



Università degli Studi di Salerno

Corso di Gestione Progetti Software

A.A. 2012/2013

Alfonso Murolo

Giulio Franco

Linda di Geronimo



System Design Document

Storia delle revisioni

Versione	Data	Autori	Descrizione
0.1	6/11/2012	Giulio Franco	Prima unione di bozza

Riferimenti

Indice

1	Introduzione.....	1
1.1	Scopo del sistema.....	1
1.2	Obiettivi di design.....	1
1.3	Trade off.....	4
2	Architettura proposta per il software.....	5
2.1	Decomposizione in sottosistemi.....	5
2.1.1	Sottosistema accessi.....	6
2.1.2	Sottosistema Management.....	7
2.1.3	Sottosistema Feedback.....	8
2.2	Layer e partizioni.....	10
2.3	Topologia del sistema.....	10
2.3.1	Diagramma dei componenti.....	11
2.3.2	Diagramma di deployment.....	12
2.4	Implementazione del controllo software.....	13
2.4.1	Flusso di controllo esterno.....	13
2.4.2	Flusso di controllo interno.....	13
2.4.3	Controllo della concorrenza.....	13
2.5	Mapping tra hardware e software.....	13
2.6	Gestione dei dati persistenti.....	14
2.6.1	Schema concettuale.....	14
2.6.2	Specifiche delle entità.....	14
2.6.3	Progetto delle transazioni.....	14
3	Sicurezza.....	15
3.1	Sicurezza e controllo degli accessi.....	15
3.2	Performance del sistema.....	19
3.3	Condizioni di Boundary.....	19
3.3.1	Inizializzazione.....	19
3.3.2	Terminazione.....	20
3.3.3	Fallimento.....	20
4	Servizi dei sottosistemi.....	22
5	Glossario.....	23

1 Introduzione

L'obiettivo del documento SDD è quello di far fronte a tutti i problemi del sistema proposto fornendo un'architettura vantaggiosa, che possa rendere il sistema più efficiente possibile. A tale scopo si specificheranno quali caratteristiche e requisiti dovranno essere migliorati e definiti. Si forniranno così informazioni sulla configurazione hardware e software del sistema.

1.1 Scopo del sistema

Il sistema "At-silo" nasce con l'obiettivo di migliorare ed ottimizzare il servizio di asilo nido messo a disposizione dall'università di Fisciano, inoltre una particolare caratteristica del sistema riguarda la possibilità di interagire, in modo intuitivo, da parte del genitore sia per consultare informazioni personali riguardanti il proprio figlio, e sia per consultare i servizi messi a disposizione tramite l'accesso al sistema previa identificazione. A tale scopo il software richiede la presenza di requisiti minimi per prestazioni e portabilità. In particolare:

Adattabilità e Portabilità: avendo come requisito non funzionale l'Usabilità, le interfacce dovranno essere intuitive e semplici su ogni sistema software e hardware utilizzato, per questo motivo verranno utilizzate le tecnologie JDBC per quando riguarda la connettività ai server e la loro gestione tramite il package Servlet e il protocollo HTTP per lo scambio di informazioni via WEB tra client e server.

Tempi di Risposta e Prestazioni: avendo come requisito non funzionale le Performance, le interrogazioni al Database devono risultare rapide, inoltre il sistema dovrà favorire il requisito di robustezza, ossia la resistenza all'immissione di input non validi. altri requisiti non funzionali da sviluppare in dettaglio saranno affidabilità, sicurezza e modularità per gli utenti. Tenendo conto dei costi prefissati, il tutto si riassume in un'unica tecnologia MySQL.

1.2 Obiettivi di design

Gli obiettivi di design rappresentano, in un prodotto software, le basi del successivo sviluppo del prodotto, perché, su di esse, si fondano le scelte prese durante la fase di implementazione.

Di seguito, sono presentati gli obiettivi del progetto @silo, ordinati secondo importanza decrescente, categorizzati in base agli utenti del sistema cui essi si applicano.

Committente

Sicurezza e tutela della Privacy

Data la criticità di alcuni aspetti burocratici come la gestione del bando, la pubblicazione di quest'ultimo, i pagamenti delle rette e i questionari anonimi di valutazione dei servizi offerti dall'asilo, il committente ritiene importante un grado elevato di sicurezza dei dati memorizzati dal sistema. In quanto i dati inseriti nel sistema fanno parte di informazioni strettamente personale, come ad esempio i dati relativi ai pagamenti delle rette che avvengono in via telematica, oppure riguardanti persone con problemi legati al mondo dell'handicap.

Adattabilità e Portabilità

Il sistema deve poter garantire le stesse funzionalità in browser differenti e su architetture hardware diverse.

Genitore

Sicurezza e tutela della privacy

Il genitore per accedere al sistema deve compilare un form di registrazione. Il sistema deve dare al genitore la sicurezza e l'affidabilità nell'inserimento dei propri dati sensibili, sia in campo di sicurezza web, sia nel caso del rispetto delle leggi in vigore sulla visibilità e sul trattamento dei dati personali. Qualora le informazioni venissero rese pubbliche, il sistema notifica questo evento al proprietario dei dati personali.

Tempo di risposta

Gli utenti del sistema compiono giornalmente delle operazioni. Il sistema si occupa quasi esclusivamente di interrogazioni al database, i genitori, quindi, consultano e modificano gli elenchi, dopo aver eseguito operazioni di login. Questo tipo di operazioni, login e consultazione/modifica, seppur oneroso per il database di grande dimensioni, non può quindi occupare più di qualche secondo per produrre risultati. In altre parole il tempo di attese dei genitori è di pochi secondi. Inoltre il sistema deve permettere al genitore di poter avere conferma dell'avvenuta iscrizione in non più di 5 secondi.

Usabilità

Il genitore è uno di quegli stakeholders del sistema “At-Silo” che ha le maggiori difficoltà nell'utilizzo del sistema informatico. L'interfaccia con cui il genitore ha un'interazione con il sistema, dovrà fare un massiccio uso di metafore, permettendo di eseguire operazioni come la compilazione di un questionario valutativo, senza l'inserimento di dati già noti al sistema, o come precedentemente sono stati già inseriti, al momento dell'autenticazione. Questo requisito è fondamentale per permettere al genitore di eseguire operazioni, con maggiore voglia, e di conseguenza di valutare l'asilo, in maniera più accurata e precisa.

Adattabilità e Portabilità

I genitori che accedono ad “At-Silo” fanno uso di un browser, che permette la navigazione tra i vari contenuti del sistema. Indipendentemente dal browser e dal sistema utilizzato dai genitori, il sistema deve essere funzionante e coerente con il design modellato.

Tolleranza

I crash di sistema devono essere ridotti al minimo. In caso di manutenzione ci devono essere avvisi che avvertano in tempo che il sito sarà in manutenzione e quindi non sarà possibile usare il servizio.

Personale asilo

Adattabilità e portabilità

Il personale dell'asilo formato dal personale amministrativo, personale formazione, delegato rettore e responsabile scientifico, indipendentemente e dal sistema utilizzato, il sistema dev'essere funzionante e coerente con il design modellato. Il sistema essendo progettato per essere eseguito su macchine diverse e utilizzato anche tramite browser, è adattabile altri sviluppi hardware e software.

