UNIVERSITA' DEGLI STUDI DI PERUGIA

FACOLTA' DI INGEGNERIA

Corso di Laurea in Ingegneria dell'Informazione

Nuovo Ordinamento

Tesi di Laurea

Realizzazione di un sistema software per lo svolgimento e gestione di prove di esame tramite interfaccia Web.

Laureando

Eugenio Favalli

Relatore

Prof. Walter Didimo

Anno Accademico 2002-2003

Capitolo 1: Introduzione 4
Capitolo 2: Svolgimento e gestione di prove di esame tramite Web 6
2.1 Il problema a grandi linee 9
Figura 2.1 Diagramma UML dei casi d'uso 9
2.2 Esigenze 10
2.3 Vantaggi dell'interazione tramite Web 11
Capitolo 3: Il sistema sviluppato 12
3.1 Funzionalità 13
3.1.1 Interfaccia amministratore 14
3.1.2 Accesso 15
3.1.3 Informazioni generali 17
3.1.4 Pannello di controllo 20
3.1.5 Interfaccia utente 28
3.1.6 Iscrizione 29
3.1.7 Consegna 30
3.2 Architettura 32
3.2.1 Step by Step 34
3.2.2 Mantenimento della sessione 35
3.2.3 Interazioni 37
3.3 Tecnologia 41
3.3.1 La scelta di Java 42
3.3.2 Le servlet 44
3.3.3 Test 45
3.4 Alcuni dettagli sulle classi 46
3.4.1 Classi principali 47
3.4.2 Moduli accessori 49
3.4.3 Librerie 52
Capitolo 4: Conclusioni e sviluppi futuri 53

Capitolo 1: Introduzione

Il Web e le sue risorse hanno riscosso sempre maggiore successo negli ultimi anni, sono diventate di uso comune e per alcuni ambiti addirittura indispensabili, hanno permesso la formazione di nuove figure professionali e di nuovi strumenti finora sconosciuti. Il mondo del Web è pertanto in continua evoluzione in ambito applicativo. Si pensi ad esempio al proliferare di piattaforme per l'e-commerce, per l'e-learning, o al continuo sviluppo di software per la ricerca e la catalogazione automatica delle informazioni distribuite tra le milioni di pagine del Web. I servizi software fruibili tramite interfaccia Web si avvalgono sempre più massicciamente di tecnologie per la creazione di pagine HTML dinamiche, offrendo un maggior livello di interazione con l'utente. Tra le più recenti tecnologie per la gestione di servizi Web interattivi, un ruolo significativo è rivestito dal linguaggio di programmazione Java, nato appositamente per essere integrato nel Web, grazie alle sue caratteristiche di portabilità e sicurezza.

L'obiettivo del seguente lavoro di tesi è stato la progettazione e la realizzazione di un servizio software per lo svolgimento e la gestione di prove di esame tramite interfaccia Web. Il sistema è stato implementato utilizzando tecnologia Servlet Java per la generazione di pagine HTML dinamiche lato server. L'utilizzo di Servlet Java offre alcuni vantaggi significativi rispetto all'uso di tecnologie più datate, quali le Common Gateway Interface (CGI), sia in termini di efficienza sia in termini di portabilità del prodotto su diverse piattaforme.

Le motivazioni che hanno spinto alla realizzazione di un tale servizio sono in primo luogo la possibilità per il docente di semplificare molte delle procedure relative alla creazione di prove di esame, come ad esempio la possibilità di generare una prova in breve tempo, attraverso l'uso di una interfaccia grafica e seguendo uno schema predefinito; favorendo il riuso di risorse precedentemente salvate in formato elettronico; o di controllare lo svolgimento della prova stessa ricevendo informazioni in tempo reale sullo stato delle consegne. L'utilizzo di un sistema software può inoltre facilitare le operazioni di catalogazione dei dati, effettuandole in maniera strutturata ed

automatica e fornendo utili strumenti per assistere il docente nella fase di correzione della prova. Per quanto riguarda le necessità degli studenti, l'impiego di un servizio Web viene incontro all'esigenza di disporre di una procedura standard per lo svolgimento di prove di esame, che semplifica le fasi di registrazione ed iscrizione, o di consegna degli elaborati realizzati. Dal punto di vista della consegna inoltre, si può notare che l'impiego delle tecnologie telematiche permette di agevolare lo scambio di dati tra studente e docente, permettendo la consegna degli elaborati prodotti, direttamente attraverso la rete e questo evita l'uso di supporti magnetici od ottici per lo scambio di dati con evidenti svantaggi dal punto di vista della catalogazione, sia per questioni di praticità che di organizzazione.

In particolare, il sistema sviluppato prevede l'utilizzo dell'interfaccia Web come strumento per la creazione di prove di esame, per il loro svolgimento e per la raccolta, l'accesso e l'analisi delle prove stesse. Il docente può, attraverso una semplice interfaccia grafica, personalizzare una prova di esame, inserendo direttamente testi o documenti aggiuntivi che definiscono la prova stessa. Lo studente sarà assistito dal sistema nella lettura dei testi che compongono la prova di esame e nelle fasi di consegna delle prove stesse. Infine, il docente può monitorare le consegne in tempo reale e richiedere la gestione automatica di uno storico da parte del sistema. Il sistema provvede altresì al confezionamento di un documento riassuntivo sullo stato delle consegne; tale documento può quindi direttamente essere integrato dal docente con le relative votazioni.

Nel seguito la trattazione si struttura come segue. Nel Capitolo 2 vengono descritte le idee di fondo che hanno portato al concepimento del sistema sviluppato e le principali caratteristiche che un servizio di gestione e svolgimento di prove di esame via Web deve avere, facendo brevi riferimenti alle motivazioni che hanno portato ad alcune scelte implementative. Nel Capitolo 3 viene descritto analiticamente il sistema sviluppato, sia nei termini delle sue funzionalità sia per quanto riguarda la sua architettura e le tecnologie utilizzate. Conclusioni e sviluppi futuri vengono forniti nel Capitolo 4.

Capitolo 2: Svolgimento e gestione di prove di esame tramite Web

Sempre maggiore importanza riveste l'impiego della tecnologia negli ambiti più disparati al fine di semplificare certi tipi di attività. In questo contesto anche l'ambito didattico viene sempre maggiormente integrato con l'uso di nuove tecnologie al fine di migliorare le tecniche di apprendimento. Si fanno rientrare sotto il termine di e-learning tutte queste metodologie orientate ad integrare la cultura digitale con l'apprendimento e in generale con tutto ciò che riguarda anche marginalmente l'insegnamento. Molti vedono nell'e-learning l'incontro tra tecnologie di rete e formazione a distanza. Ovvero e-learning (letteralmente "apprendimento elettronico") è un'espressione che indica tutte le modalità didattiche che utilizzano il Web, le reti telematiche e i sistemi per la comunicazione multimediale. Come ambienti di e-learning quindi vengono considerati particolari siti Internet creati da aziende o da istituzioni per erogare corsi di formazione e per supportare la comunicazione e la collaborazione fra gli studenti e fra studenti e docenti, in rete. Va precisato però che, secondo la definizione originale, non bisogna confondere l'e-learning con la formazione a distanza. Quest'ultima è uno strumento che permette di superare la barriera della distanza fisica nell'insegnamento. L'e-learning è molto di più: è un processo che integra insegnamento e apprendimento con l'aiuto delle tecnologie di rete. L'e-learning non consiste solo nella trasmissione di una serie di conoscenze, ma offre servizi che favoriscono l'apprendimento. In questo contesto si inserisce il sistema che si intende sviluppare, ovviamente solamente per quanto riguarda gli aspetti marginali, in quanto rappresenta unicamente uno strumento da affiancare alle normali tecniche di insegnamento. L'idea di base da cui nasce la necessità di sviluppare il sistema descritto in questo trattato consiste nella possibilità di impiego di tale applicazione all'interno di strutture didattiche dotate di laboratori con macchine opportunamente predisposte e configurate al fine di facilitare lo svolgimento di prove di esame. Il sistema offre innumerevoli vantaggi, che si evidenzieranno man mano che il software verrà analizzato, rispetto ad una normale gestione con interazione umana di una prova di esame a discapito comunque di una certa riduzione nella flessibilità.

L’ambito che gli è più congeniale risulta essere quello in cui le prove richiedono la produzione di file come ad esempio classi o programmi per esami di informatica, progetti per esami di disegno, insomma risulta particolarmente utile per la gestione e lo svolgimento di prove di esame che richiedano in qualche modo l’utilizzo del calcolatore e proprio in questo sta la potenzialità del sistema ovvero sfrutta la connessione in rete per la gestione e la raccolta degli elaborati prodotti dagli studenti senza bisogno di utilizzare supporti ottici o magnetici. Le specifiche iniziali prevedono la creazione di un sistema che supporti due tipi di interazioni: la prima consiste nella possibilità per un professore o amministratore del sistema di creare prove di esame e di gestirne l’esecuzione, nel seguito della trattazione ci si riferirà a questo tipo di interazione come interfaccia amministratore; la seconda rappresenta invece uno strumento per gli studenti per svolgere le prove di esame, questo tipo di interazione verrà riproposto nel seguito come interfaccia utente. Per quanto riguarda l’interfaccia amministratore si richiede in particolare che il sistema offra la possibilità di aprire una sessione di esame specificando del testo o inviando dei file sul server che ospita l’applicazione. Il testo dovrà essere visibile agli studenti che potranno inoltre accedere ai file tramite download. Il docente deve essere in grado di visualizzare la lista degli studenti iscritti in ogni momento. Su tale lista dovranno essere visualizzati per ogni studente nome cognome e numero di matricola. Saranno inoltre disponibili informazioni sui file correntemente inviati. Il docente deve poter definire il nome con cui i file inviati dagli studenti, qualsiasi sia il loro nome originale devono essere memorizzati sul server. Potrà decidere inoltre il numero dei file che dovranno essere inviati tramite upload dagli studenti. L’accesso al sistema da parte del docente deve avvenire a seguito di una procedura di autenticazione con inserimento di una utenza e di una password. I diritti di accesso al sistema dovranno essere stabiliti dall’amministratore del calcolatore che ospita il sistema, da non confondere con l’amministratore cui si fa riferimento quando si parla di interfaccia, ovvero il docente. L’amministratore provvederà quindi alla creazione di un account personalizzato per ogni docente autorizzato ad accedere al sistema o alla rimozione di un account non più utilizzato. La scelta del sistema impiegato per la gestione degli account non è definita nelle specifiche, ma viene demandata allo sviluppo del sistema, ovvero verranno adottate le scelte che meglio si adegueranno alle necessità implementative.

Per quanto riguarda l'interfaccia utente, si richiede di sviluppare una interfaccia grafica tramite la quale gli studenti possano interagire per sostenere la prova di esame. Lo studente deve prima di tutto registrarsi al sistema introducendo i propri dati personali ovvero nome, cognome e numero di matricola. Il software, all'atto dell'iscrizione di un nuovo studente, deve riservare automaticamente parte dello spazio sul server creando una cartella personale per ogni utente iscritto. L'utente ha a disposizione un'area di testo per effettuare operazioni di copia e incolla di testi. Si ha inoltre la possibilità di inviare un certo numero, definito al momento della creazione della prova di esame, di file generici. L'upload deve infatti permettere di inviare al server qualsiasi tipo di file ad esempio file di testo, immagini bitmap, ecc. Ogni file inviato viene salvato secondo il nome stabilito dall'amministratore al momento della generazione della prova di esame. Copie duplicate o diverse versioni di un medesimo file non devono essere sovrascritte, ma rinominate in un nuovo file. Per tenere traccia di tutte le versioni fornite per ogni file che è possibile inviare in upload viene creato o aggiornato uno storico che oltre a mantenere informazioni sulle diverse versioni ne valuta anche le relazioni temporali che tra loro intercorrono. Gli obiettivi di base introdotti dalle specifiche sono stati sempre considerati come requisiti indispensabili durante la fase di implementazione, e man mano che il progetto prendeva forma sono stati aggiunti ulteriori dettagli al sistema ed altre specifiche per rendere l'applicazione maggiormente efficiente e dotata di un numero maggiore di funzionalità e di possibilità di personalizzazione ed elasticità. Alcuni particolari introdotti durante lo sviluppo sono stati rimossi in breve poiché cozzavano con le esigenze di sicurezza necessarie all'ambito per cui il sistema è stato progettato.

2.1 Il problema a grandi linee

Si richiede di sviluppare un sistema in grado di fornire delle funzionalità di base per la creazione e gestione di prove di esame tramite interfaccia Web. Il sistema si deve articolare in due applicazioni principali di cui una servirà per l'interazione del sistema con l'amministratore: nel nostro caso per amministratore si intende il generico professore che apre una sessione di esame. La seconda applicazione dovrà invece essere utilizzata dall'utente, inteso come studente, per sostenere l'esame. Nella figura seguente si evidenziano tramite diagramma UML (unified modeling language) dei casi d'uso le interazioni che sussistono tra amministratore, utenti e le interfacce che permettono di utilizzare il sistema.

