti rilasciati dai candidati sono stati trovati interessanti i seguenti:

- Includere nella parte di applicazione dedicata al professore, una barra di navigazione o dei pulsanti di *Back* e di *ritorno alla home page* per un maggior controllo delle pagine.
- Effettuare alcune migliorie per quanto riguarda la parte di Design.
- Includere una funzionalità che consenta di poter inserire anche un'immagine a fianco della domanda.

N.	Domanda	Val. Media	%. d'accordo
1	L'interfaccia mostrata dopo il login, è risultata essere chiara	6.57	100
2	Inoltre è risultata gradevole	4.29	71
3	È stato difficile e poco intuitivo inserire una domanda	1.29	0
4	La gestione delle domande (modifiche, cancellazioni) è risultata efficiente	7	100
5	È stato intuibile far visualizzare agli studenti la domanda, permettendo loro di rispondere	6	100
6	È stato difficile mostrare i risultati ottenuti dal sondaggio agli studenti	2.43	14
7	Trova chiara la navigabilità dell'applicazione	3.14	43
8	Il servizio fornito risulta esauriente	5.71	100
9	In generale sono soddisfatto dell'app	5.71	100
10	Pensi che l' applicazione possa risultare utile	5.85	100

Tabella 1: Domande poste ai docenti, con le rispettive valutazioni. Ogni risposta va in una scala da 1 (fortemente disaccordo) a 7 (fortemente d'accordo). Percentuale d'accordo: (%) votazioni maggiori di 4.

N.	Domanda	Val. Media	%. d'accordo
1	L' interfaccia inerente la proiezione di domande risulta essere efficace alla loro comprensione	6.57	100
2	Inoltre è risultata gradevole	5.71	100
3	Il grafico dei risulta essere comprensibile	6.14	100
4	È stato difficile rispondere ad una domanda	2.29	0
5	Nell' interfaccia di risposta avresti preferito che ci fossero state riportate anche le possibili risposte	4.71	57
6	Avresti preferito ci fosse stata riportata anche la domanda	3.28	14
7	Il servizio fornito risulta esauriente	5.71	100
8	In generale sono soddisfatto dell'app	5.71	100
9	Pensi che l' applicazione possa risultare utile	5.43	100

Tabella 2: Domande poste agli studenti, con le rispettive valutazioni. Ogni risposta va in una scala da 1 (fortemente disaccordo) a 7 (fortemente d'accordo). Percentuale d'accordo: (%) votazioni maggiori di 4.

Conclusioni

I suggerimenti presentati nella sezione precedente gettano le basi per gli sviluppi futuri, andando così a migliorare ed ad arricchire le funzionalità.

In conclusione, anche se durante la fase di usability test, sono emerse alcune criticità, queste non hanno gravato sul prodotto finale, il quale è stato apprezzato e reputato utile sia per gli studenti che per i docenti. Studenti e professori avranno quindi un ulteriore strumento per poter svolgere al meglio il proprio lavoro.

Riferimenti bibliografici

- [1] Bootstrap - open source toolkit for developing with HTML, CSS, and JS - <https://getbootstrap.com/>
- [2] jQuery - a JavaScript library - <https://jquery.com/>
- [3] Michele E. Davis, Jon A. Phillips. Programming in PHP & MySQL
- [4] Ajax: Jesse James Garrett. A New Approach to Web Applications <https://www.adaptivepath.org/ideas/ajax-new-approach-web-applications/>