Usabilità

Attraverso una semplice interfaccia web, il personale dell'asilo formato dal personale amministrativo, personale formazione, delegato rettore e responsabile scientifico, potrà facilmente e velocemente apprendere il funzionamento del sistema. Gli utenti devono poter effettuare vari operazioni come la visualizzazione delle risposte dei genitori ai questionari valutativi in meno otto click.

Affidabilità

In caso di scadenze burocratiche come la scadenza del bando dell'iscrizione all'asilo Mazzetti, e la relativa gestione e la pubblicazione della relativa graduatoria, come la scadenza di un pagamento di una retta mensile o come una scadenza di una registrazione ad una eventuale gita scolastica, il sistema deve sempre funzionare ed essere disponibile evitando così l'impossibilità ad effettuare questo tipo di operazione. In altre parole il sistema non deve essere soggetto a fallimento ma essere tollerante agli errori prevenendo quelli che fanno riferimento ad inserimento di dati necessari alla compilazione corretta di un form, perdita di connessione al database, perdita di connessione alla rete(Internet), avvertendo comunque l'utente con delle notifiche inerenti all'errore verificatosi.

Facilità di apprendimento

Attraverso una semplice interfaccia grafica il personale dell'asilo potrà facilmente e velocemente apprendere il funzionamento sistema e di tutte le funzionalità.

Tolleranza

I crash di sistema devono essere ridotti al minimo.

1.3 Trade off

Interfaccia vs usabilità:

L'interfaccia del prodotto At-silo è composta da oggetti molto comprensibili all'utente che vanno a chiarire immediatamente la propria funzione. L'interfaccia è composta da schede, pulsanti e varie etichette, associate all'oggetto, che fanno intendere la loro utilità.

Sicurezza vs Efficienza:

La gestione della sicurezza viene affidata all'utilizzo del login iniziale in quanto va ad autenticare l'utente al quale sarà visualizzata solo la parte del software che gli appartiene, evitando così incongruenze di dati. Questa politica di permessi, permette di non appesantire eccessivamente il software ed è un buon compromesso tra sicurezza ed efficienza.

Comprensibilità vs Tempo:

Il codice deve essere più comprensivo possibile in modo da poter essere interpretato da altri programmatori che non hanno partecipato al progetto. Una seconda motivazione è anche per non accrescere la difficoltà dello sviluppo nella fase di testing. Il codice sarà commentato in modo da migliorare la lettura delle righe di codice anche se l'aggiunta di commenti accrescerà il tempo necessario per completare l'implementazione.

Spazio di Memoria vs Velocità:

Il prodotto dovrà memorizzare informazioni inerenti alle differenti entità riscontrate, essenzialmente il carico complessivo dei dati non influirà sulla velocità del sistema. Oltre modo le operazioni delle funzionalità implementate richiederanno un brevissimo tempo di risposta.

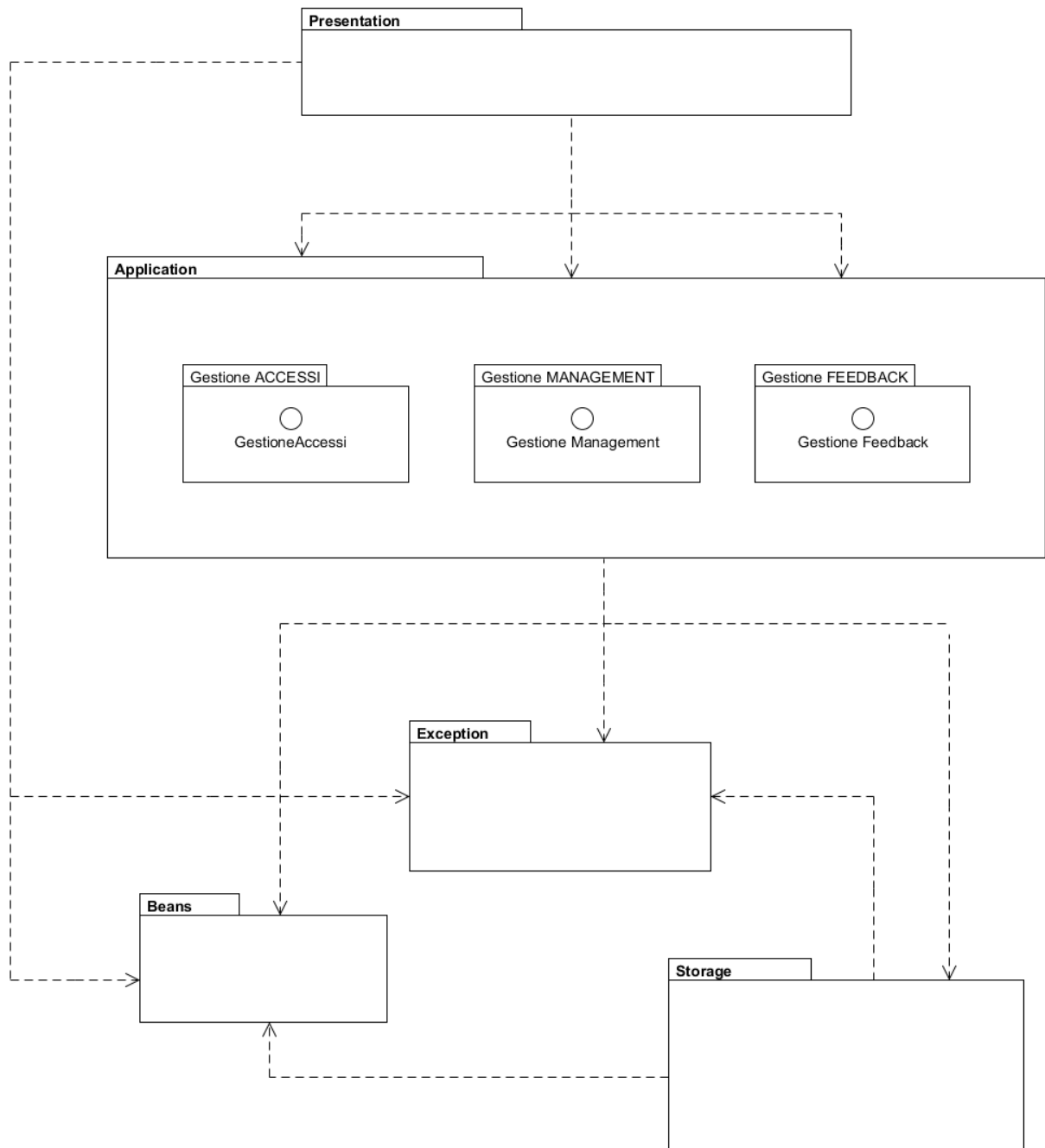
I costi per un disco fisso sono poco onerosi quindi si è scelto di dare più rilevanza alla velocità rispetto che allo spazio. La scelta di un DBMS rispecchia questa decisione in quanto i dati persistenti richiedono più spazio sul disco ma la velocità in lettura e in scrittura è molto alta.

Tempo di Rilascio vs Qualità:

Le scadenze sono parte intrinseca del progetto, il nostro sistema garantirà oltre al rispetto delle date di consegna anche la qualità giusta delle funzionalità descritte e successivamente implementate.