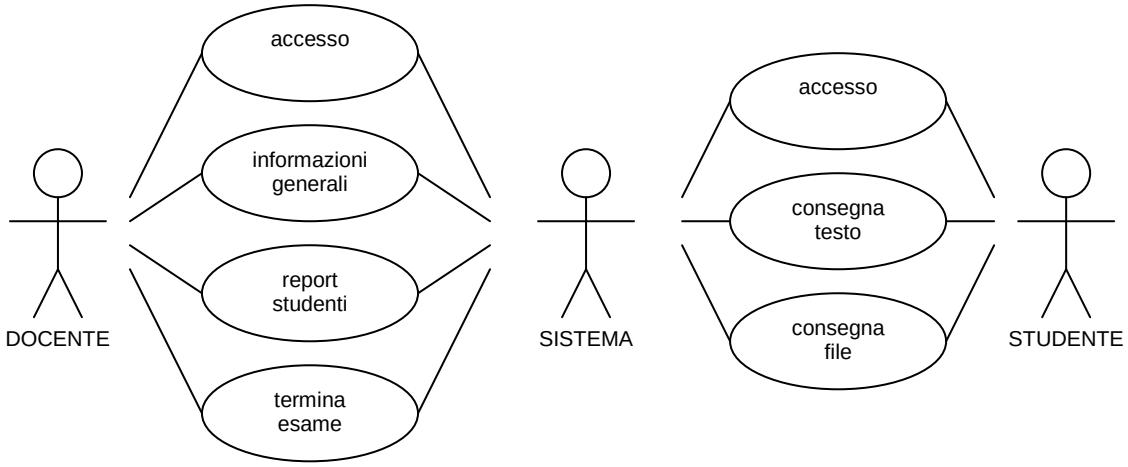

Figura 2.1 Diagramma UML dei casi d'uso

Nel diagramma le figure costituite da uomini stilizzati rappresentano gli attori, ovvero le persone o altri sistemi che interagiscono con il sistema modellato; le ellissi rappresentano invece i casi d'uso che sono le sequenze di azioni svolte dal sistema che generano un risultato osservabile; le linee rappresentano i legami tra casi d'uso ed attori. Il diagramma UML dei casi d'uso risulta particolarmente utile in questa fase di progettazione in quanto costituisce una tecnica di analisi dei requisiti, ma può risultare utile anche in fasi successive. Fornisce inoltre una visione ad alto livello delle azioni compiute dal sistema.

2.2 Esigenze

Il sistema deve essere dotato di funzionalità che permettano all'amministratore di accedere tramite autenticazione con utenza e password. Tali informazioni vengono codificate in un file dall'amministratore del calcolatore su cui risiede il sistema. si richiede la possibilità di inserire informazioni generali riguardanti la prova di esame come ad esempio il titolo o il docente cui l'esame si riferisce; un testo che rappresenta il testo del compito o istruzioni riguardo l'esecuzione; la possibilità di personalizzare l'interfaccia utente selezionando se attribuire all'utente la capacità di poter inserire direttamente testo, oppure dichiarando il numero di file che dovranno essere inviati tramite upload ed il nome con cui tali file verranno salvati dal sistema; un pannello di controllo tramite il quale avere aggiornamenti in tempo reale riguardo ad informazioni sugli studenti correntemente iscritti e sulle risposte o file pervenuti fino al momento del controllo, con la possibilità di poter verificare inoltre lo stato della documentazione fornita da ogni singolo studente in ogni momento durante la durata della prova di esame. Si richiede inoltre che il sistema sia in grado di salvare, in file diversi, tutte le versioni di testo e file fornite dagli studenti, in modo tale da verificare eventuali incongruenze o mancanze nelle risposte.

Dal lato utente si richiede che l'interfaccia generata in base alle informazioni fornite dall'amministratore consenta un accesso tramite nome, cognome e numero di matricola che provvede automaticamente ad aggiornare una lista degli studenti presenti allo svolgimento della prova di esame. Una volta ottenuto l'accesso al sistema devono essere disponibili alcuni form che permettano l'invio di testo e dei file richiesti.

2.3 Vantaggi dell'interazione tramite Web

La decisione di utilizzare l'interfaccia fornita dal servizio di rete Web deriva da alcune semplici considerazioni pratiche che ne riconoscono l'utilità e l'efficacia. In particolare si è preso spunto dall'esempio di alcuni esami che prevedono lo svolgimento di parte o tutto l'esame all'interno di laboratori dotati di macchine connesse tramite rete locale LAN. Il vantaggio immediato si riscontra nel fatto che la rete può essere impiegata oltre per condividere risorse, principalmente per lo scambio di dati e questo comporta la possibilità di evitare l'utilizzo di supporti rimovibili come ad esempio dischi magnetici od ottici per lo scambio di dati. L'architettura del sistema inoltre prevede la possibilità di effettuare copie di backup della sessione in corso di svolgimento evitando così problematiche ed eventuali danni derivanti dalla perdita totale o parziale dei dati. L'interfaccia definisce inoltre uno schema preciso e standardizzato per tutti gli utenti che permette una maggiore sistematizzazione a discapito di una leggera perdita di flessibilità dovuta alla presenza di un numero limitato di opzioni di personalizzazione della prova, tale perdita risulta però un buon compromesso se valutata a fronte dei vantaggi apportati da tale tecnica.

Capitolo 3: Il sistema sviluppato

Nel seguente capitolo verrà analizzato il sistema in tutte le sue componenti tentando di evidenziarne le caratteristiche salienti. La trattazione procederà partendo da un alto livello di astrazione e scenderà sempre più nel dettaglio fino ad accennare in breve ad alcuni dettagli implementativi. Si tenterà in primo luogo di delineare nel dettaglio tutte le funzionalità offerte dal sistema e le modalità di utilizzo. Si procederà quindi ad illustrare lo scheletro architetturale con il quale l'applicazione è stata concepita facendo riferimento alle tecniche e modelli su cui si è basato il progetto. Nel seguito verranno sottolineate le componenti tecnologiche che realizzano il prodotto software evidenziando le motivazioni che hanno portato ad alcune scelte piuttosto che ad altre. Si farà anche riferimento alle metodologie che sono state utilizzate durante le fasi di sviluppo o di test, e agli strumenti impiegati per l'installazione ed esecuzione del sistema. In questo contesto verranno forniti anche requisiti ed indicazioni generali necessari al corretto funzionamento del software. In ultima analisi si scenderà nel dettaglio fornendo alcune informazioni sulle componenti che costituiscono l'applicativo, illustrando le interazioni tra le componenti e mettendo in luce particolari aspetti del codice e di librerie esterne utilizzate.

3.1 Funzionalità

Di seguito verranno illustrate le varie funzionalità offerte dal sistema esaminando le procedure per la creazione e la gestione di prove di esame che si avvalgono dell'uso dell'interfaccia Web. Oltre che ad un elenco delle caratteristiche che presenta il software si illustrerà il loro funzionamento e verranno descritti particolari che vengono implementati dal sistema, ma che non sono visibili attraverso l'interazione tramite l'interfaccia. Per comodità ai fini della trattazione si farà distinzione tra le caratteristiche fornite dall'interfaccia amministratore e dall'interfaccia utente. Di seguito perciò i primi paragrafi si occuperanno di esaminare l'interfaccia amministratore mentre ai seguenti è demandata la trattazione del funzionamento dell'interfaccia utente. Si ricorda che in questo contesto con il termine amministratore si fa riferimento al docente con privilegio di accesso al sistema, mentre con il termine utente si indica lo studente iscritto al sistema.

3.1.1 Interfaccia amministratore

L’interfaccia amministratore è costituita da una serie di pagine Web. La prima di queste è statica e svolge le funzionalità di garantire l’accesso unicamente a chi è accreditato. Le pagine successive sono generate dinamicamente, ognuna secondo le informazioni inserite al passo precedente o ricavate estraendo informazioni da alcuni file creati dal sistema per il suo corretto funzionamento. In questi file vengono memorizzate le informazioni generali riguardanti la prova di esame, man mano che queste vengono fornite attraverso i form dell’interfaccia Web; oppure gli studenti correntemente iscritti a sostenerla; o infine vengono ricavate tramite variabili d’ambiente come ad esempio l’ora e la data di svolgimento della prova di esame.

3.1.2 Accesso

La prima pagina dell'interfaccia amministratore mostrata nella figura di seguito, è costituita da una pagina statica che permette l'acceso al sistema tramite autenticazione con utenza e password.

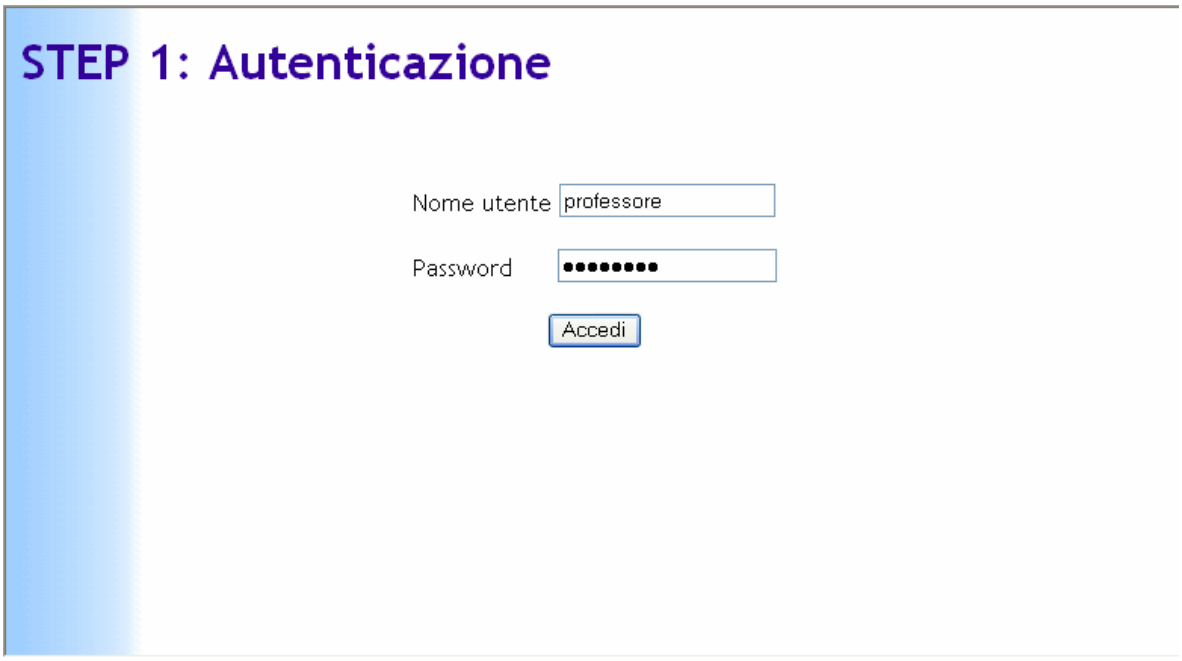

Figura 3.1 Autenticazione

L'accesso è consentito ad ogni utente registrato nel sistema con nome utente e password. Al tentativo di accesso utenza e password inseriti, vengono ricercati all'interno di un elenco di utenti accreditati. L'elenco è costituito da un file di nome "PWDList.dat", ovvero un file che contiene l'elenco dei docenti che hanno diritto di accesso al sistema con le loro rispettive password codificati secondo lo standard UTF. Al momento non viene applicata ai dati nessun tipo di crittografia che impedirebbe invece la modifica non autorizzata del file. La lista degli utenti accreditati dovrebbe essere aggiornata unicamente dall'amministratore della macchina su cui risiede il sistema. Al momento, insieme al software viene fornita una piccola applicazione chiamata PWDCreator che permette di aggiungere, modificare o rimuovere un account amministratore dalla lista contenuta nel file. Se l'utente non viene riconosciuto l'applicazione genera il messaggio di errore illustrato nella figura a pagina seguente per avvertire che non si dispone del privilegio di accesso.

