

Valido per la prima sessione regolare 2018

| | |
|--|--|
| Condizioni | |
| Valido solo per la prima sessione regolare 2018. Svolgimento: Visione progetto (domande e/o modifiche al codice) + domande generali sugli argomenti del corso. Studenti dal 2015/16 in poi: esame congiunto con Ingegneria del Software, oppure gruppi 2/3 studenti o individualmente. Traccia proposta dagli studenti e discussa con il docente, oppure quella descritta sotto. | |
| Requisiti tecnici principali | |
| <ul style="list-style-type: none">• Lato client l'applicazione utilizza HTML5, CSS, e Javascript/JQuery.• Lato server l'applicazione utilizza Java Servlet+JSP in modalita' MVC, e JDBC per il database.• L'applicazione utilizza il piu' possibile Ajax per la comunicazione di dati e updates tra client e server. E' fortemente consigliato l'uso di JSON per il trasferimento dei dati.• L'applicazione deve offrire una interfaccia grafica responsive (per esempio attraverso il framework Bootstrap o simili oppure fatto a mano)• L'applicazione deve integrare contenuto "interessante" tramite almeno una API esterna (per esempio Google Maps, Youtube, Twitter, API per Film, Flickr, ecc.). Ad esempio, non basta integrare in modo statico una mappa di Google che mostra un indirizzo preconfigurato. | |
| Traccia proposta dal docente. | |
| Si vuole realizzare un sistema informativo avanzato che offra agli utenti possibilità' di creare e gestire campionati di Fantacalcio. Si dia un nome a piacimento all'applicazione realizzata. | |
| Termini e modalita' di consegna | |
| Oltre a registrarsi su ESSE3, occorre compilare al piu' presto il seguente form | |

| | |
|---|--|
| <p>https://docs.google.com/forms/d/14yoFaVBjIiedGh9Mf-pS6_RtqCmKuA0iwh-OTFxV1Fs/edit con i nominativi del gruppo e l'appello in cui si prevede di discutere (queste scelte non sono vincolanti, ma solo per organizzarmi con i tempi).</p> <p>Il progetto dovrà essere consegnato ENTRO DUE GIORNI prima dell'appello di riferimento via GitHub (chiedere al Dott. Reale per un nuovo repository), per esempio se l'esame è giorno 21, la consegna va effettuata ENTRO giorno 19. Occorre mandare una mail (giovanni.grasso@unical.it e reale@mat.unical.it) con il link a repository git. Il progetto può comunque essere completato fino al giorno dell'esame e discusso poi dal vivo. Il codice consegnato deve contenere tutto il necessario per essere eseguibile, ed il database deve essere accessibile in remoto. Includere un Readme file con le istruzioni ed eventuali note.</p> <p>In aggiunta, occorre fornire un numero non superiore a 10 slides powerpoint (o pagine google docs) che descriva SOMMARIAMENTE l'applicazione realizzata, meglio usando UML class diagrams per le classi principali, e specificare quali dei casi d'uso sono supportati e non, eventuali bugs, lo schema delle entità e del database.</p> <p>Sottolineate le funzionalità aggiuntive implementate ed in generale ciò che pensate sia la parte migliore dell'applicazione, i framework usati, ecc.</p> | |
|---|--|

Progetto di SIW - Prima sessione 2017 e appello straordinario Aprile 2017

| | |
|--|--|
| Condizioni | |
| <p>Valido per la prima sessione 2017, per gli studenti che non faranno il progetto congiunto con il Prof. Ricca. Valido anche per l'appello straordinario di Aprile 2017 Individuale o in gruppi 2/3 studenti. Occorre comunicarmi i gruppi al più presto</p> <p>Svolgimento: Visione progetto (domande e/o modifiche al codice) + domande generali sugli argomenti del corso.</p> | |

Descrizione

Si vuole realizzare un sistema informativo avanzato per la gestione di aste online (tipo Ebay). Si dia un nome a piacimento all'applicazione realizzata.

Requisiti Funzionali e Utenti del sistema

- Le funzionalita' minime sono quelle di base funzionalita' di base di Ebay (per esempio Asta, Vendilo Subito). Mi aspetto, e verranno valutate positivamente, funzionalita' aggiuntive di un certo interesse.