2 Architettura proposta per il software

2.1 Decomposizione in sottosistemi

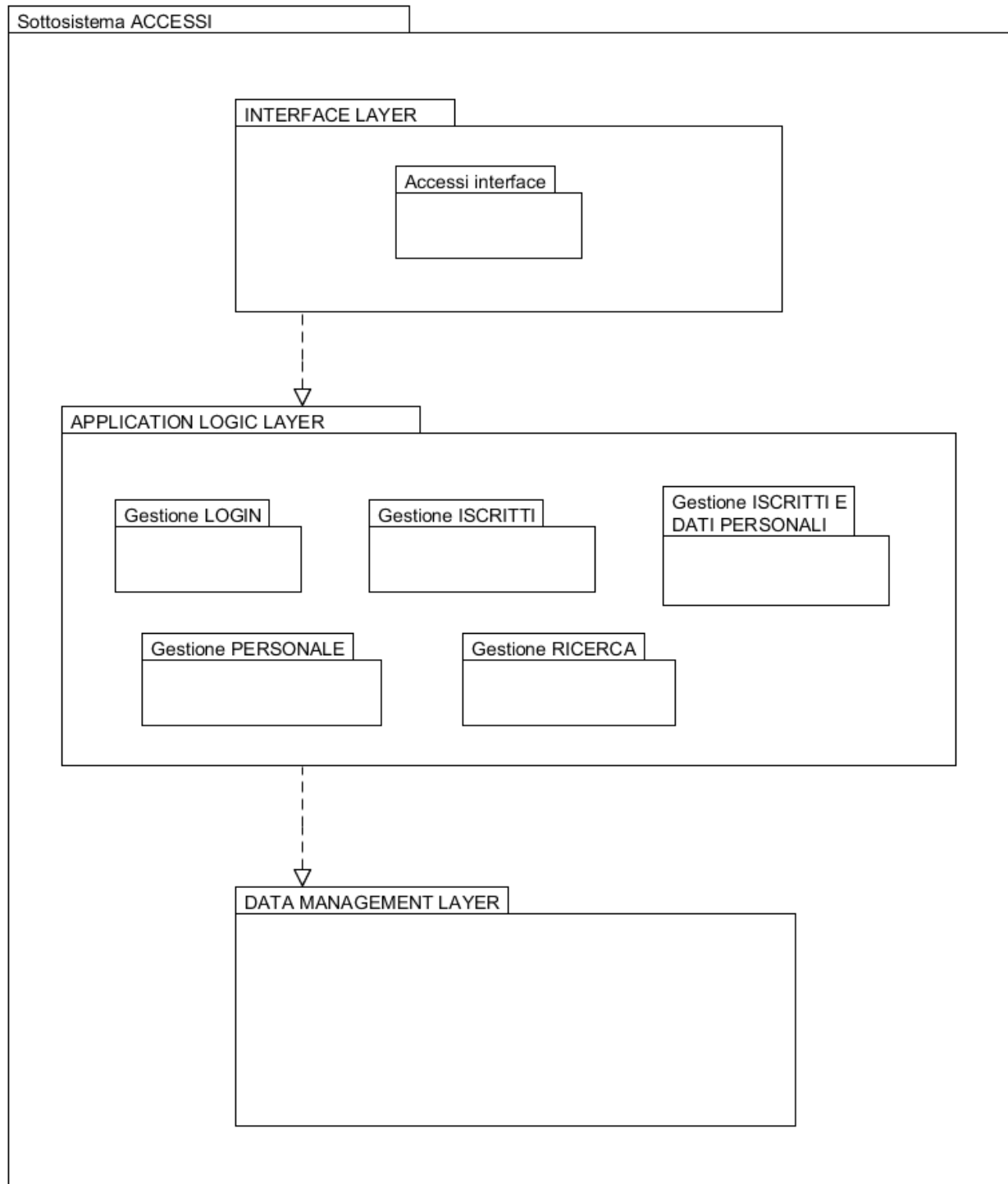


La decomposizione prevista per il sistema è composta da cinque sottosistemi che si occupano di gestire aspetti e funzionalità differenti:

1. **Presentation**: raccoglie i sottosistemi adibiti alla gestione delle interfacce grafiche;
2. **Application**: si occupa della gestione della logica applicativa del sistema;
3. **Beans**: si occupa della gestione e dello scambio dei dati tra i sistemi;

4. **Storage**: sistema che gestisce ed immagazzina i dati persistenti:
5. **Exception**: gestione delle eccezioni del sistema.