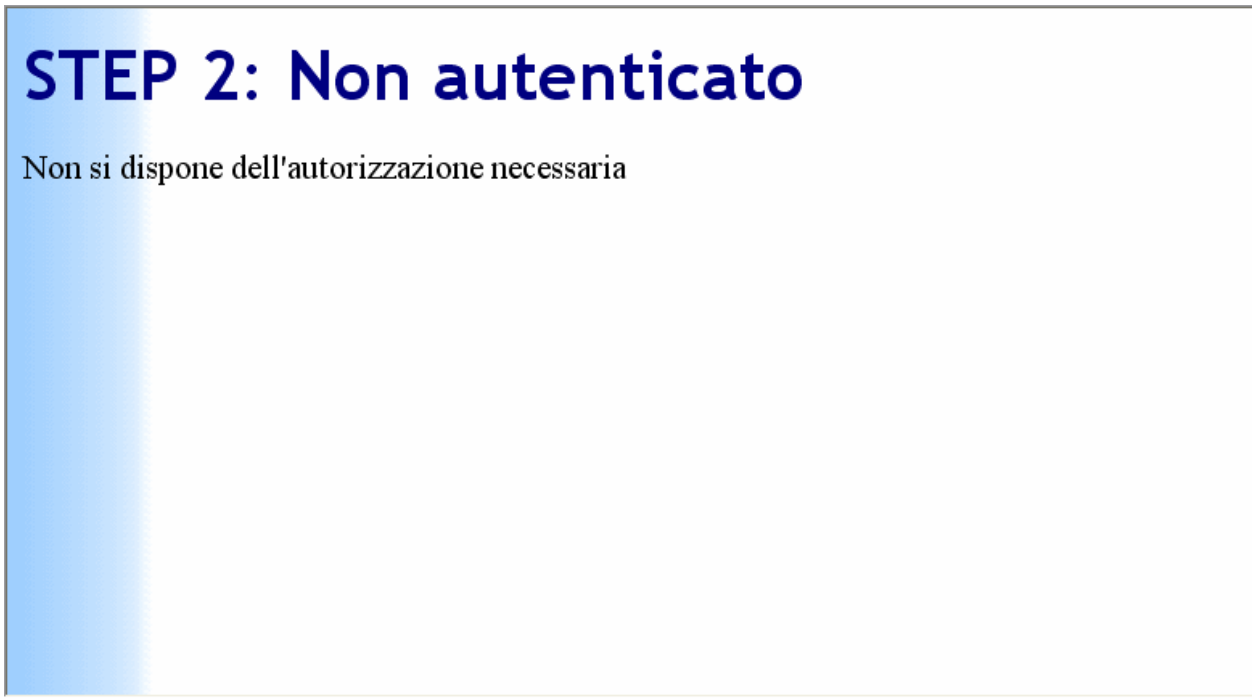

Figura 3.2 Accesso negato

3.1.3 Informazioni generali

Se si dispone dell'autorizzazione necessaria è possibile proseguire al passo successivo che permette di inserire un insieme di informazioni ed impostazioni generali per generare la prova di esame. Le informazioni introdotte a questo livello influenzeranno anche la creazione dell'interfaccia Web dello studente. L'immissione avviene tramite l'utilizzo di una serie di form che comprendono aree di testo, checkbox e tasti di azione. L'interfaccia che permette l'operazione di inserimento è mostrata in figura.

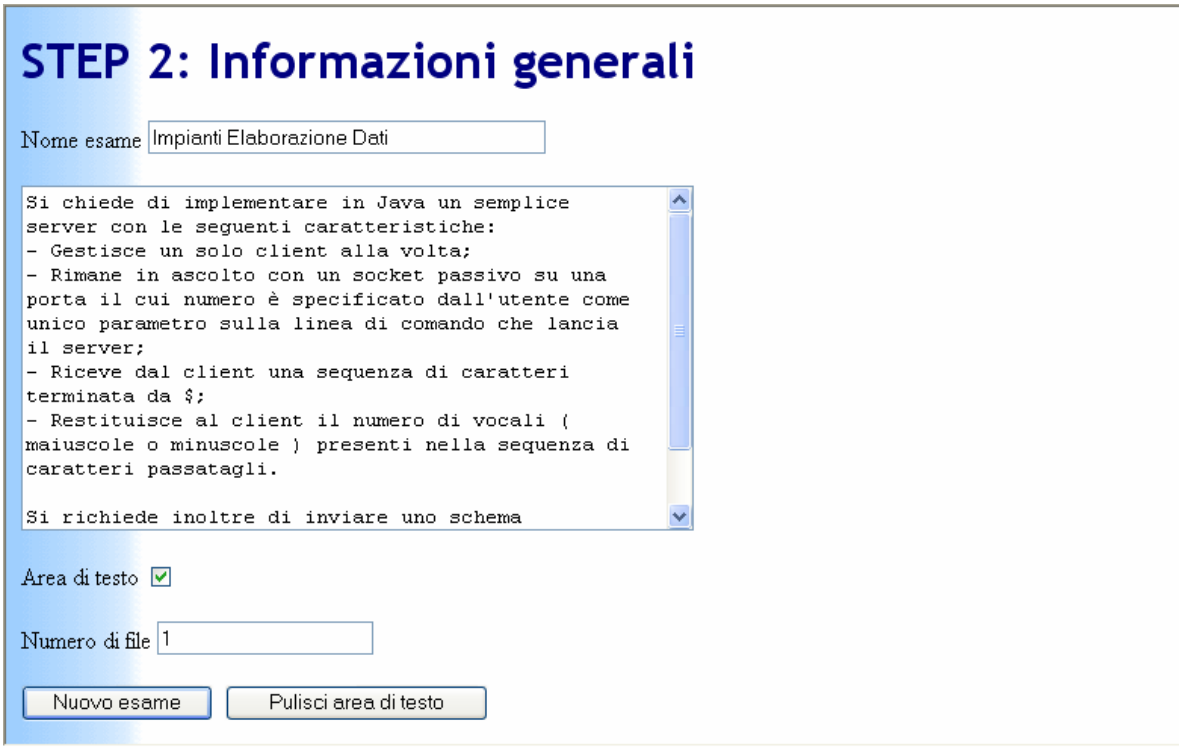

Figura 3.3 Inserimento informazioni

Esaminando il contenuto dell'immagine si vede che l'interfaccia mette a disposizione una casella di testo in cui va indicato il nome relativo all'esame che si desidera creare o in alternativa è possibile inserire il nome del professore cui fare riferimento per la corrente prova di esame. L'area di testo viene impiegata per immettere il testo della prova o eventualmente informazioni supplementari riguardanti lo svolgimento, nel caso in cui il testo sia fornito direttamente tramite file da scaricare. L'area di testo supporta le operazioni di "copia e incolla", è perciò possibile creare il testo da inserire tramite qualsiasi altra applicazione per l'editing dei testi, copiarla in memoria e quindi

incollarla nell'area di testo. Per la creazione dell'immagine in figura si è utilizzato appunto questo procedimento, infatti il testo è stato prima creato utilizzando l'applicazione "Blocco Note" di Windows e poi è stato copiato all'interno dell'area di testo. La checkbox può essere spuntata se si vuole che anche nell'interfaccia utente sia disponibile un'area di testo tramite la quale sia possibile inserire direttamente informazioni testuali con operazione di "copia e incolla". La casella di testo etichettata come "Numero di file" permette di specificare il numero di file che dovranno essere consegnati dagli utenti nel corso della prova di esame e influisce quindi sulla generazione dinamica delle pagine dell'interfaccia utente modificando il numero di form per l'upload che verranno visualizzati. Il tasto "Pulisci area di testo" serve a reimpostare il contenuto dell'area di testo mentre il tasto "Nuovo esame" permette di proseguire al passaggio successivo. Alla pressione del tasto "Nuovo esame" il sistema crea un file il cui nome dipende dalla data in cui l'esame viene sostenuto, nel quale memorizza il contenuto dell'area di testo, il titolo associato alla prova di esame, il numero di file previsti per l'upload da parte degli utenti e se la checkbox è stata spuntata. Se è stata prevista la possibilità di upload di file specificando un numero maggiore di zero nella casella "Numero di file" verrà visualizzata la schermata mostrata nella pagina seguente.

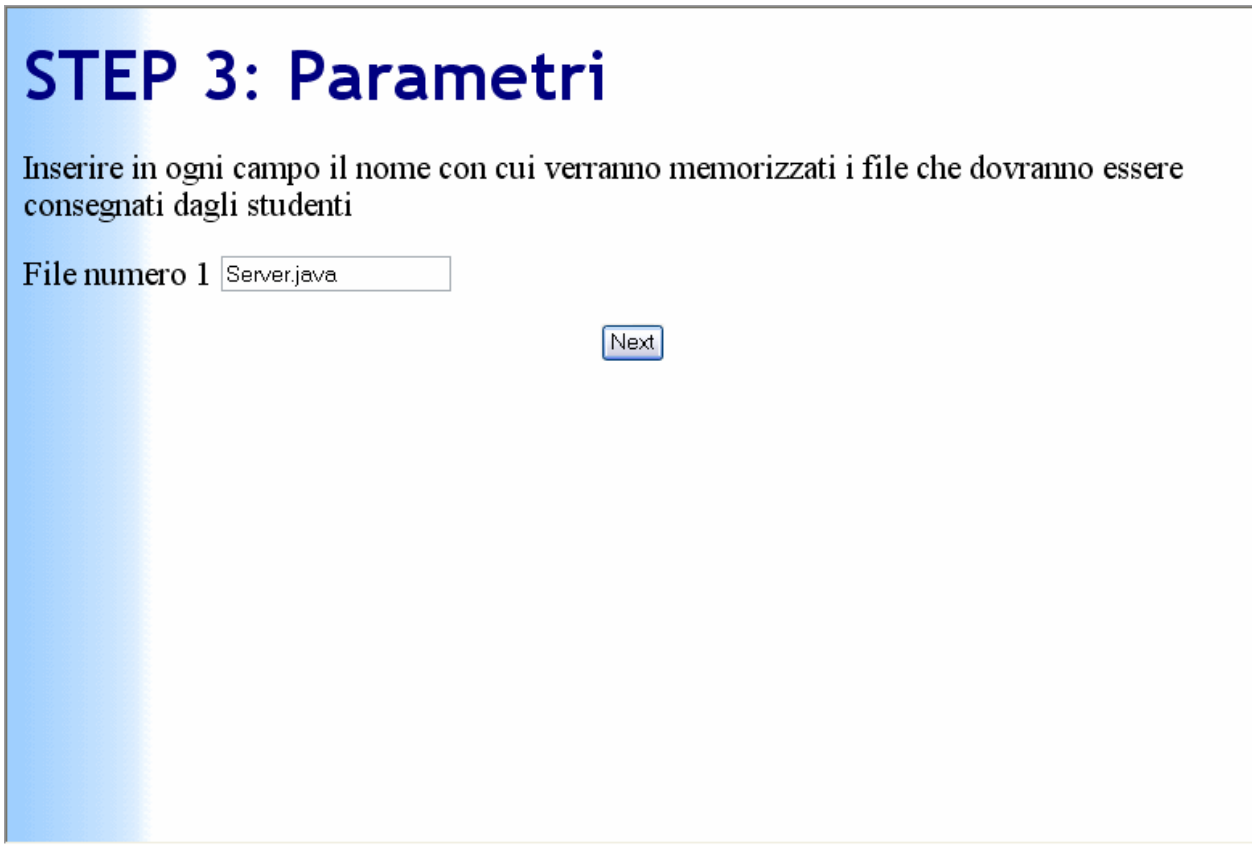

Figura 3.4 Inserimento nomi dei file

In questa sezione è possibile specificare per ogni file previsto in upload il nome con cui questo dovrà essere salvato sul server ignorando il nome del file originale. Per l'esempio in figura è stato previsto un unico file che dovrà essere inviato in upload dagli utenti. Qualsiasi file verrà inviato al server tramite upload sarà rinominato come "Server.java".

3.1.4 Pannello di controllo

Il pannello di controllo rappresenta la parte principale di tutto il sistema in quanto permette la gestione della prova di esame durante il suo svolgimento. Offre funzionalità di upload di file e di monitoraggio degli studenti iscritti.

Figura 3.5 Pannello di controllo

La figura evidenzia in primo luogo una lista dei file correntemente salvati sul server ovvero quell'insieme di file che sono stati inviati tramite upload dall'amministratore perché possano essere consultati dagli utenti. Tali file dovrebbero contenere ulteriori informazioni non fornite dal testo, consigli o parti necessarie allo svolgimento della prova di esame. Ogni elemento della lista è costituito dal nome del file cui è associato un hyperlink che permette di accedere direttamente al file per verificarne la correttezza e da un tasto "Delete" che permette di rimuovere il file in questione dal server. Nella figura a pagina seguente viene visualizzato il messaggio fornito dal sistema se la procedura di elminazione è andata a buon fine.

