

Tema Di Progetto 1

Descrizione

Acquisto On Line

Applicazione Web-based per la simulazione di un *flusso di acquisto* di un prodotto on-line. Deve essere generata una (o più) pagine di catalogo con *contenuto dinamico*, ovvero la lista dei prodotti della pagina deve provenire da una tabella. Ogni prodotto deve essere identificato da una serie di informazioni che si ritengono significative. Dalla pagina di catalogo è possibile, eventualmente, accedere ad una pagina di dettaglio prodotto con informazioni estese riguardanti il suddetto. La selezione di un elemento del catalogo deve aggiornare un carrello, realizzato tramite cookies, la cui visualizzazione avviene su una pagina appositamente realizzata dalla quale sia possibile anche eliminare un elemento precedentemente selezionato. E' opzionale la possibilità di modificare la quantità di elementi desiderati per ogni prodotto. Un Carrello può, eventualmente, essere salvato e recuperato in una sessione differente. L'insieme dei prodotti di un carrello costituisce un'ordine. L'utente può confermare l'ordine corrente che deve essere memorizzato su db. Una pagina di gestione ordini deve permettere di poter recuperare la lista degli ordini confermati ed eventualmente di annullarli.

Project 1

Macro Attività

•Acquisto On Line

- Login (opzionale) → mono/multi utente.
- Pagina di catalogo
 - Lista item dinamica (o statica)
 - Dettaglio prodotto (opzionale)
- Carrello (session management: server | client)
 - Selezione della quantità (opzionale)
 - Eliminazione di un prodotto dal carrello.
 - Salvataggio / recupero del carrello (opzionale)
- Pagina di conferma ordine
- Pagina lista ordini confermati (ozionale)
 - Annullamento ordine (opzionale)

Services

funzionalità di accesso al db

Blogics

user management (opt)
catalog management
basket management
order management

Bflows (beans)

login
catalogo
basket
gestione ordini

Presentation level (jsp)

lista prodotti
dettaglio prodotto
carrello
conferma ordine
lista ordini effettuati

Tema Di Progetto 2

Descrizione

Web Mail

Gestore di posta online. La ricezione di nuove email viene simulata tramite l'inserimento in una tabella (manualmente) delle informazioni tipiche di una email: from, to, subject, body. L'applicazione permette di leggere la posta in arrivo, discriminando quali mail sono già state lette da quelle non ancora lette. Ogni Email può essere eliminata. E' possibile scrivere nuove email, delle quale rimane traccia come posta inviata. Deve essere possibile selezionare in quale funzionalità accedere da una pagina principale o da navbar tra: leggi posta in arrivo, posta eliminata, posta inviata, invia messaggio.

POSTA IN ARRIVO: lista di tutte le email (from,to,subject,letto:si|no) arrivate (ma non cancellate). Selezionando una di queste mail si visualizza il testo del messaggio (il che equivale a marcarla come email letta). Sempre da questa pagina si può cancellare una mail.

POSTA ELIMINATA: lista delle mail (from, to, subject) eliminate dalla posta in arrivo. Opzionale la possibilità di cancellarla definitivamente o di recuperarla (visualizzabile in posta in arrivo).

NUOVO MESSAGGIO: si edita un nuovo messaggio di posta elettronica che può essere visualizzato in posta inviata.

POSTA INVIATA: lista delle mail (from, to, subject) inviate, cioè scritte dall'utente.

Project 2

Macro Attività

•Web Mail

- Login (opzionale) → mono/multi utente.
- Posta in arrivo
 - Lista dinamica delle mail in arrivo
 - Discriminare (con un colore) la posta letta da quella nuova
 - Il body può essere visualizzato nella stessa pagina o in una di dettaglio
 - Una mail nuova deve poter passare dallo stato di NUOVA LETTA o ELIMINATA
- Posta eliminata
 - Lista delle mail nello stato ELIMINATA
 - Opzionale: cambio di stato (NUOVA|LETTA)
- Posta inviata
 - Mail nello stato INVIATA

