



UNIVERSITÀ DEGLI STUDI DI SALERNO



UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA



Summer Center

System Design Document

Top manager

Professore	Tutor
De Lucia Andrea	Pecorelli Fabiano

Partecipanti

Cognome	Nome	Matricola
Dalia	Antonio	0512106118
Donnarumma	Gerardo	0512106064
Esposito	Vincenzo	0512106070

Storico versioni

Data	Versione	Descrizione	Autori
30/01/2021	1.0	Creazione del documento, introduzione, architettura del sistema e servizi dei sottosistemi	Donnarumma Esposito
10/01/2021	1.1	Aggiunta schema relazionale e descrizione	Esposito, Donnarumma
18/02/2021	1.2	Revisione del documento	Tutti

Sommario

Top manager	2
Partecipanti.....	2
Storico versioni	2
Introduzione	4
Scopo del sistema	4
Obiettivi di progettazione	4
Definizioni, acronimi e abbreviati	6
Riferimenti.....	7
Panoramica.....	7
Architettura del sistema	8
Panoramica.....	8
Decomposizione del sistema in sottosistemi	8
Mappatura Hardware/Software.....	10
Scelte progettuali del database:	11
Condizioni limite	13
Fallimento.....	13
Servizi dei sottosistemi.....	14
Interfaccia utente	14
Gestione autenticazione	14
Gestione account	14
Gestione iscrizioni.....	14
Gestione pagamento.....	15
Storage	15

Introduzione

Scopo del sistema

Il sistema proposto ha come scopo la digitalizzazione delle iscrizioni al centro estivo Summer Center. Durante le iscrizioni vengono spesso compilati lunghi moduli, lasciati vuoti campi obbligatori o commessi errori nel formato dei vari campi compilati a mano, perciò il sistema ha l'intento di digitalizzare la sottomissione dei moduli da parte dei genitori. Facendo così i segretari non dovranno passare ore ed ore a ricontrollare i moduli e sistemare tutte le questioni amministrative per permettere l'ingresso dei bambini al centro estivo. Questione non meno importante è il sistema di pagamento, un sistema online, veloce e affidabile.

Il sistema fornirà al genitore la possibilità di iscrivere un bambino e finalizzare subito l'iscrizione con il pagamento oppure di iscrivere un bambino, continuare con altre iscrizioni e pagare successivamente tutte le iscrizioni che si desiderano completare. Quindi il sistema cerca di rendere quanto più semplice possibile la compilazione dei moduli di iscrizione, ma anche di essere quanto più flessibile possibile in modo da poter completare subito le iscrizioni con il pagamento o pagare quando si ritiene più necessario (entro i tempi stabiliti). Il genitore per ogni iscrizione di un bambino potrà scegliere le settimane prenotate anche non sequenziali, permettendo così di non effettuare tante iscrizioni di un singolo bambino se tale bambino dovesse frequentare il centro in periodi differenti.

Il genitore avrà anche la possibilità di disdire una iscrizione, la quale sarà visionata dal segretario per decidere l'eventuale rimborso da applicare. Il responsabile potrà registrare tutti i segretari necessari al sistema e visionare l'andamento delle iscrizioni.

Obiettivi di progettazione

Criteri di performance

- **Tempi di risposta:** il sistema deve consentire una navigazione rapida, minimizzando la latenza di risposta, la quale dovrà essere non superiore a 1 secondo per la prima risposta da dare al client.
- **Memoria:** la maggior parte della memoria necessaria al funzionamento del sistema è data dalla quantità di dati che il centro estivo deve mantenere, ci si aspetta inizialmente uno spazio di memoria occupato intorno alle decine di Gigabytes. Se un domani il sistema volesse mantenere una history dei dati per questioni amministrative o se il numero di utenti dovesse aumentare notevolmente, sarebbe importante utilizzare la tecnica del RAD per aumentare le prestazioni, la sicurezza ed anche la tolleranza contro eventuali guasti. La tecnica consigliata è il RAID 10, poiché in caso di guasti legge solo l'immagine dei dati non danneggiata e ne memorizza una copia nell'unità di sostituzione, lasciando invariate le altre operazioni di lettura e scrittura. Quindi ci offre sia una certa capienza di memorizzazione e sia performance e sicurezza in caso di danneggiamento.
- **Throughput:** il sistema deve consentire un numero di tasks pari all'iterazione di cento utenti connessi simultaneamente, garantendo una certa fluidità e operazioni sempre consistenti. L'architettura del Web Server e il DBMS saranno in grado di gestire accessi concorrenti e operazioni consistenti.