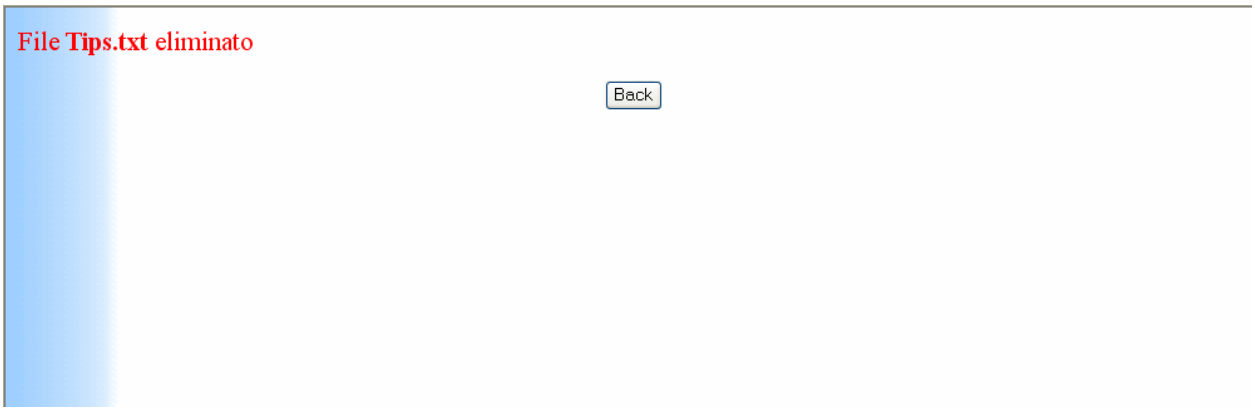

Figura 3.6 Conferma eliminazione

La lista dei file salvati sul server viene mantenuta dal sistema nel file “File.lst”. Il form successivo viene impiegato per l’upload dei file, tramite il tasto “Sfoglia…” si accede al dialog di selezione del file mostrato nella figura sottostante:

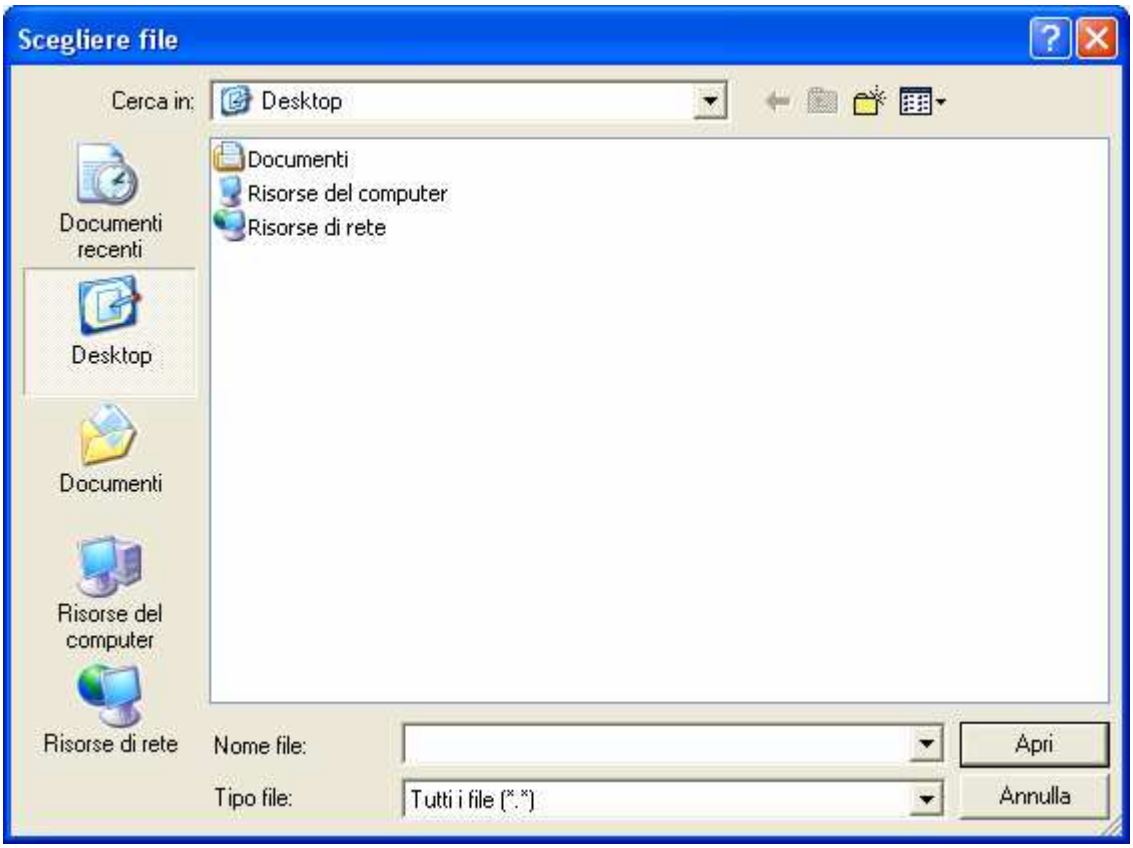

Figura 3.7 Dialog di selezione file

Il percorso relativo al file selezionato comparirà quindi nella casella di testo a fianco del tasto “Sfoglia…” e sarà possibile inviarlo e salvarlo sul server premendo il tasto “Invia”.

In figura viene visualizzato il messaggio di conferma relativo all'invio del file "Tips.txt".

Figura 3.8 Conferma upload

Sia nel caso di eliminazione che di upload di un file nelle pagine di conferma generate è presente un tasto "Back". Agendo su tale tasto si viene reindirizzati al pannello di controllo aggiornato. La funzione del tasto "Reimposta" è semplicemente quella di cancellare il contenuto della casella di testo. Come evidenziato dal testo nella figura 3.5 inoltre la dimensione massima dei file che possono essere spediti in upload è di 1 MB. L'introduzione di tale limite è dovuta al fatto che spesso per upload di dimensioni eccessive il server rimane in ascolto per un tempo elevato diminuendone drasticamente l'efficienza, inoltre per questioni di condivisione di spazio con altri utenti si è ritenuto giusto limitare la porzione di memoria disponibile per ogni amministratore al fine di evitare di raggiungere condizioni di saturazione di spazio. Nel caso in cui nessun file sia stato inviato tramite upload verrà visualizzata in colore rosso la scritta mostrata nella figura sottostante:

Figura 3.9 Nessun file salvato

Utilizzando la funzione di aggiornamento fornita dal browser utilizzato è possibile verificare in ogni momento lo stato attuale del sistema e la situazione relativa agli studenti iscritti alla prova di esame. Nell'esempio mostrato, utilizzando il tasto di aggiornamento mostarto in figura, fornito da Internet Explorer si provoca l'aggiornamento dei dati visualizzati nel pannello di controllo.

Figura 3.10 Tasto "Aggiorna"

A questo proposito è possibile verificare dall'immagine 3.5, che viene indicato l'orario dell'ultimo aggiornamento in modo da tenere costantemente sotto controllo la validità dei dati forniti dal sistema. A seguire viene fornita una tabella con l'elenco degli studenti iscritti al momento dell'aggiornamento della pagina e la loro situazione attuale. La tabella viene generata dinamicamente e presenta la colonna intitolata come "Testo" unicamente se al momento dell'introduzione delle informazioni generali è stata selezionata la checkbox relativa all'area di testo. Nella tabella vengono visualizzate le informazioni di accesso per ogni studente ovvero nome cognome e numero di matricola, ed informazioni riguardanti lo stato della consegna delle parti che compongono la prova di esame. La colonna "Testo", se prevista, indica lo stato della consegna di informazioni testuali tramite l'area di testo. I simboli visualizzati assumono i seguenti significati:

- E' stata consegnata almeno una versione di informazione testuale.
- Nessuna informazione ancora pervenuta.

Nella colonna "File consegnati" viene visualizzato il numero di file correntemente inviati tramite upload rispetto al numero totale di file previsti dalla prova di esame. Al nome degli studenti è associato un collegamento ipertestuale che permette di accedere ad informazioni maggiormente dettagliate relative alla situazione di consegna per lo studente in questione.

Nell’esempio si avrà:

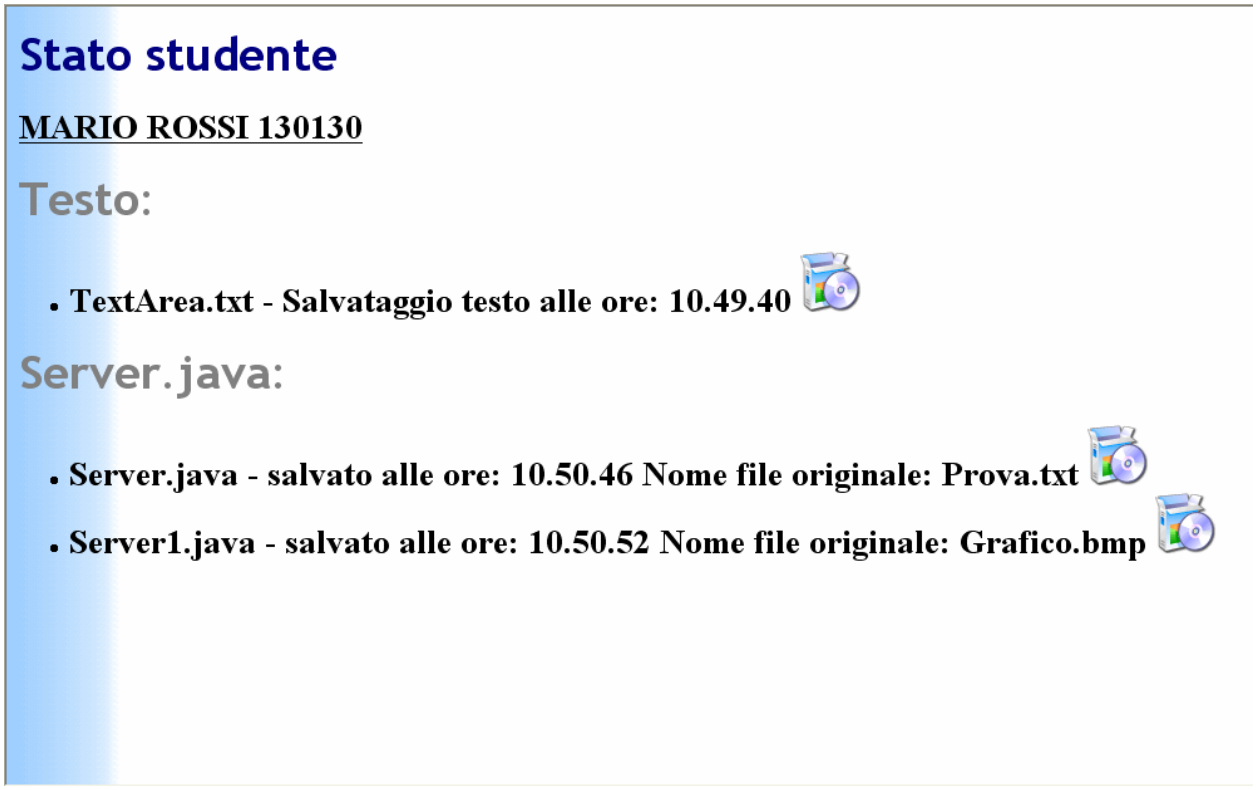

Figura 3.11 Dettagli studente

Nella figura sono riepilogate le consegne dell’area di testo e dei file tramite upload fornendo anche informazioni riguardanti l’ora di consegna, il nome con cui vengono salvate le varie versioni del file e il nome originale del file inviato in upload. Agendo sull’immagine è possibile accedere direttamente alla versione del file cui è associata. Se il formato del file è supportato dalla visualizzazione del browser questo verrà mostrato direttamente in finestra, altrimenti verrà fornita la possibilità di salvarlo in locale tramite dialog di download mostrato a pagina seguente.

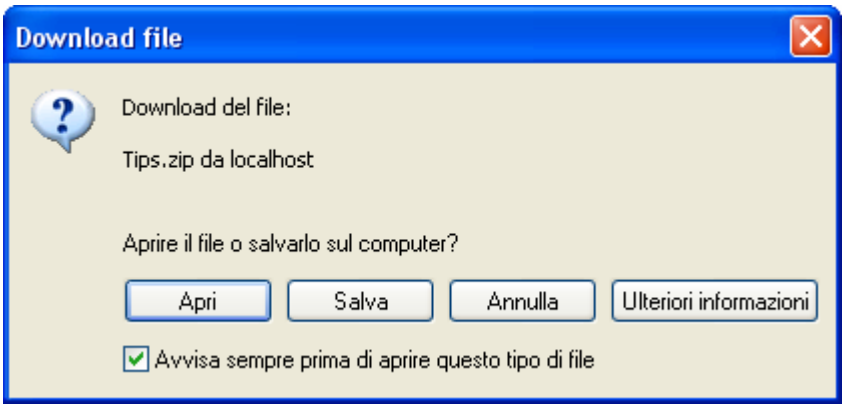

Figura 3.12 Dialog di download

Agendo sul tasto “Termina” il sistema rimuove i file utilizzati per la creazione della sessione di esame e salva una copia di backup della lista degli studenti presenti e delle cartelle contenenti i file e il contenuto dell’area di testo se prevista, per ogni utente.

Figura 3.12 Termine esame

Il collegamento ipertestuale permette di accedere direttamente alla cartella sul server che ospita il sistema dove è stato effettuato il backup dei dati relativi alla sessione di esame appena conclusa.