Services

funzionalità di accesso al db

Blogics

mail management

Bflows (beans)

mail

Presentation level (jsp)

home page

Inmail

Delmail

Outmail

Tema Di Progetto 3

Descrizione

Web Source Safe

Applicazione web-based per la gestione concorrente del contenuto di un insieme di documenti. L'applicazione permette a più utenti di modificarne il contenuto (salvato su db). Un utente può modificare un documento solamente dopo averlo "estratto", azione che lo rende unico proprietario del documento fino a quando non lo "archivia". Solo dopo l'archiviazione un altro utente può prendere il possesso dello stesso documento. L'archiviazione comporta anche l'aggiornamento del contenuto del documento. E' necessaria un flusso di autenticazione degli utenti (login e password) che determina l'inizio di una sessione di lavoro. Deve essere realizzata una pagina che dia la lista di tutti i documenti disponibili e del loro stato (libero, estratto da). Possibilità di effettuare un'operazione di estrazione su un file nello stato di libero (non estratto). Tale azione porta ad una pagina dove è possibile modificarne il contenuto. Il termine della modifica comporta l'archiviazione del documento con relativo aggiornamento del contenuto su db.

Sono possibili alcune migliorie:

- History delle modifiche con possibilità di ritornare a versioni precedenti
- Estrazione multipla di file da parte dello stesso utente
- Annullamento dell'estrazione con ritorno alla versione precedente
- Modifica e archiviazione temporalmente distinti

Project 3

Macro Attività

•Web Source Safe

- Login (session management)
- Lista documenti disponibili con relativo stato
 - Dettaglio dell'history del documento (opzionale)
 - Possibilità di ritornare ad una qualunque versione dell'history (opzionale)
- Estrazione di un documento libero
 - Pagina di modifica del documento (textarea)
 - Conferma della modifica e relativa ARCHIVIAZIONE

Services

funzionalità di accesso al db
session management

Blogics

Bflows (beans)

Presentation level (jsp)

Tema Di Progetto (risolto)

Descrizione

Agenda

Applicazione Web-based per la gestione di un agenda multiutente. Solamente gli utenti autenticati possono accedere alle funzionalità dell'agenda che devono essere le seguenti:

- Inserimento nuovo contatto

- Memorizza su db un nuovo contatto, facendo attenzione a eventuali duplicati. Una possibile chiave logica di verifica potrebbe essere la terna NOME COGNOME EMAIL.

- Modifica

- Deve essere possibile aggiornare i dati di un record inserito con la funzionalità descritta al punto 1.

- Elimina

- Cerca

- Possibilità di ricercare i contatti in modalità multiple (nome, cognome, ...)

I contatti inseriti da un utente sono privati, nel senso che nessun utente può accedere alle informazioni inserite da un altro utente in nessuna modalità.