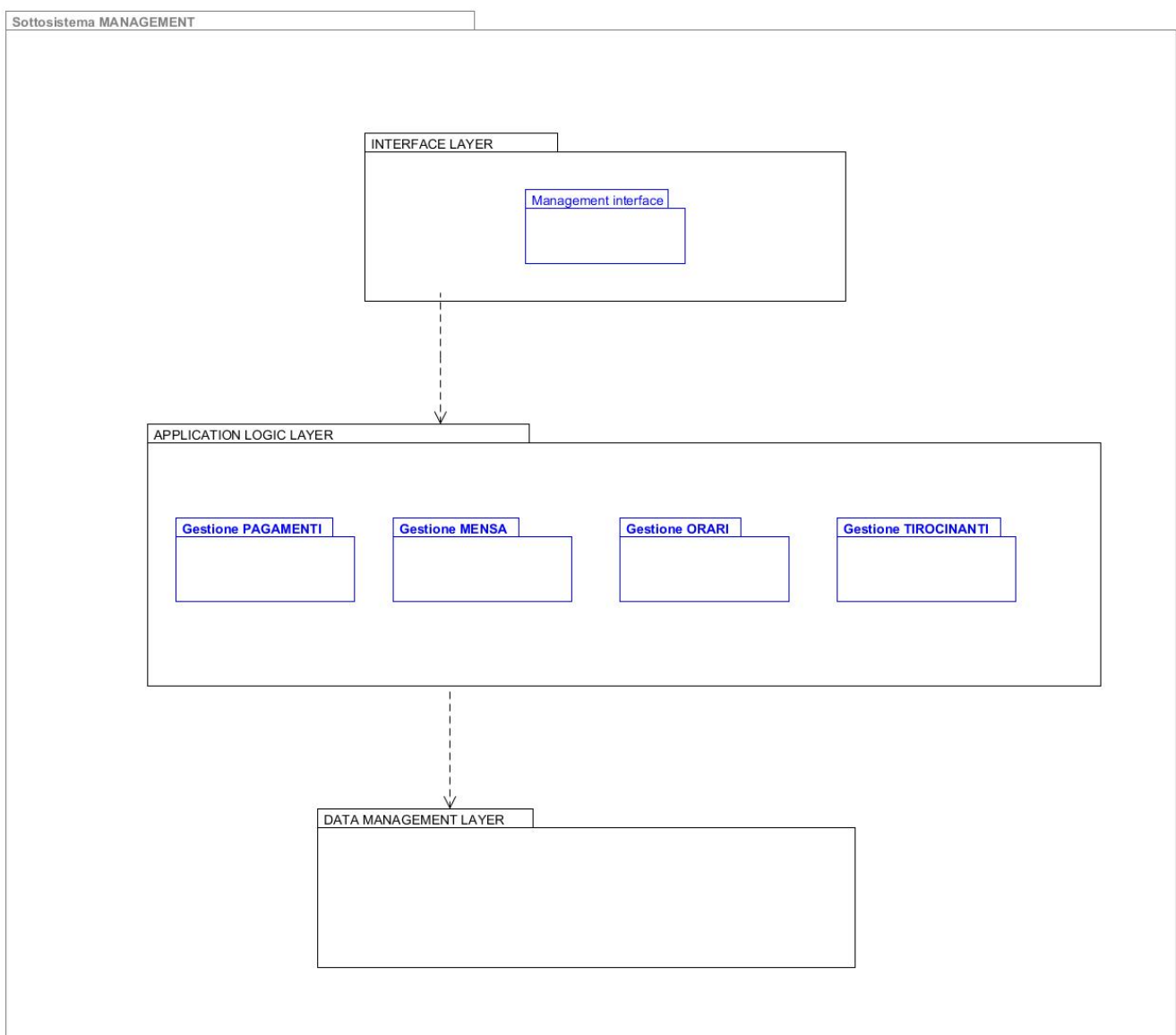
2.1.1 Sottosistema accessi



- Interface layer
 - **Accessi Interface**: interfacce grafiche con cui i vari utenti gestori del servizio interagiscono con il sistema.
- Application Logic Layer

- **Gestione Login:** modulo che si occupa di gestire il controllo del login da parte degli utenti
- **Gestione Iscritti:** modulo che si occupa di gestire l'iscrizione degli utenti al sistema e di controllarne lo stato
- **Gestione Iscritti e Dati Personali:** modulo che permette di gestire (aggiungere, cancellare, modificare, visualizzare) i dati e le informazioni relative agli utenti del sistema
- **Gestione Personale:** modulo che si occupa di gestire gli utenti che fanno parte del personale asilo (modifica, cancellazione, aggiunta)
- **Gestione Ricerca:** modulo che si occupa di gestire, attraverso dei parametri, la ricerca e la visualizzazione dei dati appartenenti a utenti del sistema
- Data Management layer

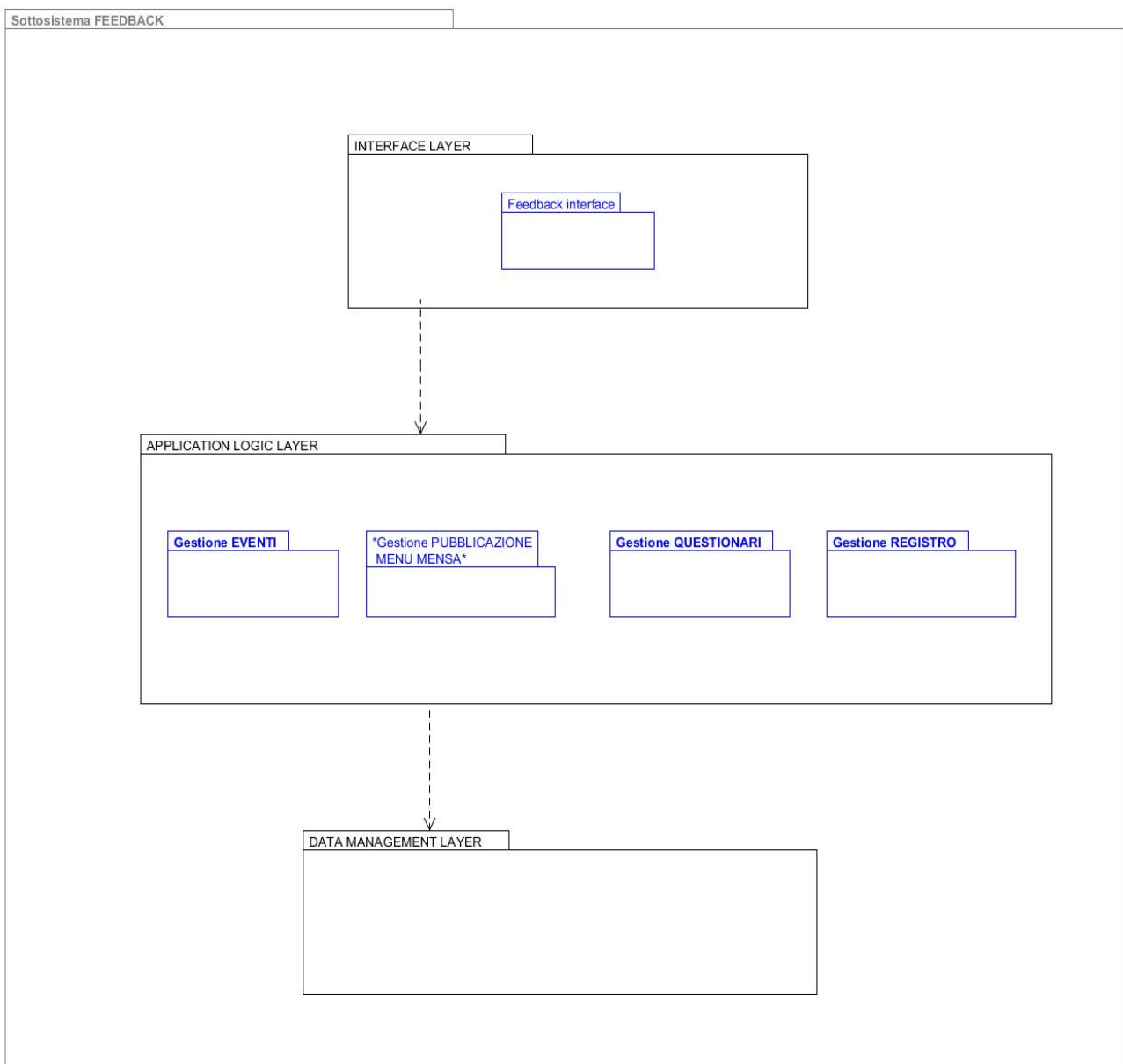
2.1.2 Sottosistema Management



- Interface layer
 - **Management Interface:** interfacce grafiche con cui i vari utenti gestori del servizio interagiscono con il sistema.

- Application Logic Layer
 - **Gestione Pagamenti:** modulo che si occupa di gestire il controllo e la visualizzazione dei pagamenti effettuati dagli utenti
 - **Gestione Mensa:** modulo che permette di richiedere ed effettuare modifiche al piano-pasto dei bambini.
 - **Gestione Orari:** modulo che permette di richiedere ed effettuare modifiche alla fascia oraria di utenza dei bambini.
 - **Gestione Tirocinanti:** modulo che si occupa di gestire il periodo di tirocinio di studenti di Scienze della Formazione.
- Data Management layer