Nella figura seguente si può verificare l'effetto del collegamento ipertestuale:

Figura 3.13 Copia di backup

La locazione in cui effettuare il backup relativo ad una prova di esame è stabilito automaticamente dal sistema. In primo luogo, se non esiste, viene creata una cartella il cui nome è costituito dalla parola "Esami" seguita dalla data in cui la prova viene svolta. In questo modo i backup riferiti ad esami che si svolgono nello stesso giorno saranno salvati tutti nella stessa posizione e ciò renderà anche più facile la ricerca poiché gli esami sono divisi in base al giorno di svolgimento. All'interno di questa cartella ne viene creata una ulteriore il cui nome è costituito dal nome attribuito all'esame durante la fase di raccolta delle informazioni, seguito dall'orario in cui la prova è stata avviata. Questo permette di discriminare ad esempio esami con lo stesso nome. Infatti poiché il sistema supporta lo svolgimento di una sola prova di esame alla volta questo assicura che non ci sia ambiguità tra due esami con lo stesso nome in quanto possono essere comunque discriminati in base all'orario. La ricerca di una prova conclusa risulta pertanto piuttosto agevole in quanto basta selezionare la cartella che rappresenta il giorno in cui ha avuto luogo la prova e ricercare al suo interno la cartella che corrisponde per titolo dell'esame e per orario di inizio. Durante la fase di backup il sistema provvede a creare anche un file denominato "Votazioni.xls" che, come evidente dall'estensione, rappresenta un foglio di calcolo per l'applicazione Microsoft Excel. In tale foglio il sistema, accedendo ai dati relativi agli studenti che hanno sostenuto la prova, crea una riga per ogni utente, nella quale inserisce nome, cognome e numero di matricola, e creando una colonna per ogni parte di cui si compone la prova. Se la prova prevede il riempimento dell'area di testo si avrà una colonna dedicata a questa parte. Tale foglio di calcolo viene creato con il fine di riempirlo con le votazioni degli studenti relative alla prova cui si riferisce. Nella figura

seguente si riporta un immagine di un foglio creato a seguito di una prova di esame con due studenti iscritti. La prova consiste nella consegna di una parte testuale e di due file tramite upload.

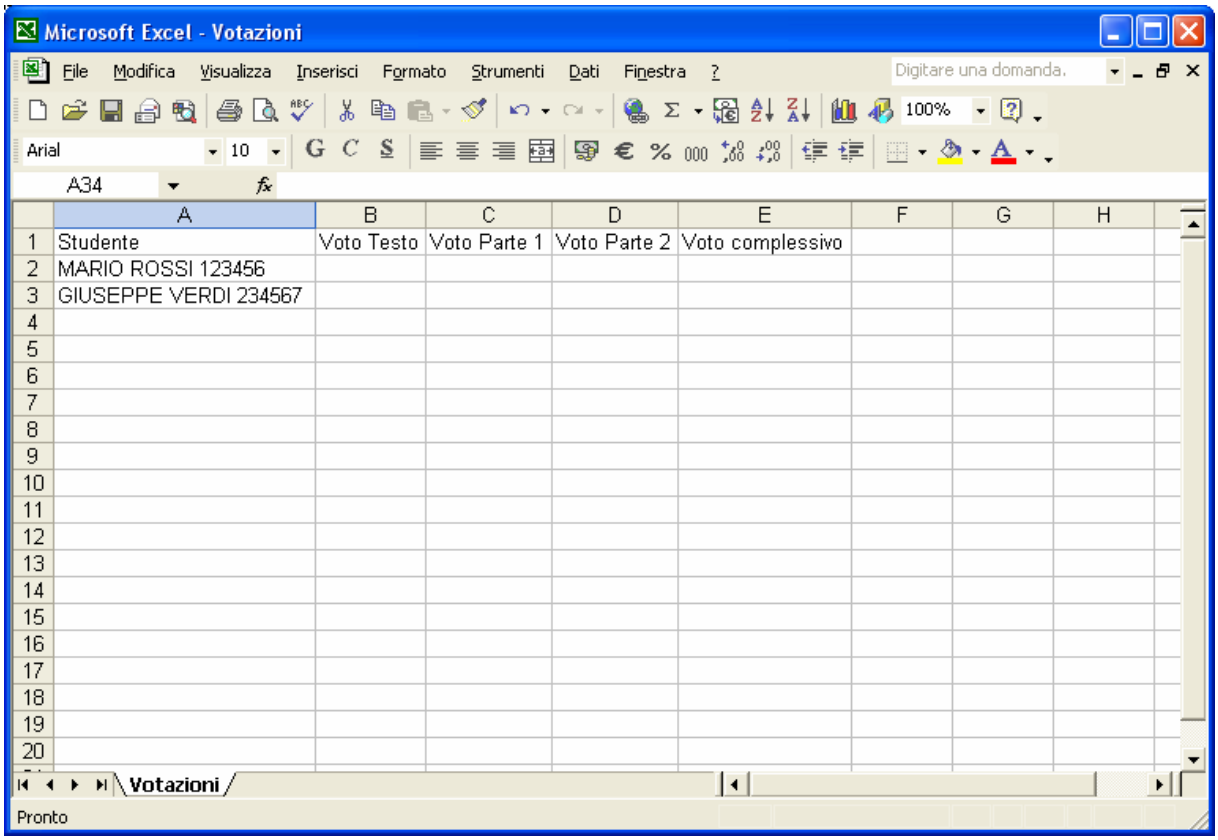

Figura 3.14 Foglio di calcolo

3.1.5 Interfaccia utente

L’interfaccia utente presenta unicamente due fasi: una prima fase di accesso al sistema per la raccolta di dati sullo studente, ovvero vengono richiesti nome cognome e numero di matricola. Tale fase viene realizzata tramite una pagina web statica. La seconda fase, che rappresenta lo svolgimento della prova di esame, permette di inserire testo o di effettuare upload di file. Viene realizzata tramite composizione di una pagina web dinamica basata principalmente sulle informazioni inserite al momento della creazione della prova di esame.

3.1.6 Iscrizione

La prima fase consiste in effetti in una sorta di iscrizione alla prova di esame: viene richiesto di accedere al sistema tramite introduzione di nome cognome e numero di matricola.

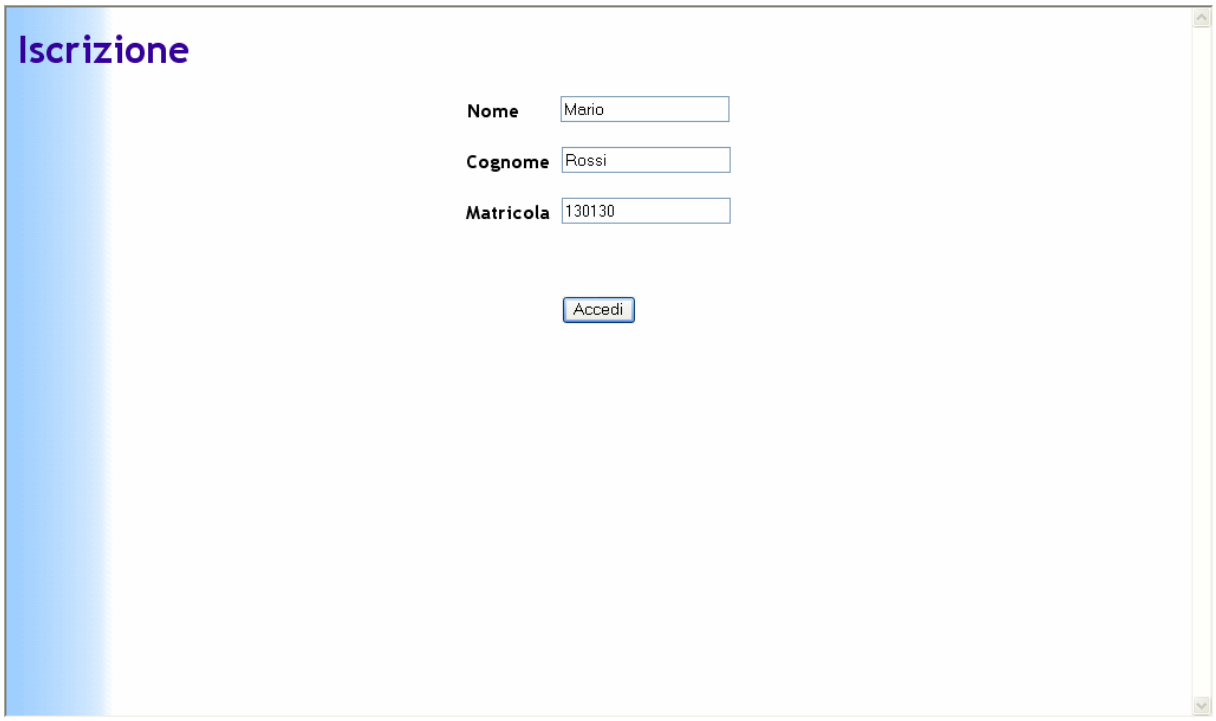

Figura 3.15 Iscrizione

Queste informazioni vengono utilizzate dal sistema per aggiornare la lista degli studenti presenti alla prova di esame e per aprire una sessione specifica per ogni utente connesso. La lista degli studenti viene mantenuta in un file denominato “Student.lst” nel quale per ogni studente vengono memorizzati nome, cognome e numero di matricola. Se l’utente non risulta già registrato, al momento dell’iscrizione, il sistema gli riserva parte dello spazio sul server creando una cartella che ha per nome i dati personali dello studente e nella quale il sistema salverà tutti i file o il testo consegnati dallo studente.

3.1.7 Consegna

Il passo successivo presenta la seguente schermata:

Figura 3.16 Svolgimento

Al momento del primo accesso il sistema accetta l'iscrizione,

Figura 3.17 Iscrizione studente

mentre alle successive conferma il riconoscimento dell'utente mostrando i dati registrati.

Figura 3.18 Riconoscimento studente

Vengono visualizzati il nome dell'esame, il testo e la lista di file salvati sul server. Per ogni file nella lista è associato un collegamento ipertestuale che permette di scaricarlo. L'area di testo viene visualizzata unicamente se prevista al momento della creazione della prova di esame. Il testo digitato oppure inserito con operazione di "copia e incolla" nell'area può essere inviato e salvato sul server agendo con il tasto "Consegna". Premendo il tasto "Pulisci area di testo" si cancella invece il contenuto dell'area di testo. In basso sono presenti tanti form per l'upload quanti ne sono stati specificati al momento della creazione della prova di esame.

Figura 3.19 Form di upload

Come mostrato in figura ogni form reca come intestazione, nel caso in esame "Grafico", il nome selezionato dall'amministratore con il quale il file, qualunque sia il suo nome originale, verrà salvato. Il funzionamento risulta simile a quello visto nel caso dell'interfaccia amministratore con l'unica eccezione che in questo caso i file vengono rinominati.

3.2 Architettura

Al fine di mantenere la struttura del sistema la più semplice e compatta possibile si è scelto un approccio architetturale costituito da una serie di moduli elementari che svolgono un insieme minimo di funzioni. Questo garantisce una maggiore leggibilità del codice ed aumenta la facilità di manutenibilità ed espandibilità del software. A questi vantaggi va inoltre aggiunto quello di una maggiore flessibilità del sistema che permette di generare pagine consecutive in gran parte personalizzabili. D'altro canto l'introduzione di una architettura modulare introduce problematiche relative alla comunicazione e scambio di informazioni riservate tra un modulo e l'altro in quanto non persiste un unico stato del sistema. Ad esempio la trasmissione di informazioni relative all'accesso con autenticazione come utenza e password o riconoscimento dell'utente accreditato in ogni modulo, introducono ulteriori difficoltà. Si richiede di evitare di fare ricorso all'utilizzo di altre tecniche come ad esempio l'uso dei cookies per evidenti motivi di riservatezza e sicurezza. Lo sviluppo del software ha seguito un approccio di tipo "Rapid prototyping" che prevede lo sviluppo in tempi brevi di una serie di prototipi funzionanti del sistema. Questa tecnica permette nella maggior parte delle situazioni di effettuare una migliore cattura dei requisiti, inoltre infonde maggiore fiducia negli sviluppatori che si stanno raggiungendo i risultati ottenuti. Nell'esempio in questione inoltre la creazione di un prototipo permette di valutare ed affrontare in maniera più concreta i rischi insiti nell'impiego di tecnologie alternative. Dall'altro lato gli svantaggi legati al "Rapid prototyping" si riscontrano nell'elevato dispendio di tempo legato alla realizzazione di prototipi che verranno in seguito scartati. Nella figura a pagina seguente si tenta di schematizzare l'approccio seguito.