Criteri di affidabilità

- **Robustezza:** il sistema deve essere in grado di gestire gli input errati, informando l'utente di eventuali errori.
- **Affidabilità:** il sistema deve poter garantire affidabilità nei servizi offerti, soprattutto per quanto riguarda il sistema di pagamento delle iscrizioni. (?)
- **Disponibilità:** il sistema deve essere disponibile ogni qualvolta l'utente vuole poter usare il sistema stesso, quindi 24 ore su 24 e 7 giorni su 7. Il software dovrà essere distribuito su più server tramite tecniche di bilanciamento del carico per poter offrire maggiori performance ed essere disponibile nel caso di eventuali guasti. Per reagire ai guasti è anche utile la tecnica del RAID 10 dei dischi rigidi, la quale in tempi notevoli riesce a effettuare le varie copie.
- **Fault tolerance:** il sistema deve essere in grado di operare anche sotto condizioni di errore o malfunzionamenti dovuti ai vari sottosistemi o sistemi esterni con cui s'interfaccia.
- **Sicurezza:** il sistema deve essere in grado di sostenere e gestire gli attacchi di malintenzionati senza subire danni gravosi. Il sistema implementerà il protocollo di comunicazione https per proteggere l'integrità e la riservatezza dei dati scambiati tra gli utenti e il sistema.

Criteri di costo

- **Costi di sviluppo:** è stimato un costo di sviluppo pari a 400 ore per la progettazione e lo sviluppo del software.
- **Costo di deployment:** è stimato un costo di installazione e di training pari a 48 ore.

Criteri di manutenzione

- **Estensibilità:** il sistema deve facilmente garantire l'aggiunta di nuove funzionalità dettate dall'esigenza del centro o degli utenti.
- **Adattabilità:** il sistema è realizzato per un centro estivo specifico, ma potrebbe essere adattato a vari centri senza grosse difficoltà.
- **Portabilità:** il sistema deve essere portabile su varie piattaforme e ciò è permesso anche grazie all'accessibilità del sistema tramite un qualsiasi browser.
- **Leggibilità:** il sistema deve essere comprensibile dalla lettura del codice, cercando di scrivere codice secondo le buone prassi della programmazione ad oggetti.
- **Tracciabilità dei requisiti:** il sistema deve prevedere in modo semplice la mappatura dei requisiti in codice, permettendo modifiche o l'aggiunta di funzionalità in modo agevole.

Criteri di usabilità

- Il sistema deve essere chiaro, comprensibile e semplice da usare, in modo da permettere la digitalizzazione della compilazione dei moduli senza incongruenze ed errori, cercando di informare l'utente quanto più possibile sui percorsi da seguire per rendere semplice l'iscrizione dei bambini al centro.

Definizioni, acronimi e abbreviati

Nome	Descrizione
Summer Center	Nome del sistema da realizzare.
Genitore	Utente che accede al sistema per iscrivere bambini al centro estivo.
Bambino	Non è un utente che accede al sistema, ma rappresenta l'insieme di dati relativi ad un figlio che viene iscritto al centro estivo.
Segretario	Utente che accede al sistema per controllare le iscrizioni e rispondere alle eventuali disdette.
Responsabile	Utente amministratore che accede al sistema per monitorare l'andamento delle iscrizioni al centro.
DBMS	Database Management System
HTML	HyperText Markup Language
HTTPS	HyperText Transfer Protocol over Secure Socket Layer
JS	JavaScript
JSP	JavaServer Pages
MVC	Model View Controller
RAID	Redundant Array of Independent Disks
RAID 10	Un sistema RAID 1+0 è un RAID che viene usato sia per replicare che per condividere dati tra diversi dischi. RAID 10, infatti, legge solo l'immagine dei dati non danneggiati e ne memorizza una copia nell'unità di sostituzione, lasciando invariate tutte le altre operazioni di lettura e scrittura.
TomCat	Apache TomCat è un'implementazione open-source delle tecnologie Java Servlet, JavaServer Pages e Java WebSocket. (fonte: Apache TomCat)
RAD	Requirements Analysis Document