Requisiti Tecnici

- Lato client l'applicazione utilizza HTML5, CSS, e Javascript/JQuery.
- Lato server l'applicazione utilizza Java Servlet+JSP in modalita' MVC, e JDBC per il database.
- L'applicazione utilizza il piu' possibile Ajax per la comunicazione di dati e updates tra client e server. E' fortemente consigliato l'uso di JSON per il trasferimento dei dati.
- L'applicazione deve offrire una interfaccia grafica responsive (per esempio attraverso il framework Bootstrap o simili oppure fatto a mano)
- L'applicazione deve integrare contenuto "interessante" tramite almeno una API esterna (per esempio Google Maps, Youtube, Twitter, API per Film, Flickr, ecc.). Ad esempio, non basta integrare in modo statico una mappa di Google che mostra un indirizzo preconfigurato.

Termini e modalita' di consegna

Oltre a registrarsi su Uniwex, occorre compilare al piu' presto il seguente form <https://goo.gl/forms/pJ8SJMA5Dsk4ULnV2> con i nominativi del gruppo e l'appello in cui si prevede di discutere (queste scelte non sono vincolanti, ma solo per organizzarmi con i tempi).

Il progetto dovra' essere consegnato solitamente due giorni prima dell'appello di riferimento via dropbox/gdrive/altro. Vi verra' mandato un avviso con il termine di consegna.

Il progetto può comunque essere completato fino al giorno dell'esame e discusso poi dal vivo. Il file consegnato deve contenere tutto il codice per essere eseguibile, ad esempio esportarlo come progetto eclipse. Il database deve essere accessibile online. Includere un Readme file con le istruzioni ed eventuali note.

In aggiunta, occorre fornire un numero non superiore a 10 slides powerpoint (o pagine google docs) che descriva SOMMARIAMENTE l'applicazione realizzata, meglio usando UML class diagrams per le classi principali, specifichi quale dei casi d'uso sono supportati e non, eventuali bugs, lo schema delle entità e del database.

Sottolineate le funzionalità aggiuntive implementate ed in generale ciò che pensate sia la parte migliore dell'applicazione, i framework usati, ecc.

Progetto di SIW - Giugno/Luglio/Settembre 2016

| Condizioni | |
|--|--|
| Valido per Giugno Luglio e Settembre 2016. In caso di insuccesso ad uno degli appelli, verrà assegnato un nuovo progetto per l'appello successivo. Gruppi 2/3 studenti se frequentanti Svolto individualmente se non frequentanti Nessun limite di voto Svolgimento: Visione progetto (domande e/o modifiche al codice) + domande generali sugli argomenti del corso. | |

| | |
|--|--|
| | |
|--|--|

Descrizione

Si vuole realizzare un sistema informativo avanzato per la vendita dei biglietti per eventi di vario genere. Si dia un nome a piacimento all'applicazione realizzata.

Principali Entita' in Gioco

Un Evento e' associato ad una categoria, per esempio, Sport, Concerti, Mostre, Teatro, Cinema, Altro, e possibilmente sottocategorie (*Sport>Calcio,Tennis Concerti>Rock,Pop Cinema>Azione*). Prendere spunto da un sito reale di ticketing.

Ogni Evento e' associato ad una serie di informazioni descrittive quali il luogo, la data di una certa edizione dell'evento (in caso di eventi ricorrenti come cinema, concerti), l'artista/squadra/film, informazioni generiche dell'evento ed altro come necessario per i casi d'uso specificati in seguito.

Per ogni evento esistono una o piu' tipologie di biglietti (per esempio palco/platea, tribuna superiore, prime file) che differiscono per prezzo.

Funzionalita' e Utenti del sistema

L'utente occasionale del sito deve essere in grado, tra l'altro, di:

- Ricercare eventi per categoria, data, luogo, artista, prezzo.
- Selezionare eventi aggiungendoli nel carrello
- Modificare il carrello corrente
- Finalizzare l'ordine, registrandosi al sito (o facendo login) e pagando (per il pagamento si assuma sara' fatto offline).