Project

Macro Attività

•Agenda

- Login → inizio sessione
- Inserimento contatto
- Trova
- Modifica
- elimina

Services

funzionalità di accesso al db
sessione

Blogics

user management
person management

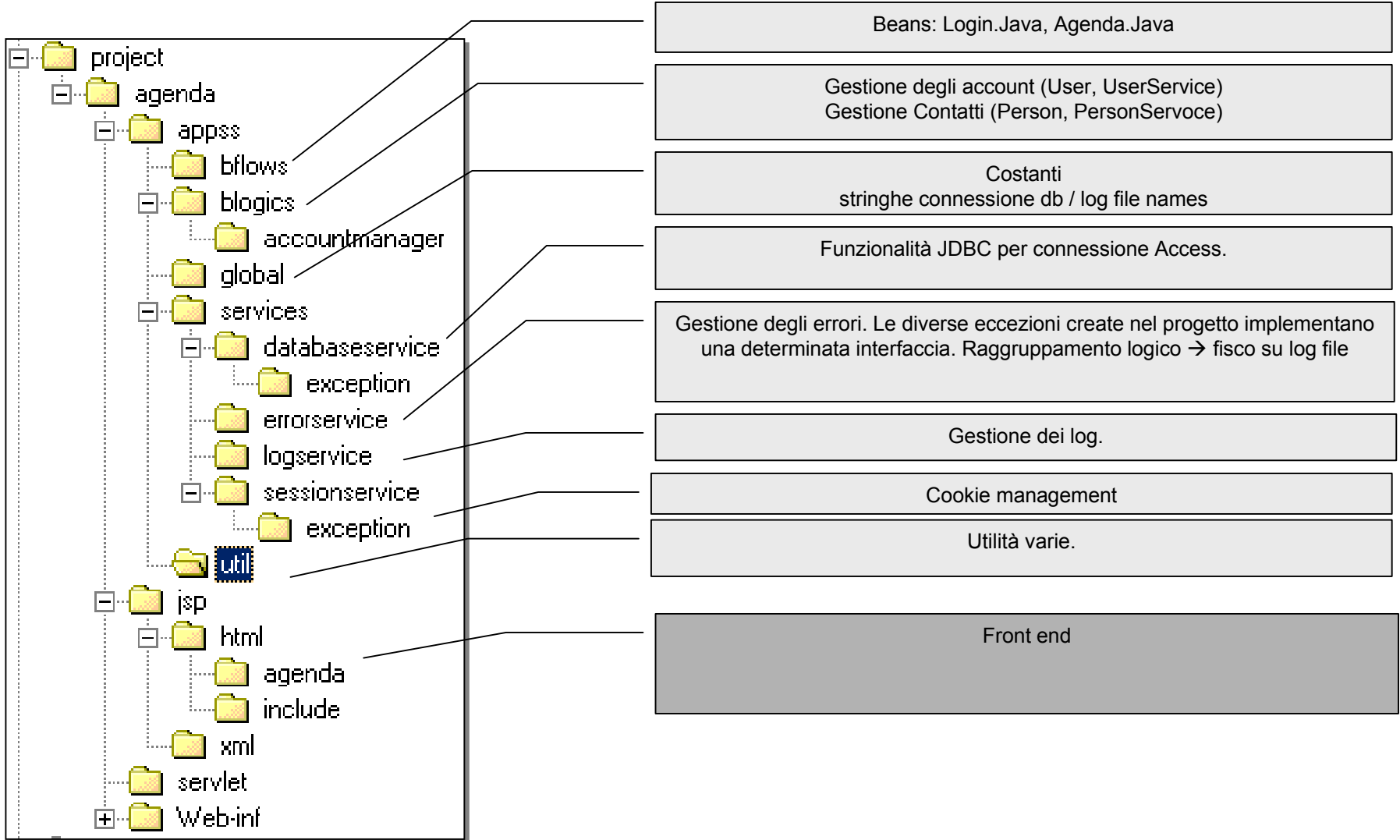
Bflows (beans)

login
agenda

Presentation level (jsp)

login
aggiunta
modifica
elimina
trova

Struttura Applicazione



Scheda Di Valutazione

Sono ammessi progetti sviluppati fino a un massimo di 2 persone. Il numero di sviluppatori per progetto deve essere direttamente proporzionale alla complessità dello stesso. La scheda seguente riassume i criteri di valutazione.

Documentazione			Sviluppo		
<i>Analisi funzionale</i> (WHAT)	<i>Analisi tecnica</i> (HOW)	<i>JavaDoc</i>	<i>Funzionalità Realizzate</i>	<i>Code Style</i>	<i>Usabilità e Grafica</i>
Analisi delle problematiche. Individuazione di come le varie funzionalità devono operare singolarmente e tra loro. Possibili sviluppi futuri.	Analisi delle soluzioni adottate. Caratteristiche tecniche utilizzate. Diagrammi E-R. Architettura dell'applicazione.	Documentazione prodotta tramite JavaDoc. Rispecchia la capacità di commento del codice.	L'applicazione realizzata deve essere funzionante. Possono essere utilizzate tutte le tecnologie viste durante il corso. Nessuna limitazione.	Il codice deve essere commentato, ben leggibile e presentare caratteristiche di ri-usabilità	L'applicazione deve avere caratteristiche di buona usabilità. L'appeal grafico non è un aspetto secondario nello sviluppo di applicazioni.