PS	Problem Statement
----	-------------------

Riferimenti

- [Problem Statement](#)
- [Requirement Analysis Document](#)
- [Apache Tomcat® - Welcome!](#)
- [Apache Tomcat 9 \(9.0.43\) - SSL/TLS Configuration How-To](#)
- [RAID - Wikipedia](#)

Panoramica

Questo documento mostra le fasi e i dettagli della progettazione del sistema proposto. corrente mostra i dettagli della progettazione del sistema. Abbiamo illustrato gli obiettivi di design e l'architettura proposta. Le fasi previste nell'architettura proposta sono:

- Decomposizione del sistema in sottosistemi
- Mappatura Hardware/Software
- Gestione dei dati persistenti
- Sicurezza e controllo degli accessi per definire gli accessi e le operazioni permesse da ogni tipologia di utenza
- Controllo del flusso globale di sistema, cioè quali operazioni eseguire ed in che ordine
- Condizioni limite, che descrivono l'avvio e lo spegnimento del sistema.

Infine, abbiamo una sezione dedicata ai servizi dei sottosistemi, dove abbiamo specificato i servizi che ogni sottosistema fornisce.

Architettura del sistema

Panoramica

Il sistema è decomposto in sottosistemi secondo lo stile architetturale Model-View-Control:

- **Model:** rappresentano tutti i sottosistemi a conoscenza della logica applicativa e dell'accesso ai dati.
- **View:** si occupano di rappresentare le informazioni visualizzate dagli utenti e di gestire l'iterazione tra i dati degli utenti e il sistema sottostante.
- **Control:** si occupano di ricevere l'iterazione dell'utente attraverso le View e di eseguire le operazioni che interessano il Model che generalmente portano ad un cambio di stato delle View.

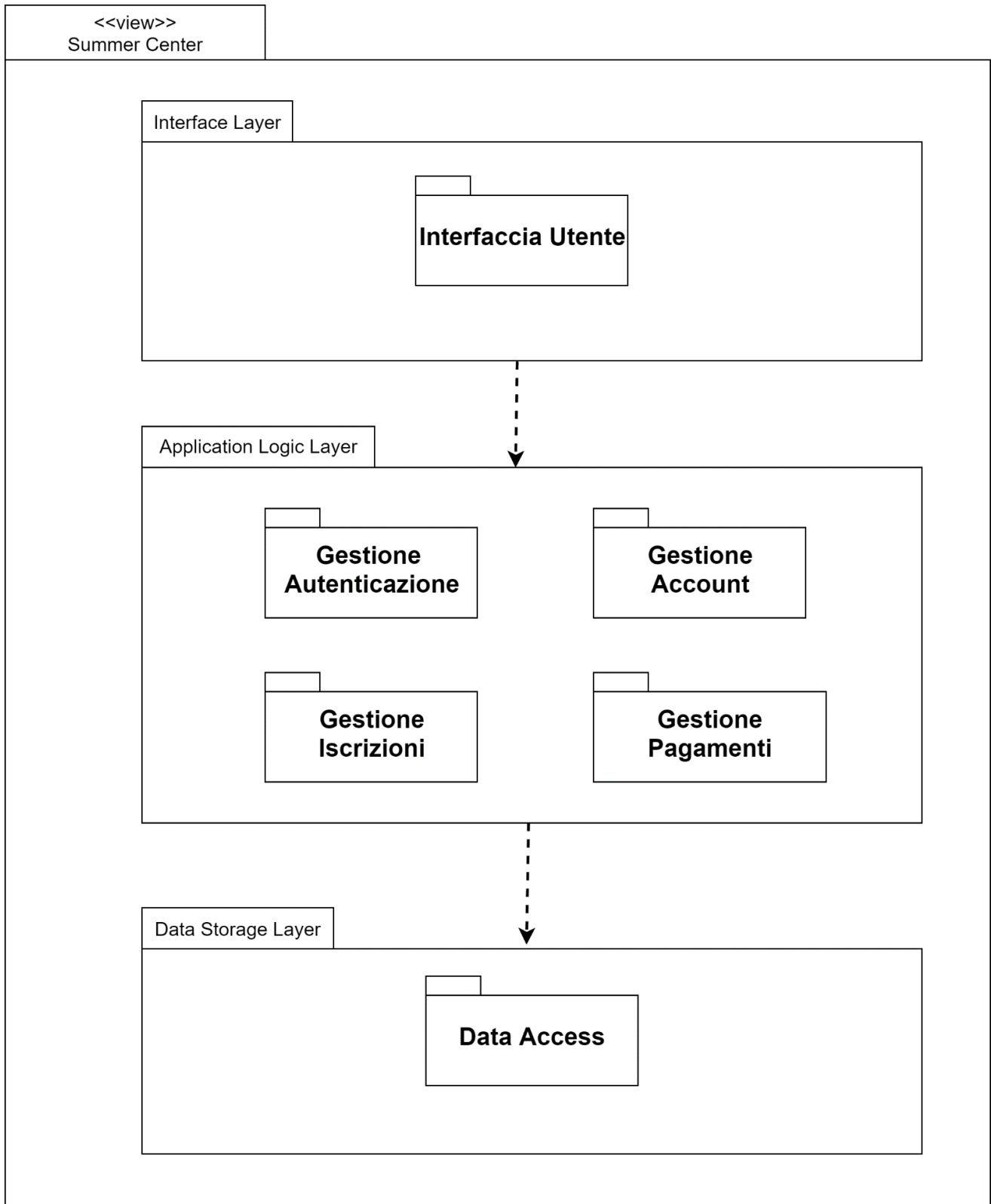
Anche se questo modello potrebbe aumentare la complessità introdotta dall'aggiunta di classi aggiuntive per le separazioni dei componenti, abbiamo comunque deciso di applicare il modello MVC per riusabilità del codice in modo da poter riusare parti di codice per diversi aspetti dell'applicazione e per velocizzare lo sviluppo assegnando ad ogni componente del team una delle parti sopraelencate.