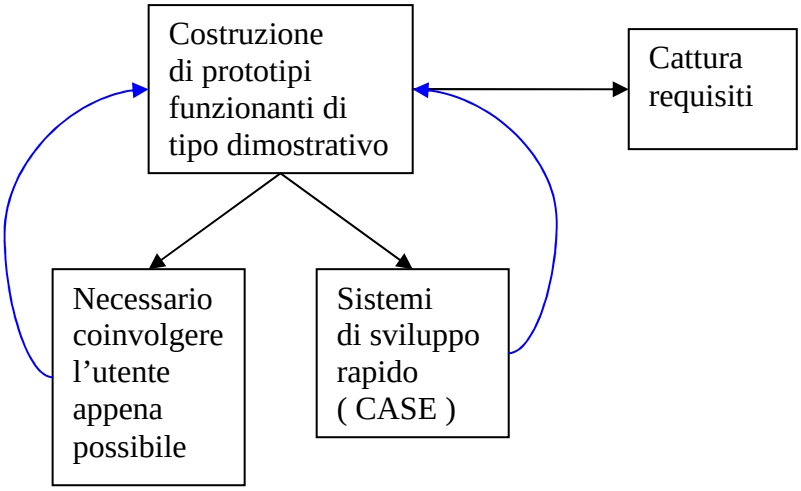

Figura 3.20 Rapid prototyping

3.2.1 Step by Step

Come accennato nell'introduzione, l'architettura prevede sia per la generazione dell'interfaccia amministratore, che per l'interfaccia utente un approccio modulare. In generale ogni modulo si occupa della generazione di una singola pagina Web. Ogni pagina viene creata in base alle informazioni inserite nel modulo precedente perciò l'architettura assume una struttura denominata "Step by step". In figura, senza entrare nei dettagli tecnici-implementativi, viene fornito uno schema generale della costituzione del sistema e dell'interazione fondamentale tra i moduli.

INTERFACCIA AMMINISTRATORE:

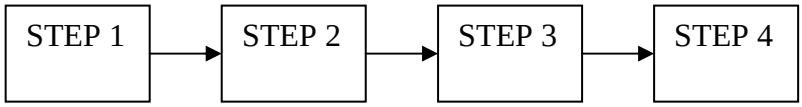

Figura 3.21 Interfaccia amministratore

INTERFACCIA UTENTE:

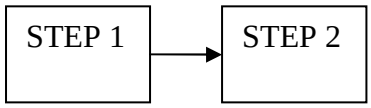

Figura 3.22 Interfaccia utente

3.2.2 Mantenimento della sessione

Come già accennato nella prefazione al paragrafo, al fine di mantenere informazioni rilevanti per la gestione del sistema, sull'identità dell'utente che interagisce è necessario stabilire una sessione di comunicazione. Questa viene instaurata solamente per quanto riguarda l'interazione utente-sistema e non è prevista per l'interfaccia amministratore-sistema poiché l'applicazione software è stata progettata con l'intento e con le limitazioni fisiche che impongono la creazione e gestione di una sola prova di esame nello stesso momento. Al fine del mantenimento della sessione, si è provata in primo luogo, una implementazione che facesse ricorso ai cookies. Il sistema al momento dell'iscrizione creava un cookie denominato "esame-cookie" nel client connesso, salvando come valori i dati inseriti al momento dell'iscrizione ovvero nome cognome e numero di matricola. Durante la fase di accesso al sistema, il server provvedeva a richiedere la lista dei cookies del client e controllava se l'utente fosse già registrato, nel qual caso mostrava un messaggio di benvenuto per confermare l'avvenuto riconoscimento. Nel caso contrario provvedeva ad una nuova iscrizione. Tale scelta progettuale è stata però abbandonata per aumentare la portabilità poiché non tutti i sistemi abilitano ricezione ed invio dei cookies. Questa scelta deriva da considerazioni pratiche su questioni di violazione della privacy e riservatezza dei dati. Per evitare l'utilizzo dei cookies si è scelto di stabilire una sessione al momento dell'iscrizione che provoca l'aggiornamento di una lista mantenuta dal server in cui sono registrati tutti gli utenti iscritti al momento. Come nel caso dei cookies un utente già registrato viene riconosciuto se tenta nuovamente l'iscrizione ed il sistema mostra un messaggio di benvenuto per confermare l'avvenuto riconoscimento. Nel passaggio da una pagina Web all'altra le informazioni sull'utente connesso vengono mantenute passandole come parametri della richiesta del protocollo HTTP. In questo modo l'utente viene sempre riconosciuto finché mantiene la sessione. Nel caso in cui l'utente interrompa la sessione ad esempio visitando una pagina al di fuori del sistema, le informazioni vengono perse e l'utente non è più riconosciuto. Per accedere nuovamente in maniera corretta al sistema sarà necessario ripetere la fase di iscrizione che permetterà al sistema di riconoscere l'utente. Molta importanza è rivestita

dall’operazione di riconoscimento dell’utente in quanto ad ogni client connesso viene riservato dal sistema uno spazio personale sul server in cui vengono salvati i file e le informazioni testuali consegnate. Gli svantaggi di tale tecnica rispetto all’utilizzo dei cookies risiedono però nel fattore sicurezza per quanto riguarda l’autenticazione dell’accesso, in quanto la conoscenza dei dati personali ovvero nome, cognome e numero di matricola di un altro utente permettono con un po’ di abilità e di conoscenza del protocollo HTTP di accedere alle zone personali, ciò comporta una violazione della sicurezza in quanto permette ad un utente che non dispone dell’autorizzazione necessaria di accedere e quindi modificare i contenuti personali di un altro utente. Il sistema risulta perciò da questo punto di vista, senza ulteriori protezioni di mascheramento della sessione o di criptaggio dei parametri piuttosto vulnerabile.

3.2.3 Interazioni

L'analisi del protocollo utilizzato per l'applicazione e dei messaggi scambiati dai moduli verrà ora separata per maggiore chiarezza espositiva tra interfaccia amministratore e interfaccia utente.

INTERFACCIA AMMINISTRATORE

Lo schema presentato in precedenza dell'interazione tra moduli viene ora perfezionato e dettagliato.

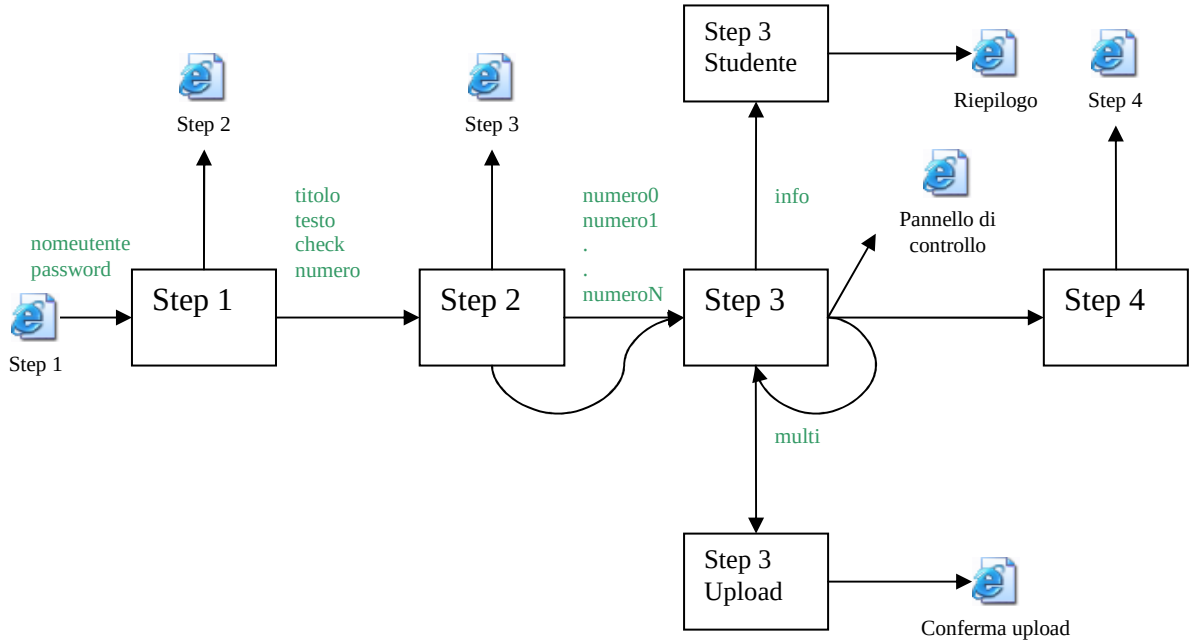

Figura 3.23 Interfaccia amministratore

La prima pagina Web rappresentata come "Step 1" è una pagina Web statica che permette l'inserimento di utenza e password per l'accesso dell'amministratore al sistema. Questi parametri evidenziati nella figura in verde, vengono quindi passati al primo modulo del software che genera la pagina dinamica "Step 2".

Al secondo passaggio i parametri rilevanti sono:

- titolo: rappresenta il titolo definito per la prova di esame
- testo: il testo contenuto nell'area di testo
- check: assume il valore ON se è stata selezionata la checkbox, altrimenti vale OFF
- numero: rappresenta il numero di file che si è stabilito dovranno essere forniti in upload

La freccia di collegamento curva rappresenta l'operazione di redirect effettuata dallo "Step 2" nel caso non siano stati richiesti file da inviare in upload, ovvero quando il parametro "numero" assume valore zero. In questo caso la richiesta del nome con cui rinominare i file inviati non deve essere effettuata, perciò il secondo modulo non genera la pagina Web che gli compete, ma reindirizza il sistema direttamente al passo successivo. Nel caso contrario in cui il parametro "numero" assume valore maggiore di zero, viene generata la pagina "Step 3" impiegata per raccogliere i parametri "numeroi" dove i é compreso tra zero e il valore immesso per il parametro "numero"; ogni parametro "numeroi" rappresenta il nome con cui devono essere rinominati i file inviati tramite upload. A seconda dell'azione richiesta il modulo nominato "Step 3" può generare un riepilogo della situazione di uno studente selezionato; salvare sul server un file inviato in upload oppure terminare la sessione corrente di esame. Nel caso in cui si selezioni l'opzione di upload di un file il modulo "Step 3 Upload" si fa carico di salvare il file sul server e di generare una pagina che informa l'amministratore dell'esito dell'operazione. Il parametro che riceve è una richiesta di tipo multipart-request che include il file inviato. Se si seleziona l'opzione di riepilogo il modulo invia il parametro "info" che contiene informazioni riguardanti nome, cognome e numero di matricola dello studente di cui si chiede il riepilogo. Il modulo "Step 3 Riepilogo" accede ai dati riguardanti lo studente indicato dal parametro "info" e genera una pagina contenente il riepilogo dello stato delle consegne da parte dello studente. Se si agisce sul pulsante termina il sistema effettua le copie di backup della sessione di esame corrente, elimina i file temporanei e crea la pagina "Step 4" che permette di accedere al backup effettuato.

La struttura ricorsiva del modulo “Step 3” evidenziata in figura dalla freccia che si richiude sullo stesso blocco è stata aggiunta per meglio esprimere il fatto che il pannello di controllo deve essere continuamente aggiornato affinché possa fornire dati realistici sullo svolgimento della prova di esame.

INTERFACCIA UTENTE

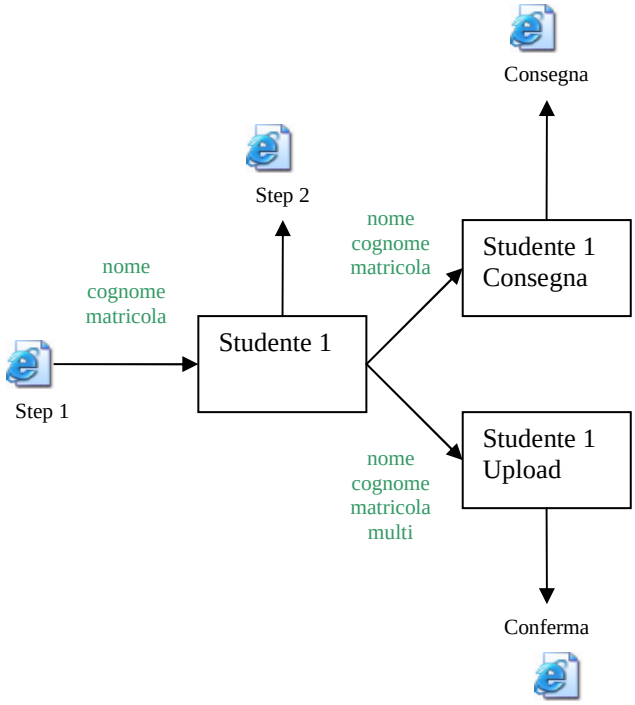

Figura 3.24 Interfaccia utente

La pagina denominata “Step 1” rappresenta la pagina statica iniziale che permette di iscriversi alla sessione di esame corrente inserendo nome, cognome e numero di matricola. Questi valori sono passati sottoforma dei parametri rispettivamente “nome”, “cognome” e “matricola” al primo modulo che realizza l’interfaccia utente. Tale modulo si occupa di generare la pagina Web dinamica “Step 2” in base alle informazioni definite durante la creazione della prova di esame. In particolare visualizza l’area di testo unicamente nel caso in cui sia stata richiesta; crea tanti form per l’upload quanti ne sono stati indicati. Verifica inoltre se l’utente che sta accedendo è già registrato per la sessione di esame corrente, producendo due messaggi diversi a seconda che l’utente compaia o no nella lista degli studenti presenti e si occupa di registrare l’utente nel caso in cui non lo sia. Come già illustrato nel paragrafo

precedente, al fine del mantenimento della sessione i vari moduli dell'interfaccia utente si scambiano ogni volta informazioni riguardo l'utente connesso al sistema tramite l'uso dei parametri "nome", "cognome" e "matricola". Il modulo "Studente 1 Consegna" si occupa del salvataggio sul server del contenuto dell'area di testo, mentre il modulo "Studente 1 Upload" si fa carico di salvare sul server il file inviato tramite upload. Entrambi i moduli generano una pagina di conferma nel caso in cui l'operazione sia andata a buon fine.