Il sistema prevedere due tipologie di utenti registrati:

- Cliente
- Organizzatore

Il sistema permette all'organizzatore, tra l'altro, di:

- Creare/modificare/cancellare un evento,
- Mettere in vendita/sospendere la vendita dei biglietti per un evento,

- Stabilire/modificare i prezzi per ogni tipologia di biglietto,
- Consultare lo stato corrente delle vendite di un evento in corso,
- Consultare lo storico per ogni evento organizzato in passato,
- Altro (e' possibile introdurre altre funzioni).

Un cliente puo', tra le altre cose:

- Creare un ordine di biglietti per uno o piu' eventi,
- Finalizzare e pagare per un ordine
 - (per semplicita' si assuma che i pagamenti saranno fatti offline)
- Consultare lo storico dei suoi acquisti

Requisiti Tecnici

- Lato client l'applicazione utilizza HTML5, CSS, e Javascript/JQuery.
- Lato server l'applicazione utilizza Java Servlet+JSP in modalita' MVC, e JDBC per il database.
- L'applicazione utilizza il piu' possibile Ajax per la comunicazione di dati e updates tra client e server. E' fortemente consigliato l'uso di JSON per il trasferimento dei dati.
- L'applicazione deve offrire una interfaccia grafica responsive (per esempio attraverso il framework Bootstrap)
- L'applicazione deve integrare contenuto "interessante" tramite almeno una API esterna (per esempio Google Maps, Youtube, Twitter, API per Film, Flickr, ecc.).

Requisiti Funzionali

- Il sistema fornisce le funzionalita' descritte sopra, supportando eventi per un numero minimo di 3 categorie di eventi diversi (e relative sottocategorie), per esempio {Sport>Calcio, Tennis} {Teatro>Opera,Balletto} {Cinema>Animazione, Drammatico}
- Un cliente puo' recensire un evento a cui ha partecipato. Gli eventi sono percio' associati ad una lista di recensioni ed un feedback (tipo amazon).
- Un cliente puo' creare una o piu' "liste dei desideri", su cui salvare gli eventi di interesse per acquisti futuri per esempio.
- Un cliente puo' rimettere in vendita un biglietto acquistato, ad un prezzo non superiore a quello di acquisto. I soldi incassati dalla vendita rimangono a suo credito nel sistema per l'acquisto di altri biglietti in futuro.
- Bonus:

- Ogni utente puo' condividere una lista dei desideri con altri utenti del sistema (realizzando una cosa tipo lista-nozze) e gli altri utenti possono comprare eventi per lui
- Saranno considerati un plus ogni funzionalita' aggiuntive di una certa rilevanza, per esempio la possibilita' di selezionare i posti disponibili per un evento in modalita' grafica, oppure l'utilizzo di piu' API web, o utilizzo delle API di geolocation per localizzare la posizione dell'utente e mostrare gli eventi in prossimita'.

Termini e modalita' di consegna

Oltre a registrarsi su Uniwex, occorre compilare al piu' presto il seguente form

<http://goo.gl/forms/qtzOEEt8yV28PyQT2> con i nominativi del gruppo e l'appello in cui si prevede di discutere (queste scelte non sono vincolanti, ma solo per organizzarmi con i tempi).

Il progetto dovra' essere consegnato solitamente due giorni prima dell'appello di riferimento via email/dropbox/altro. Vi verra' mandato un avviso con il termine di consegna.

Il progetto puo' comunque essere completato fino al giorno dell'esame e discusso poi dal vivo. Il file consegnato deve contenere tutto il codice per essere eseguibile, ad esempio esportarlo come progetto eclipse. Il database deve essere accessibile online. Includere un Readme file con le istruzioni ed eventuali note.

In aggiunta, occorre fornire un numero non superiore a 10 slides powerpoint (o pagine google docs) che descriva SOMMARIAMENTE l'applicazione realizzata, meglio usando UML class diagrams per le classi principali, specifichi quale dei casi d'uso sono supportati e non, eventuali bugs, lo schema delle entita' e del database.

Sottolineate le funzionalita' aggiuntive implementate ed in generale cio' che pensate sia la parte migliore dell'applicazione, i framework usati, ecc.