Decomposizione del sistema in sottosistemi

Il sistema è una web application, logicamente separata in tre strati:

- **Presentation Layer**, i client si interfacciano con questo livello per interagire con il sistema. Il Presentation Layer accede all'Application Layer per avere i servizi necessario che i client richiedono.
- **Application Layer**, si occupa di gestire la logica applicativa e fornire i servizi richiesti al Presentation Layer. Inoltre, comunica con il Data Storage per la gestione dei dati persistenti.
- **Data Storage Layer**, si occupa della gestione dei dati persistenti e comunica con l'Application Layer fornendo i servizi richiesti.

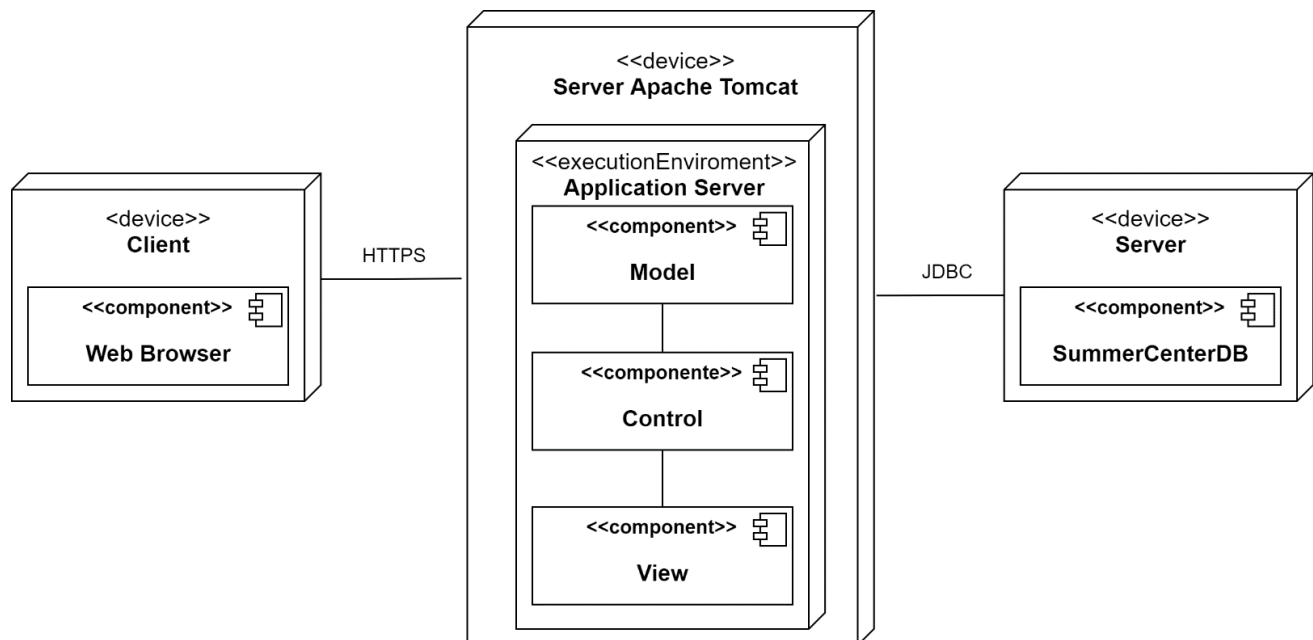
Strutturando così il sistema, potremmo avere tre istanze differenti per lo stesso application server.