3.3 Tecnologia

Il sistema è stato sviluppato utilizzando il linguaggio di programmazione Java. Il software è infatti costituito da un insieme di classi che implementano delle servlet ed alcune classi accessorie utilizzate unicamente al fine di semplificare la modularizzazione del sistema. Per la compilazione delle servlet è stato necessario introdurre il package esterno “servlet.jar”, non compreso nella normale distribuzione del compilatore Java. Per l’upload dei file viene inoltre fatto uso di un package esterno reperito sul Web. Per compilare il software è necessario che il server sia opportunamente configurato, ovvero deve essere installata una versione aggiornata di JSDK (Java software development kit) opportunamente configurata affinchè possa avere accesso ai package esterni introdotti. Per maggiore flessibilità inoltre una delle classi accessorie cui precedentemente accennato, permette di specificare i percorsi e le directory da utilizzare sul server e l’indirizzo IP del server senza che tali informazioni debbano essere inserite in ogni modulo del software. Per l’esecuzione del software occorre un Web server che supporti le servlet o un Web server associato ad un contenitore di servlet. Anche JSDK offre un tool per eseguire e testare servlet, ma non per gestire richieste HTTP.

3.3.1 La scelta di Java

Java appena è uscito è stato accolto con molto entusiasmo dalla comunità mondiale dei progettisti di software e dei provider di servizi Internet, questo perché Java permetteva agli utenti di Internet di utilizzare applicazioni sicure e indipendenti dalla piattaforma, che si possono trovare in qualsiasi punto della rete. Java è quindi nato come linguaggio per la rete, per affiancare l' Hyper Text Markup Language (HTML), il quale non è un linguaggio di programmazione vero e proprio, ma un linguaggio basato sull'uso di tag (marcatori), e per dargli quella sicurezza che l'HTML non ha. Da quando è nato Java, sulla rete si è iniziato a poter parlare di numeri di carte di credito e di informazioni sicure, notizia che ha molto affascinato le grosse società mondiali, le quali hanno trasformato la vecchia Internet, rete utilizzata unicamente dalle università e dai centri di ricerca, nell'attuale mezzo di comunicazione aperto a tutti. Il linguaggio di programmazione Java è stato creato verso la metà degli anni novanta, è il più recente tra i linguaggi dello stesso tipo, e per questo è ancora in fase evolutiva, tanto che ogni anno circa ne viene rilasciata una nuova release.
Da linguaggio nato solo per la rete è divenuto un vero e proprio linguaggio di programmazione, paragonabile, dal punto di vista delle funzionalità, al ben noto C++. Java e la maggior parte degli altri linguaggi possono però essere paragonati solo dal punto di vista delle funzionalità, perché sono fondamentalmente molto diversi, infatti Java compila i sorgenti dei suoi programmi in un codice detto Bytecode, diverso dal linguaggio della macchina su cui è compilato, mentre linguaggi come il C++ compilano i sorgenti dei programmi in un codice che è il codice della macchina (dove per macchina si intende l'insieme di computer e sistema operativo) su cui è eseguito. Al Java viene sempre associato infatti il motto: compile once, run everywhere ovvero è necessario compilare il codice una sola volta affinché possa essere eseguito ovunque. Quindi per eseguire un programma Java occorre avere uno strumento che è chiamato Java Virtual Machine, la quale interpreta il bytecode generato dal compilatore Java e lo esegue sulla macchina su cui è installato. Grazie alla Java Virtual Machine Java è indipendente dalla piattaforma, infatti il programma compilato Java è legato alla JVM e non al sistema operativo, sarà quindi possibile eseguire lo stesso programma Java, compilato una sola volta su una qualunque macchina con un compilatore Java versione

X, su una piattaforma Windows e su una piattaforma Linux, per fare questo però c'è bisogno che sia Windows che Linux abbiano installato una Java Virtual Machine che supporti la versione X di Java. Le due JVM installate sulle due piattaforme diverse sono lo stesso programma compilato una volta per Windows ed una volta per Linux, come avveniva con i programmi scritti in linguaggi come il C/C++. Una Java Virtual Machine è implementata anche nei vari Browser (Come Netscape e Explorer) per poter eseguire i programmi Java incontrati nella rete, le applet. Questo però, unito al fatto che Java ancora si evolve, causa degli ovvi problemi di incompatibilità, infatti capita sempre che il più moderno Browser supporti una versione precedente di Java rispetto all' ultima versione rilasciata dalla Sun Microsystem, inoltre bisogna tener presente che non tutti gli utenti di Internet navigano usando l'ultima versione di Netscape o di Internet Explorer. Quindi volendo creare un'applet ed inserirla in un nostro documento HTML, dobbiamo tenere presente questi problemi, e cercare di fare un programma che sia compatibile con la maggior parte delle JVM inplementate nei vari browser. Considerato il target del sistema realizzato però, ovvero l'utilizzo del sistema in laboratori dotati di macchine connesse in rete si suppone che tutte le macchine abbiano la stessa configurazione e che siano opportunamente impostate per l'esecuzione di programmi Java. In questo caso perciò i vantaggi introdotti dalla portabilità di Java favoriscono la diffusione del sistema su qualsiasi tipo di piattaforma. Il pacchetto software è stato sviluppato facendo uso del JSDK Java™ 2 SDK (Software Development Kit), Standard Edition versione 1.4.0 distribuita gratuitamente dalla Sun Microsystems™, Inc. Un altro problema che introduce Java è la lentezza di esecuzione delle applicazioni, infatti, come già detto, il linguaggio è interpretato, quindi le istruzioni Java prima di essere eseguite dalla macchina vengono tradotte dalla JVM, ovvero per eseguire ogni istruzione il computer eseguirà un numero di istruzioni macchina che è più del doppio delle istruzioni che eseguirebbe se la stessa istruzione fosse stata scritta in C. Allo stesso modo, questo problema non influenza il tipo di applicazione sviluppata, in quanto i requisiti in fatto di velocità richiesti sono minimi e le classi hanno una struttura semplice e compatta che non richiede un elevato tempo di esecuzione.

3.3.2 Le servlet

Le servlet sono moduli scritti in Java che estendono le funzionalità del server, ovvero in questo caso Web server, nell'architettura client/server con metodo request/response. Essendo scritte in Java possono essere inserite nella maggior parte di essi grazie all'indipendenza dalla piattaforma come precedentemente illustrato. Le servlet vengono utilizzate al posto dei normali script CGI, e risiedono sul lato server. Esse forniscono un modo per generare dinamicamente pagine Web.
In effetti le servlet sono un' ottima alternativa alle usuali CGI, in quanto consentono di generare dinamicamente pagine Web che sono sia facili da scrivere che veloci da visualizzare. Grazie alle servlet, dal lato client si puo' interagire con database remoti attraverso il protocollo HTTP con gli usuali metodi di POST e GET. Vediamo ora dei possibili usi delle servlet:

- Supportare richieste multiple in modo concorrente, come per esempio conferenze on-line
- Aggiornare, eliminare o consultare dati contenuti in un database remoto tramite protocollo TCP/IP
- Applicazioni di e-commerce
- Fare dei forward ad altre servlet su altri server, allo scopo di bilanciare il carico di lavoro

La differenza principale tra le servlet e le comuni CGI è che le servlet non richiedono la creazione di un nuovo processo, consentendo un utilizzo più efficiente della memoria e del processore. Tutto ciò grazie alla presenza della Java Virtual Machine che carica dinamicamente una nuova classe, in poche parole la servlet diventa parte integrante del server Web.

3.3.3 Test

In seguito all'implementazione sono state svolte diverse attività di test e di debug. Il software è stato testato per alcune ore su una macchina dotata di server Jackarta–Tomcat versione 4.1 per verificarne la stabilità e l'assenza di errori nel codice. L'applicazione è stata provata con i browser più diffusi: Internet Explorer versione 6.0 e Netscape Navigator 7.02. Dopo aver apportato le necessarie modifiche per risolvere i problemi riscontrati il software è stato impiegato per una simulazione di creazione e gestione di prova di esame su una piccola rete locale costituita da quattro macchine connesse tramite un hub. La prova ha fornito i risultati previsti. Semplici prove di carico sono state effettuate al fine di determinare la stabilità del sistema al crescere del numero di studenti che accedono al sistema. I risultati delle prove garantiscono che il sistema non presenta anomalie nel funzionamento quando il numero degli studenti non supera venti. In realtà anche se non ancora testato, l'architettura del sistema dovrebbe essere in grado di sostenere un numero qualsiasi di iscrizioni a patto che la macchina che ospita il server abbia spazio sufficiente da dedicare ad ogni utente.

3.4 Alcuni dettagli sulle classi

Senza scendere eccessivamente nel particolare verranno ora mostrati alcuni dettagli implementativi del software descrivendo le classi, le varie funzionalità, i protocolli utilizzati. Tra le classi principali vengono considerate quelle che realizzano ciascuno step, ovvero quelli che in precedenza sono stati definiti moduli. Si parlerà quindi di classi accessorie utilizzate per semplificare la struttura dell'architettura software in modo da rendere le componenti principali il più semplici possibile oppure per integrare funzionalità mancanti nei package utilizzati. Due sono i package esterni impiegati nel funzionamento del software: il pacchetto "servlet.jar" sviluppato da Sun Microsystems™ per la gestione delle servlet e delle richieste tramite protocollo HTTP; e il pacchetto "cos.jar" impiegato per gestire richieste di tipo multipart utilizzate nell'upload dei file, tale pacchetto è stato sviluppato da Jason Hunter.

3.4.1 Classi principali

Anche in questo caso risulta conveniente scindere la trattazione in due parti: una relativa all'interfaccia amministratore e l'altra relativa all'interfaccia utente.

INTERFACCIA AMMINISTRATORE

Di seguito viene riportato un semplice diagramma UML delle classi che introduce le classi principali e ne mostra le interazioni e i messaggi scambiati.

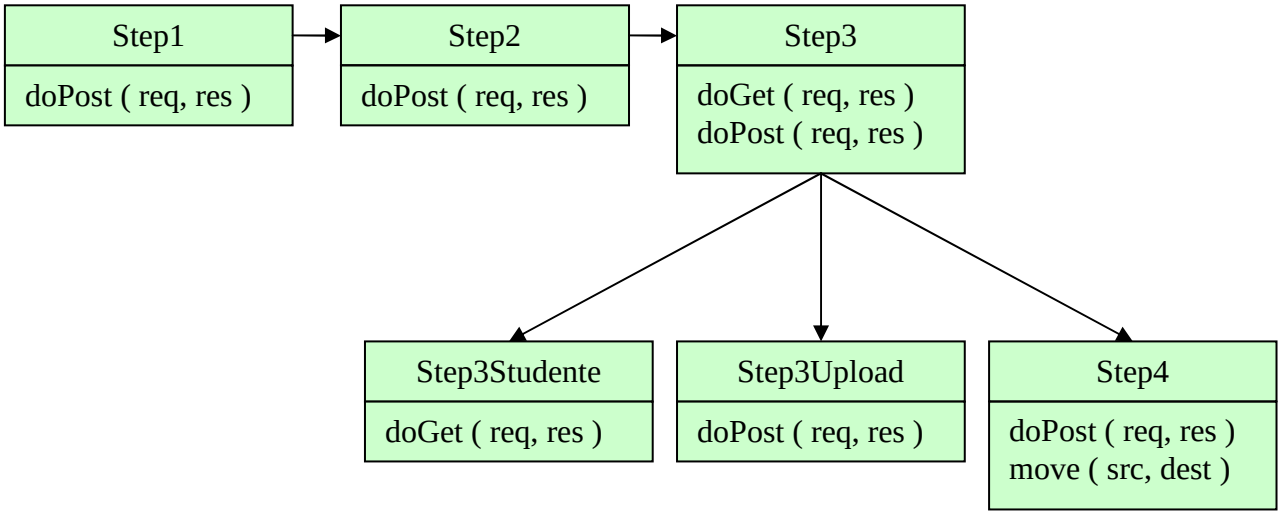

Figura 3.25 Interfaccia amministratore

In figura non è stato dichiarato il tipo dei parametri formali unicamente per questioni di chiarezza grafica. I parametri verranno maggiormente dettagliati man mano che verranno introdotti. Come evidente dal diagramma si è fatto principalmente uso del metodo "POST" per la gestione delle richieste HTTP. Nello "Step3" si sono condensate in un'unica servlet sia le funzionalità della creazione del pannello di controllo che quelle di conferma dell'eliminazione di un file dal server gestite rispettivamente dal metodo "GET" e dal metodo "POST". I parametri formali dei

metodi doGet (…) e doPost (…) ovvero quelli che sono stati indicati come “req” e “res” sono oggetti istanza rispettivamente delle classi HttpServletRequest e HttpServletResponse, definite nel package “servlet.jar” che gestiscono messaggi di richiesta e di risposta secondo il protocollo http. Il metodo move (…) della classe “Step4” rappresenta un metodo di appoggio utilizzato internamente dalla classe per spostare un file dal percorso di origine indicato tramite la stringa “src” al percorso di destinazione, indicato con la stringa “dest” per eliminare i file temporanei e creare la copia di backup.