2.1.3 Sottosistema Feedback

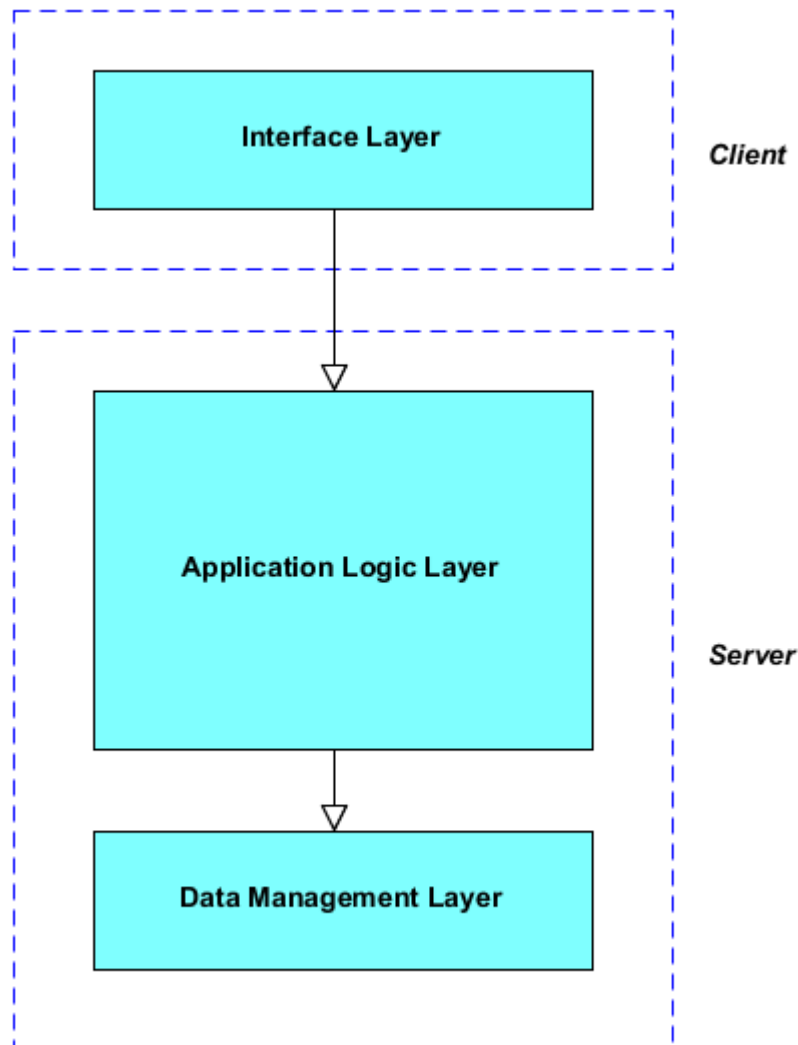


- Interface layer
 - **Feedback Interface:** interfacce grafiche con cui i vari utenti gestori del servizio

interagiscono con il sistema.

- Application Logic Layer
 - **Gestione Eventi:** modulo che si occupa di gestire il controllo e la visualizzazione degli eventi creati dagli utenti
 - **Gestione Pubblicazione Menu Mensa:** modulo che permette ad uno specifico utente di manutere il menu della mensa , inserendolo ,modificandolo e cancellandolo. Inoltre permette la visualizzazione del menu a tutti gli utenti.
 - **Gestione Questionari:** modulo che permette di gestire i questionari, inserendoli, modificandoli per un particolare utente, compilarli e visualizzare le statistiche per altri utenti.
 - **Gestione Registro:** modulo che si occupa di gestire in maniera computerizzata un registro standard cartaceo. Permette anche all'utente psico-pedagogo di visionare le attività svolte dall'educatore.
- Data Management layer

2.2 Layer e partizioni



L'organizzazione dei sottosistemi segue la logica **three-tier**. Si possono dunque individuare i seguenti layers:

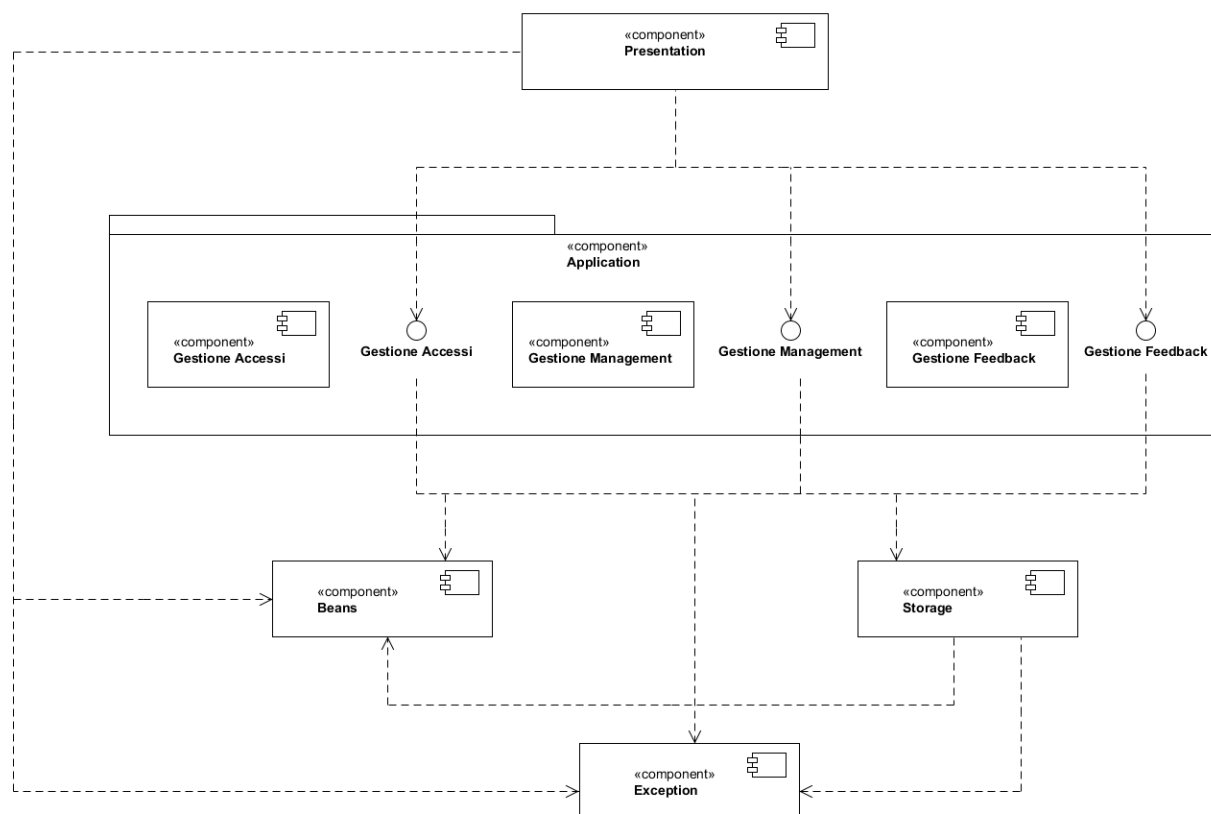
- **Interface Layer:** lato client, è il layer caratterizzato dalle interfacce grafiche che permettono all'utente di interagire col server;
- **Application Logic Layer:** lato server, è il layer che contiene e gestisce le operazioni necessarie a compiere i servizi che l'utente richiede;
- **Data Management Layer:** lato server, è il layer che ospita e gestisce le entità persistenti (database e relativo DBMS).

La scelta è ricaduta sull'architettura three-tier allo scopo di gestire con facilità ed indipendentemente i sistemi di elaborazione dati e quelli relativi all'interfaccia grafica (una modifica al livello presentation non presenterà complicazioni in altri sistemi).

2.3 Topologia del sistema

Di seguito verrà descritta la topologia dell'architettura prevista per il sistema.