Mappatura Hardware/Software

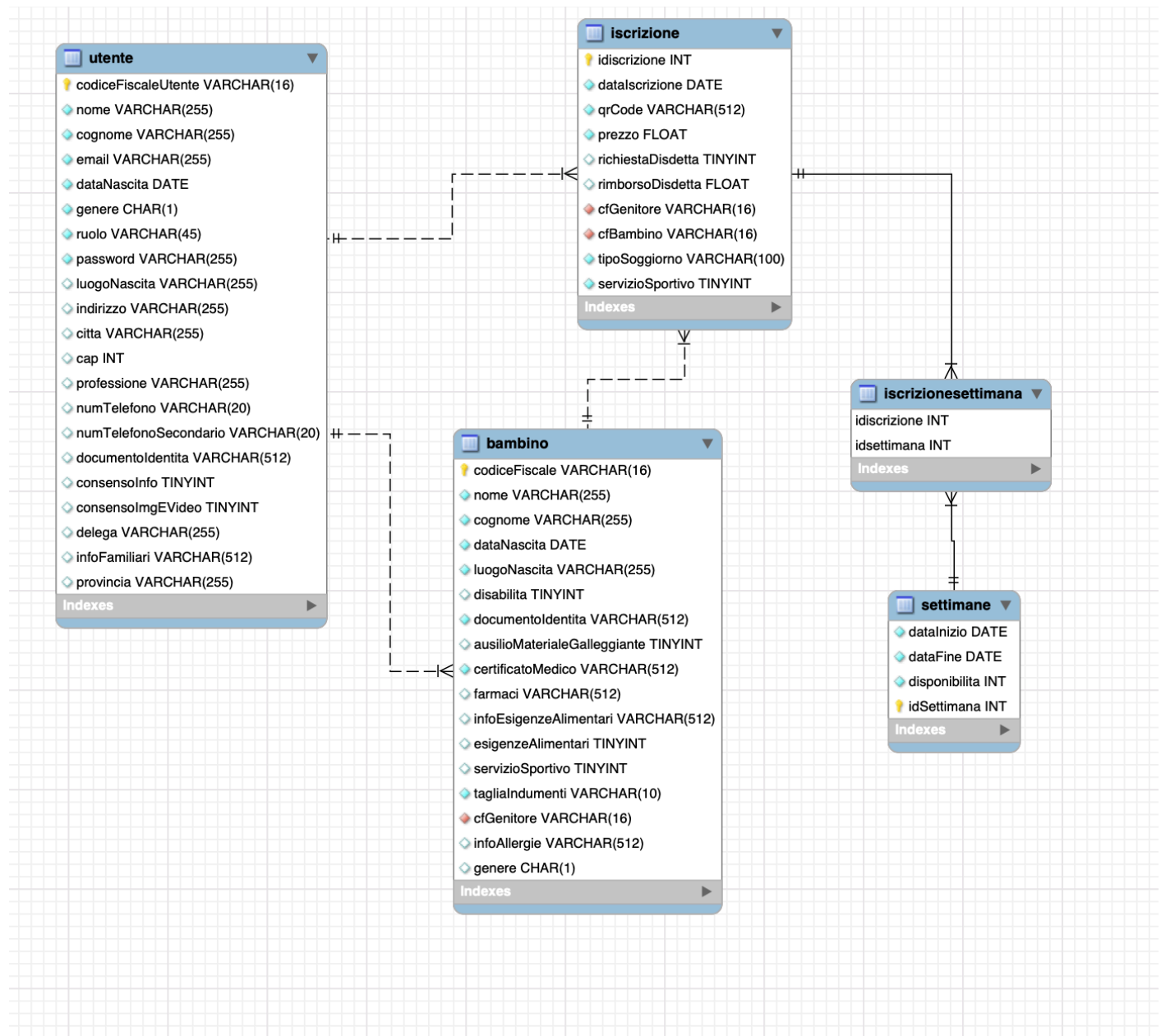
Il sistema proposto è una web application, quindi qualsiasi client con un Web Browser può accedere all'applicazione.