INTERFACCIA AMMINISTRATORE

La struttura dell’interfaccia utente risulta maggiormente compatta come si può ben intuire dal seguente diagramma UML delle classi.

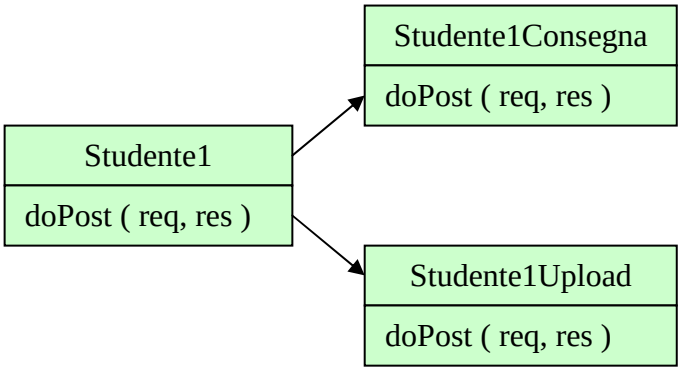

Figura 3.26 Interfaccia utente

3.4.2 Moduli accessori

Due sono i moduli accessori che vengono utilizzati per semplificare o estendere le funzionalità del software. La prima classe “Config” con chiaro significato del nome, permette di ridurre i tempi di ricompilazione in quanto permette di specificare il path per i file sul server e l’ndirizzo IP del server. Tali informazioni sono necessarie a tutti i moduli del sistema e possono perciò essere ricavati direttamente dalla classe di configurazione senza dover essere impostati in ogni modulo. Questa tecnica contribuisce quindi a migliorare la portabilità introdotta dal Java permettendo di facilitare e di abbreviare l’installazione del software su sistemi diversi.

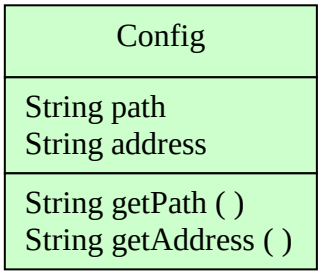

Figura 3.27 Classe Config

Le variabili “path” e “address” sono variabili di classe e vengono appunto utilizzate per mantenere informazioni sul percorso utilizzato per accedere e creare file sul server e sull’indirizzo del server. I metodi pubblici getPath (…) e getAddress (…) vengono richiamati dai vari moduli per ottenere le informazioni contenute nelle variabili di classe. La classe “PersonalFileRenamePolicy” serve ad estendere le funzionalità offerte dal package “cos.jar” impiegato per l’upload dei file. Tale package permette di istanziare oggetti della classe “MultipartRequest” che gestiscono le richieste di tipo multipart-request del protocollo HTTP utilizzate per l’invio dei file. Uno dei costruttori della classe “MultipartRequest” permette di definire anche una politica di ridenominazione dei file da utilizzare nel caso si intenda salvare un file con lo stesso nome di un file già esistente. Tale politica deve essere realizzata come una classe che implementa l’interfaccia “FileRenamePolicy”. Il package fornisce una classe di default

chiamata “DefaultFileRenamePolicy” che incrementa un contatore e lo aggiunge al termine del nome del file, prima dell’estensione, ogni volta che rinomina un file perché ne esiste già uno con il medesimo nome. Nell’upload di file da parte dell’interfaccia amministratore non viene utilizzata nessuna politica di ridenominazione in quanto si assume che possa esistere un’unica versione di un file con lo stesso nome. Per quanto riguarda l’interfaccia utente viene utilizzato un oggetto istanza della classe “PersonalFileRenamePolicy” che oltre alla funzionalità di contatore incrementale della classe di default aggiunge la possibilità di modificare il nome del file e l’estensione. Inoltre la classe offre anche le funzionalità per creare uno storico dei salvataggi effettuati generando un file in cui vengono memorizzati in ordine cronologico, riportando l’ora del salvataggio, il nome originale del file inviato tramite upload e il nome con cui è stato rinominato. La classe viene schematizzata come segue:

PersonalFileRenamePolicy
String name String ext String path String filename int currentCount
PersonalFileRenamePolicy (String name, String path) File rename (File f) boolean createNewFile (File f)

Figura 3.28 Classe PersonalFileRenamePolicy

Le variabili d’istanza “name” ed “ext” servono a memorizzare il nuovo nome che dovrà essere associato al file, “path” il percorso in cui salvare sul server, “filename” il nome originale associato al file inviato tramite upload e “currentCount” il valore attuale del contatore utilizzato per rinominare il file. Il costruttore come accennato permette di definire il nome con cui dovranno essere rinominati i file tramite il parametro “name” e con il parametro “path” imposta la variabile d’istanza omonima. Il

metodo principale è il metodo rename (…) che restituisce a partire da un file passato come parametro, il riferimento ad un file salvato secondo la politica di ridenominazione. Il metodo createNewFile (…) è un metodo di appoggio utilizzato dal metodo rename (…) per verificare se esiste già un file con lo stesso nome di quello passato come parametro. Restituisce “true” se tale condizione non si verifica.

3.4.3 Librerie

Come già accennato in precedenza il software sviluppato deve alcune delle sue funzionalità all'utilizzo di package esterni. Il primo dei due è il package "servlet.jar" che contiene a sua volta i package "javax.servlet" e "javax.servlet.http". Tale package può essere scaricato gratuitamente dal sito della Sun Microsystem e va installato nella directory dove sono contenute le librerie del JRE. Il package contiene tutte le definizioni relative alle servlet in due classi abstract predefinite: GenericServlet ed HttpServlet, che forniscono delle implementazioni di default dei metodi dell'interfaccia Servlet; in particolare HttpServlet fornisce una implementazione per richieste e risposte di tipo HTTP. In pratica la classe HttpServlet definisce il metodo init (…) invocato dal Web browser al momento della richiesta di attivazione della servlet in modo che discrimini tra le richieste di tipo POST o di tipo GET invocando i metodi opportunamente definiti doPost (…) e doGet (…). Il parsing della query string viene effettuato automaticamente e i parametri con i loro rispettivi valori possono essere acceduti utilizzando il metodo getParameter (…) a cui viene passato tramite una stringa statica il nome del parametro e il metodo ne restituisce il valore. Il metodo getParameter (…) è uno dei metodi d'istanza dell'oggetto "req" già citato precedentemente, istanza della classe HttpServletRequest. Tale oggetto viene passato come parametro formale sia al metodo doGet (…) che al metodo doPost (…) ed offre funzionalità avanzate e ad alto livello per la gestione di richieste con protocollo HTTP. Allo stesso modo l'oggetto "res" passato come parametro ai metodi doGet (…) e doPost (…), referenzia una istanza della classe HttpServletResponse che offre funzionalità per la gestione dell'output della servlet, in particolare permette di accedere allo stream di uscita o di effettuare il redirect verso un'altra servlet o una pagina Web. La libreria "cos.jar" sviluppata da Jason Hunter e distribuita gratuitamente previa accettazione dei termini definiti nella licenza viene considerata come recita lo stesso autore "a must have class library for servlet developers". Include classi per la gestione del parsing dei parametri, gestire upload di file, gestire connessioni tramite socket, per agire come server RMI, classi per permettere alle applet di comunicare con le servlet, per permettere alle servlet di inviare messaggi tramite posta elettronica o per verificare la presenza delle API per le servlet.

Capitolo 4: Conclusioni e sviluppi futuri

Il sistema sviluppato nella sua prima versione offre funzionalità piuttosto limitate strettamente relative alla creazione e gestione delle prove di esame tramite interfaccia Web. Di fatto molti sono i miglioramenti che possono essere introdotti e che faranno parte di una eventuale futura versione del software. Dal punto di vista dell'efficienza il miglioramento più importante si può ottenere ricercando una gestione maggiormente oculata, delle informazioni che utilizza il sistema per la gestione delle prove di esame. Si può quindi pensare di utilizzare dei database per la memorizzazione dei dati in maniera più ordinata piuttosto che in file di testo come avviene ora. Il vantaggio del database risiede nel fatto che è una struttura compatta ed estremamente versatile in quanto permette non solo una catalogazione efficiente ed efficace dei dati, ma tramite l'utilizzo di linguaggi come ad esempio SQL è possibile formulare delle query di ricerca per ottenere con semplicità informazioni in altro modo non correlate, riguardanti un particolare studente o una parte della prova o per ottenere diversi ordinamenti delle informazioni ad esempio per ordine alfabetico o per voto decrescente e così via. Si può avere così ad esempio una migliore visione generale dell'andamento della prova di esame in punti che possono essere risultati più critici rispetto ad altri. L'utilizzo del database dovrebbe quindi essere riservato in primo luogo per memorizzare la lista degli studenti iscritti allo svolgimento della prova di esame con i loro dati: nome, cognome e numero di matricola. Inoltre dal punto di vista dell'amministratore del sistema potrebbe risultare comodo avere una visione d'insieme delle prove di esame svolte in un certo periodo di tempo perciò è possibile salvare anche le informazioni relative alle copie di backup effettuate al termine di ogni esame in un ulteriore database. Come già accennato risulta quindi utile l'utilizzo del database per la memorizzazione dei risultati delle prove. A questo proposito se una prova di esame si articola in più parti, ad esempio è preceduta da un esonero o una prova scritta, è possibile introdurre una interazione con i risultati della prova precedente al fine di mediare i voti delle due o più prove, e di calcolare secondo parametri stabiliti dall'amministratore il voto complessivo da associare alla prova di esame. Un'altra innovazione, che risulta piuttosto immediata, consiste nel permettere al sistema di interagire con altri software in modo da estendere le proprie funzionalità. A titolo di

esempio si potrebbe pensare di far in modo che il sistema permetta di importare liste di iscrizione agli esami attraverso la rete Internet. Un miglioramento che solitamente può essere attribuito a qualsiasi sistema software è la ricerca di una maggiore flessibilità. Anche in questo caso la creazione di una prova di esame prevede uno schema piuttosto rigido orientato ad una certa serie di applicazioni, ad esempio a prove di esame che richiedono testi e file generici. Si può pensare perciò di estenderne le funzionalità aggiungendo una serie di opzioni che lascino maggiore libertà nella creazione dell'interfaccia utente. Ad esempio si potrebbe voler creare un questionario a risposte multiple misto a risposte aperte brevi. Questa struttura non è prevista dal sistema e deve perciò essere implementata. Dal punto di vista dell'interfaccia utente l'applicazione risulta piuttosto completa a parte la mancanza di flessibilità nella sua generazione di cui parlato in precedenza. Si ritiene infatti che tale interfaccia debba essere mantenuta il più semplice e compatta possibile al fine di non creare incomprensioni o di esulare dalle finalità del sistema che dal punto di vista dell'utente risiedono unicamente nella possibilità di sostenere prove di esame facendo uso dell'interfaccia Web. Risulta quindi necessario limitare al minimo interazioni che non siano strettamente connesse a questo obiettivo per non distrarre lo studente dal sostenimento della prova. Alcune versioni future del sistema verranno eventualmente rilasciate per garantire maggiore stabilità ed aumentare il fattore sicurezza in un ambito come questo in cui risulta particolarmente rilevante garantire l'autenticazione dell'utente, la correttezza dei dati inviati e l'impossibilità di interagire per gli utenti ammessi a sostenere la prova di esame. Si effettueranno quindi anche controlli per rilevare errori nel codice ed ulteriori test magari con prove di carico più impegnative.