2.3.1 Diagramma dei componenti



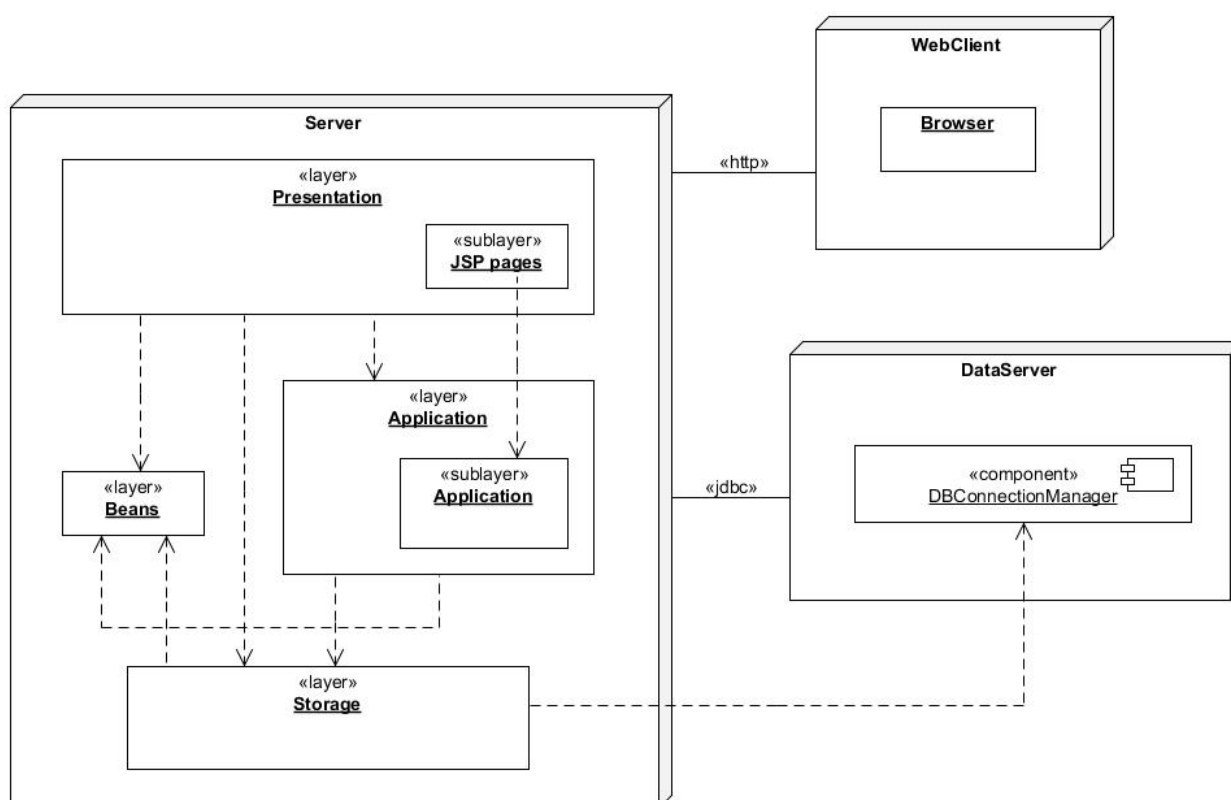
Il componente Presentation è composto dalle sottocomponenti relative alle funzionalità del sistema @silo, ciascuna delle quali contiene gli script e i fogli di stile che gestiscono le interfacce grafiche per le funzionalità corrispondenti.

Il componente Application è formato da sottocomponenti contenenti le classi e i vari moduli che controllano le funzionalità del sistema.

Il componente Beans contiene tutte le classi bean del sistema e le classi che gestiscono le collezioni di tali dati.

Il componente Exceptions contiene le eccezioni che possono essere lanciate dal sistema. Infine il componente Storage contiene le classi e i moduli che controllano la comunicazione con il DBMS.

2.3.2 Diagramma di deployment



Lo stile architetturale Client/Server, impone la disposizione dei componenti del sistema su diverse macchine. A tal fine è possibile avere una o più macchine Client che si occupano della presentazione, su interfacce grafiche, dei dati che vengono scambiati nel sistema attraverso i beans . Il Client, inoltre, si occupa di richiedere al server le informazioni che l'utente richiede tramite l'interfaccia. I Client richiedono servizi all' Application Server, questa macchina provvede alla logica applicativa del sistema, gestisce la persistenza dei dati e lo scambio di essi. La comunicazione tra i Client e l' Application server è gestita da HTTP.

L'Hypertext Transfer Protocol (HTTP) (protocollo di trasferimento di un ipertesto), invece, è usato come principale sistema per la trasmissione di informazioni sul web. Le specifiche del protocollo sono gestite dal World Wide Web Consortium. L'HTTP funziona su un meccanismo richiesta/risposta (Client/Server): il Client esegue una richiesta ed il Server restituisce la risposta. Inoltre per gestire le problematiche di sicurezza per le autenticazioni, il Client infatti dovrà autenticarsi a pagine protette per ricevere informazioni, si userà anche il protocollo HTTPS, acronimo di **HyperText Transfer Protocol over Secure Socket Layer**, cioè HTTP con SSL. Le specifiche del protocollo HTTP sono gestite dal W3C, mentre le specifiche del protocollo SSL

Nell'uso comune il Client corrisponde al browser ed il server al sito web. Il DataServer rappresenta la macchina in cui vengono memorizzati i dati, ovvero dove risiede il DBMS del sistema. La comunicazione tra l'Application Server e il Data Server è gestita dalle funzioni PHP di interfacciamento con MySQL. Tali funzioni, che iniziano con il prefisso **mysql_** permettono tutte le operazioni. Per il forum sarà usato l'applicativo **phpBB**, un sistema di gestione forum open source.

2.4 Implementazione del controllo software

2.4.1 Flusso di controllo esterno

Nel sistema @silo il flusso di controllo software sarà distribuito su più Client e più Server.

La logica applicativa sarà implementata sul Server e ai Client verranno mostrati i risultati avuti dall'esecuzione di alcune richieste ricevute dagli utenti.

Per quanto riguarda lo storage dei dati persistenti lo scambio di informazioni e di messaggi tra server e database management system (nel nostro caso mySql) verrà realizzato tramite l'implementazione di un dispatcher JDBC. Ciascuna funzionalità disponibile nel sistema @silo potrà richiamare il dispatcher JDBC per soddisfare le proprie richieste di memorizzazione ,ricerca e modifica dei dati.

2.4.2 Flusso di controllo interno

Il sistema @silo verrà utilizzato da una molteplicità di utenti, quindi esso dovrà garantire efficienza e coerenza per ogni utente di qualsiasi tipologia che si connetta al sistema.

Sui Client sarà eseguita un'applicazione che utilizza un flusso di controllo di tipo Event-driven, in cui il ciclo principale attende un evento esterno; Una volta che l'evento ha luogo, il programma invia all'handler le caratteristiche dell'evento ed esso viene spedito all'oggetto appropriato. Questo tipo di flusso di controllo ci permetterà di sfruttare una migliore efficienza delle interfacce grafiche, le quali risulteranno molto più flessibili ed usabili per gli utenti.

2.4.3 Controllo della concorrenza

Nel sistema @silo si possono avere diversi accessi agli oggetti in base all'utente che sta utilizzando il sistema.