Come Web Server utilizzeremo Apache Tomcat 9.0, dove le pagine JSP/HTML, JS e CSS realizzeranno la UI del sistema e le Java Servlet la logica applicativa. Il client comunica con il Web Server attraverso il protocollo di comunicazione HTTPS. Il server può comunicare con il database, attraverso le API di JDBC. Il Web Server e il DBMS MySql comunicano attraverso le API di JDBC. Inizialmente sia il Web Server che il DBMS saranno istanziati sulla stessa macchina server.



Gestione dei dati persistenti

La gestione dei dati persistenti avverrà attraverso il DBSM MySQL. Attraverso il diagramma delle classi abbiamo individuato le entità che saranno memorizzate nel database. Nel nostro caso tutte le entità saranno rese persistenti in un database.



Scelte progettuali del database:

- Nella traduzione della gerarchia di utenti (Segretario, Responsabile, Genitore) all'interno del database abbiamo inglobato le entità figlie nell'entità padre. Inoltre è stato aggiunto un nuovo attributo (`ruolo`), che discrimina la sotto-entità a cui l'istanza appartiene.

Abbiamo utilizzato questo approccio con il vantaggio di avere un'unica tabella senza vincoli esterni e per ridurre le operazioni di collegamento nel recupero delle informazioni.

Lo svantaggio è che ci potrebbero essere molti valori nulli in corrispondenza degli attributi specifici delle sotto-entità, ma avendo l'entità Segretario e Responsabile con pochi valori, i valori nulli non sono un problema.

- Per quanto riguarda il salvataggio di file (come documentIdentità, certificatoMedico...), abbiamo deciso di salvare i file sul server per evitare di appesantire il database e di avere più efficienze nelle operazioni più frequenti. Nel database, salviamo il riferimento al file come una stringa.

Sicurezza e controllo degli accessi

La comunicazione tra client e server avviene attraverso il protocollo HTTPS, per garantire che non ci siano intercettazioni durante la comunicazione.

Le informazioni sensibili degli utenti saranno trattate in maniera sicura, l'accesso al sistema avviene tramite le credenziali username e password. Per la password sarà usata cifratura basata sulle funzioni hash, che attuano una crittografia detta a "senso unico", permettendo di ottenere una stringa di una determinata lunghezza e "incomprensibile" a partire da una qualsiasi stringa, in modo che l'operazione inversa sia impossibile.

	Utente	Iscrizioni	Bambini	Pagamento
Genitore	Login()	VisualizzaIscrizioni()	VisualizzaBambini()	ScegliModalitàDiPagamento()
	Registrazione()	ModificaIscrizioneBambino() EliminaIscrizioneBambino()	VisualizzaDettagliBambino()	
	Logout()	DisdiciIscrizioneBambino() VisualizzaDettagliIscrizioneBambino() SelezionaIscrizioniDaPagare() VisualizzaDatiPersonali() ModificaDatiPersonali()		
Segretario	Login()	VisualizzaIscrizioniPagate()	VisualizzaDettagliBambinoIscritto()	
	Logout()	VisualizzaDettagliIscrizionePagata() RicercaIscrizioniPagate()		
Responsabile	Login()	RegistraSegretari()		
	Logout()	ControllaAndamentoCentro()		

Controllo del flusso globale

Il controllo globale del software è di tipo Event-Driven Control. Il ciclo principale attende un evento esterno, quando l'evento si verifica viene spedito all'oggetto appropriato. Il controllo risiede in un dispatcher che invoca i servizi di un sottosistema.

Quindi il Web Server gestisce le richieste del client smistandole ai Control (nel nostro caso utilizzeremo Java Servlet), che gestiranno la richiesta interagendo, se necessario, con i Model e infine aggiorneranno le View per il rendering della UI.

Condizioni limite

Fallimento

Nel caso di un fallimento dovuto all'hardware o al software si cercherà di ripristinare il sistema allo stato precedente all'errore.

Nel caso di un fallimento da parte del DBMS, si cercherà di ripristinare lo stato attraverso i meccanismi di backup messi a disposizione del DBMS Mysql.

Nel caso di malfunzionamenti nei dischi rigidi, grazie alla tecnica del RAID 10 avviene l'eventuale swap.

Servizi dei sottosistemi

Interfaccia utente

Gestirà l'interfaccia delle pagine presentate all'Utente attraverso l'uso di HTML, JSP, JS e CSS.

Gestione autenticazione

- **Login:** consente di effettuare l'accesso al sistema.
- **Logout:** consente di effettuare la disconnessione dal sistema.
- **Registrazione:** consente ad un Genitore di registrarsi al sistema.
- **Verifica Account:** consente di verificare un account appena creato da un genitore.

Gestione account

- **Inserisci dati personali:** consente di inserire i dati personali di un Genitore.
- **Modifica dati personali:** consente di modificare i dati personali di un Genitore.
- **Visualizza dati personali:** consente di visualizzare i dati personali di un Genitore.
- **Visualizza bambini:** consente di visualizzare i bambini che sono stati iscritti al sistema.
- **Visualizza dettagli bambino:** consente di visualizzare i dettagli.
- **Cambia password:** consente di cambiare la password ad un utente.
- **Nuova password:** consente di inserire una nuova password ad un Segretario al primo login.
- **Cambia ruolo:** consente ad un Segretario o un Responsabile di cambiare il proprio ruolo.
- **Registra segretario:** consente ad un Responsabile di registrare un nuovo Segretario al sistema.
- **Visualizza andamento centro:** consente al Responsabile di visualizzare l'andamento del centro.

Gestione iscrizioni

- **Seleziona iscrizioni da pagare:** consente di selezionare le iscrizioni valide che un Genitore che vuole pagare.
- **Iscrizione bambino:** consente ad un Genitore di iscrivere un bambino al centro.
- **Modifica iscrizione bambino:** consente ad un Genitore di modificare un'iscrizione non pagata di un bambino.
- **Elimina iscrizione bambino:** consente ad un Genitore di eliminare un'iscrizione non pagata.
- **Visualizza iscrizioni:** consente ad un Genitore di visualizzare tutti le iscrizioni finalizzate.
- **Visualizza dettagli iscrizione:** consente di visualizzare i dettagli di una singola iscrizione.
- **Disdetta iscrizione:** consente ad un Genitore di effettuare la disdetta di un'iscrizione.
- **Approva disdetta:** consente ad un Segretario di approvare una disdetta.
- **Visualizza iscrizioni:** consente ad un Segretario di visualizzare tutti le iscrizioni finalizzate.
- **Ricerca iscrizioni:** consente ad un Segretario di ricercare delle iscrizioni.

- **Visualizza dettagli iscrizione pagata:** consente ad un Segretario di visualizzare i dettagli di un'iscrizione.
- **Controllo iscrizione Qr Code:** consente ad un Segretario di controllare la validità di un'iscrizione tramite il Qr code associato.

Gestione pagamento

- **Modalità di pagamento:** consente ad un Genitore di scegliere la modalità di pagamento.
- **PayPal:** consente ad un Genitore di pagare un'iscrizione tramite PayPal.
- **Pagamento con carta:** consente ad un genitore di pagare un'iscrizione tramite una carta di credito o debito.

Storage

Si occuperà di offrire i servizi per la memorizzazione dei dati persistenti ai sottosistemi sopraindicati.