Concorrenza: non sarà possibile effettuare operazioni su un oggetto mentre si stanno manipolando (salvando o modificando) istanze dello stesso tipo. Sarà invece possibile effettuare operazioni su oggetti di altro tipo. Gli utenti potranno concorrere accedendo allo stesso oggetto solo in lettura e non in scrittura in quanto questo potrebbe portare ad un'inconsistenza dei dati. E' necessario che l'accesso in scrittura sia fatto, dai diversi utenti, in sequenza e non in contemporanea, rispettando un certo ordine negli accessi. La gestione di questa concorrenza è gestita dal DBMS, ma anche dalla sincronizzazione degli eventi offerta dal linguaggio di programmazione scelto. Inoltre, ogni utente potrà svolgere una sola operazione per volta. Verrà implementato un controllo per ogni funzionalità del sistema, non sarà ammesso l'utilizzo concorrente di più funzioni all'interno dello stesso Client.

2.5 Mapping tra hardware e software

Il sistema software @silo è distribuito su due nodi, Client e Server.

La configurazione HW necessaria per i Client è la presenza di una connessione ad internet nel caso di WebClient o di una connessione LAN nel caso di client Stand Alone, poiché il sistema @silo necessita di collegamenti remoti al server in cui è contenuto il sottosistema dello storage e quello delle application.

Inoltre i nodi Client necessitano di un sistema operativo e di una macchina virtuale Java (o JVM) per i clienti StandAlone oppure di un browser web per i WEBClient.

Le configurazioni SW dei Client sono dunque le seguenti:

- la macchina usata dai Dipendenti dell'asilo, su cui gira la versione 6.0 di Java Virtual Machine;
- la macchina usata dai Clienti su cui girerà un browser compatibile con HTML 5 (ad esempio Internet Explorer, Safari, Chrome, Opera e Firefox).

I nodi cliente permettono l'accesso al sistema tramite collegamenti remoti per lo scambio dei dati con in nodo server. I collegamenti sono gestiti da componenti off-the-shelf: HTTP è stato utilizzato per collegare il nodo “web” client al nostro server.

La configurazione HW minima del server comprende una connessione ad internet ed una alla rete locale insieme ad un hardware capace di immagazzinare grandi quantità di dati consistenti (HardDisk). Il nodo server, necessita inoltre di un software in grado di gestire le connessioni con più client e di una tecnologia che gestisca i dati persistenti. Per far ciò abbiamo deciso di utilizzare tecnologie già esistenti, quali Apache, che si occupa del lavoro del webServer e Mysql, un DMBS che gestisce i nostri dati persistenti.

Le configurazioni SW del Server sono:

- la macchina su cui risiederà il DMBS (MySQL) con la relativa Base Dati del sistema.
- la macchina su cui risiederà il server Web (Apache) .

Le macchine potranno anche risiedere in diverse parti del server o su due server separati.

Ogni sottosistema risiede nei seguenti modi:

- Presentation: Risiede sul Client del Dipendente e del Cliente.
- Application: Risiede sul Server Web.
- Beans: Risiede sul Server con la base dati
- Storage: Risiede sul Server con la base dati
- Exception: Risiede sul Server con la base dati.

2.6 Gestione dei dati persistenti

2.6.1 Schema concettuale

2.6.2 Specifica delle entità

2.6.3 Progetto delle transazioni

3 Sicurezza

Per garantire un buon standard di sicurezza il software “At-silo” si avvale di diverse protezioni.

Il primo sistema di protezione dati è l'autenticazione. Gli User-ID e password devono essere fornite dall'amministratore per i dipendenti, mentre per i genitori è il sistema che auto genererà il nome utente e la password. Il nome utente resterà uguale mentre per la password se il genitore vuole cambiarla potrà farlo.

Il sistema provvederà a creare gli account necessari ed a dare loro i corretti privilegi. E' quindi impossibile per qualsiasi utente, che sia genitore o personale, manipolare dati che non rientrano nei loro privilegi.

3.1 Sicurezza e controllo degli accessi

Il sistema “At-silo” prevede un'autenticazione che sia stata confermata attraverso la mail ricevuta quando si è fatta la registrazione al sistema.

L'autenticazione avviene attraverso l'inserimento, da parte dell'utente all'interno di una apposita interfaccia di login, di un username e di una password. La verifica della correttezza delle credenziali inserite è verificata attraverso una query al database contenente la lista degli utenti registrati al sistema, dei loro username e della loro password finalizzata a garantire la sicurezza del sistema.

La suddetta interfaccia di login è costituita da un campo di testo per l'username, un campo di testo per la password e un bottone di login [si fa riferimento allo use-case del login].

Ogni utente può appartenere solo ad uno dei gruppi elencati, ed a seconda del gruppo di appartenenza il sistema provvede a fornirgli gli strumenti adeguati, impedendo che possa eseguire operazioni non appartenenti al suo livello utente.

Le politiche di accesso vengono riportate di seguito con lo scopo di rendere chiare le operazioni consentite ad ogni livello utente.

Attore / Entità	Iscrizione	Bando
Genitore	CreaAccount() CancellaAccount() ModicaAccount()	InviaBando() RinunciaCandidatura()
Personale amministrativo		InserisciBando() InserisciSpecificheGraduatoria() ModificaSpecificheGraduatoria() PubblicaGraduatoria() VisualizzaGraduatoria() VisualizzaUtentiPerIdoneità()
Personale formazione		VisualizzaIscritti()
Direttore		InserisciBando() InserisciSpecificheGraduatoria() ModificaSpecificheGraduatoria() PubblicaGraduatoria() VisualizzaGraduatoria() VisualizzaUtentiPerIdoneità()

Attore / Entità	Bambino	Genitore
Genitore	VisualizzaStatoPropriaIscrizione() CancellaIscrizione() RitiroBambino()	VisualizzaPropriDati() VisualizzaDatiFiglio() VisualizzaDatiFigli() ModifcaPropriDati() ModificaDatiFiglio() RicercaAltroGenitore()
Personale amministrativo	VisualizzaIscritti() AccettazioneBambino()	VisualizzaDatiGenitore() VisualizzaDatiFigli() ModifcaDatiGenitore() ModificaDatiFiglio() VisualizzaStatoPagamenti() RicercaGenitore()
Personale formazione	VisualizzaIscritti()	VisualizzaDatiGenitore() VisualizzaDatiFiglio() RicercaGenitore()
Direttore	VisualizzaIscritti() AccettazioneBambino()	VisualizzaDatiGenitore() VisualizzaDatiFigli() ModifcaDatiGenitore() ModificaDatiFiglio() VisualizzaStatoPagamenti() RicercaGenitore()

Attore / Entità	Personale formazione	Personale amministrativo
Genitore	RicercaPersonaleFormazione() VisualizzaSchedaPersFormazione()	
Personale amministrativo	RicercaPersonaleFormazione() VisualizzaSchedaPersFormazione() AggiungiPersonaleFormazione() LicenziaPersonaleFormazione() ModificaDatiPersonaleFormazione()	RicercaPersAmministrativo() VisualizzaPersAmministrativo() ModificaPersAmministrativo()
Personale formazione	RicercaPersonaleFormazione() VisualizzaSchedaPersFormazione() ModificaDatiPersonaleFormazione()	
Direttore	RicercaPersonaleFormazione() VisualizzaSchedaPersFormazione() AggiungiPersonaleFormazione() LicenziaPersonaleFormazione() ModificaDatiPersonaleFormazione()	RicercaPersAmministrativo() VisualizzaPersAmministrativo() ModificaPersAmministrativo() AggiungiPersAmministrativo() LicenziaPersAmministrativo()

Attore / Entità	Registro	ProgrammaEducativoSettimanale
Genitore	VisualizzaStoricoRegistro()	
Psico-pedagogo	VisualizzaStoricoRegistro()	InserisciProgrammaSettimanale() ModificaProgrammaSettimanale() VisualizzaProgrammaSettimanale()
Educatore Didattico	VisualizzaStoricoRegistro() Inserisci() Modifica()	VisualizzaProgrammaSettimanale()
Direttore	VisualizzaStoricoRegistro()	

Attore / Entità	Attività	Questionario
Genitore	VisualizzaAttivitàOdierna()	CompilaQuestionario() VisualizzaListaQuestionari() RicercaQuestionario()
EventPlanner	VisualizzaAttivitàOdierna() AggiungiAttività() ModificaAttività()	VisualizzaListaQuestionari() InserisciQuestionario() ModificaQuestionario() EliminaQuestionario() VisualizzaDettagliQuestionario() VisualizzaStatistiche() RicercaQuestionario()
Educatore Didattico	InserisciAttivitàInRegistro() ModificaAttivitàInRegistro()	VisualizzaProgrammaSettimanale()
Direttore		

Attore / Entità	Evento
Genitore	VisualizzaEvento()
UtenteAutenticato	VisualizzaListaEventi() RicercaEvento()
EventPlanner	InserisciEvento() ModificaEvento() EliminaEvento()
Direttore	

Attore / Entità	Menu
UtenteAutenticato	VisualizzaMenu()
Impiegato	InserimentoMenu() ModificaMenu() EliminaMenu()
EventPlanner	
Direttore	

Attore / Entità	Piano-Pasto	Extra
Genitore	RichiestaModificaPastoSingolo()	
Impiegato Asilo	VisualizzaPianoPasto() ModificaPianoPastoSingolo() ModificaPianoPastoSegreteria()	VariazioniExtra()
Responsabile Scientifico		
Direttore	VisualizzaPianoPasto() ModificaPianoPastoSingolo() ModificaPianoPastoSegreteria()	

Attore / Entità	Orario	Orario Utenza
Genitore	RichiestaModificaOrarioSingolo() ()	
Impiegato Asilo	ModificaOrarioBambino() ModificaOrarioSegreteria() VisualizzaOrario()	
Responsabile Scientifico		
Direttore	ModificaOrarioBambino() ModificaOrarioSegreteria() VisualizzaOrario()	InserisciFasciaOraria() ModificaFasciaOraria() RimuoviFasciaOraria()

Attore / Entità	Richiesta	Promemoria
Genitore		
Impiegato Asilo		InviaPromemoria() PromemoriaPagamento() PromemoriaFattura()
Responsabile Scientifico		
Direttore		InviaPromemoria() PromemoriaPagamento() PromemoriaFattura()

Attore / Entità	Fattura	Pagamento
Genitore	VisualizzaFatturaMensile()	VisualizzaStoricoPagamenti()
Impiegato Asilo		StatoPagamento() RegistraPagamentoManuale() FatturaPagamento()
Responsabile Scientifico		
Direttore		

Attore / Entità	Tirocinante	Feedback Tirocinante
Genitore		
Impiegato Asilo		
Responsabile Scientifico	AggiungiTirocinante() ModificaTirocinante() RimuoviTirocinante() VisualizzaTirocinanti()	FeedbackTirocinanti()
Direttore	VisualizzaTirocinanti()	FeedbackTirocinanti()

Nota: un attore può effettuare tutte le operazioni indicate per gli attori più generali

3.2 Performance del sistema

Per i riferimenti di performance fare riferimento ai casi d'uso dettagliati nel Documento di Analisi dei Requisiti: esso conterrà requisiti qualitativi di performance dettagliati per ciascuno dei casi d'uso modellati.

3.3 Condizioni di Boundary

3.3.1 Inizializzazione

Start-up (primo avvio)

Per il primo start-up del sistema “At-silo” è necessario l'avvio di un web server che fornisca servizi ai WebClient e di un Database MYSQL per la gestione dei dati persistenti, e l'amministratore potrà avviare il server di “At-silo”.

In seguito, tramite l'interfaccia di Login del client, sarà possibile autenticarsi tramite opportune credenziali (login e password) come Amministratore del sistema.

Una volta effettuato l'accesso come Amministratore di sistema, “At-silo” presenterà all'utente il workspace di amministratore, dal quale si possono effettuare tutte le operazioni che il sistema fornisce.

Start-up (sessione generica)

Il dipendente avvia il sistema.

Gli viene presentata l'interfaccia di login del sistema.

Inserendo le proprie credenziali (login e password), all'interno dei campi di testo appositi, effettua l'accesso al sistema. Le interfacce di login sono molto semplici e condividono più o meno gli stessi elementi: due campi di testo e il pulsante di login.

Quando l'utente clicca su quest'ultimo, lato server, il sistema accede al DB e provvede alla verifica della correttezza delle informazioni fornite cercando all'interno del database tra gli utenti registrati al sistema.

Se l'accesso avviene con successo, l'operazione di login viene memorizzata in un file di log.

Un utente genitore avvia il sistema semplicemente utilizzando un browser e accedendo al sito internet di “At-silo” e compilando i due campi di login.

Start-up (a seguito di un fallimento)

Il sistema accede al Database per ricaricare i dati persistenti, in caso di dati inconsistenti, effettua un ripristino all'ultimo backup disponibile, ripopolando il DB con dati corretti.

In seguito, il sistema presenta la finestra di login come un normale Start-up.

3.3.2 Terminazione***Terminazione lato Server***

Il sottosistema che risiede nel Server prevede la terminazione corretta del server. Il sistema una volta terminato non accetterà più richieste da tutti i client.

Terminazione lato Client

Prevede terminazione del sottosistema Client, dal lato Web. Quando viene terminata correttamente la procedura, vengono inviati messaggi agli altri sottosistemi al fine di gestire correttamente i dati in fase di chiusura e salvare quelli modificati durante la sessione ed infine vengono arrestati i flussi di lavoro ancora aperti.

3.3.3 Fallimento

Possono verificarsi diversi casi di fallimento del sistema:

1. Nel caso in cui vengano meno i link di comunicazione tra Client e Server, i dati che non sono stati precedentemente salvati nel DB non verranno ripristinati alla ri-connessione. Quindi il Client sarà tenuto ad effettuare nuovamente lo startUp (sessione generica) e il re-inserimento dei dati non salvati [USE_CASE_A_58].
2. Nel caso in cui si verifichi un'interruzione inaspettata dell'alimentazione (sia lato server che lato client), non sono previsti metodi che ripristinino lo stato del sistema a prima dello spegnimento inaspettato [USE_CASE_A_61].

3. Un altro caso di fallimento potrebbe derivare dal software stesso che causa una chiusura inaspettata dovuta ad errori commessi durante la fase di implementazione [USE_CASE_A_60].
4. Un altro caso di fallimento potrebbe essere dovuto ad un errore critico nell'hardware del server o del client [USE_CASE_A_59].

4 Servizi dei sottosistemi

5 Glossario