



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria Software*- Prof. C. Gravino



System Design Document

Riferimento	
Versione	1.4
Data	23/01/2022
Destinatario	Prof. C. Gravino
Presentato da	Sabato Celentano Daniele Donia Gennaro Rascato Pasquale Somma
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
15/11/2021	1.0	Prima stesura	Sabato Celentano Daniele Donia Gennaro Rascato Pasquale Somma
09/12/2021	1.1	Revisione	Sabato Celentano Pasquale Somma
19/12/2021	1.2	Modifica diagramma interfacce	Daniele Donia Gennaro Rascato
13/01/2022	1.3	Modifica definizione e schema deployment diagram	Pasquale Somma Daniele Donia
23/01/2022	1.4	Aggiunta casi d'uso per condizioni limite	Pasquale Somma Daniele Donia Gennaro Rascato Sabato Celentano



Sommario

Revision History.....	2
1. Introduzione	5
1.1 Obiettivo Sistema	5
1.2 Design Goal.....	5
1.2.1 Criteri di performance	5
1.2.2 Criteri di dependability	6
1.2.3 Criteri di costo	6
1.2.4 Criteri di manutenzione	7
1.2.5 Criteri di usabilità.....	7
1.2 Trade-off.....	8
1.3 Acronimi	9
1.4 Riferimenti.....	10
1.5 Organizzazione del Contenuto.....	10
2. Architettura del Sistema Corrente.....	11
3. Architettura del Sistema Proposto	12
3.1 Panoramica.....	12
3.2 Decomposizione in sottosistemi.....	13
3.2.1 Decomposizione in Layer	13
3.2.2 Decomposizione in Sottosistemi.....	15
3.3 Mapping hardware/software	17
3.3.1 Deployment diagram	18
3.3.2 Component diagram	19
3.4 Gestione dati Persistenti	20
3.4.1 Modello concettuale	20
3.4.2 Dizionario dei Dati	21
3.4.3 Tabella dei volumi.....	22
3.4.4 Mapping	22
3.5 Controllo accesso e sicurezza	23
3.5.1 Global access table	23
3.6 Controllo flusso globale del sistema	23
3.7 Condizioni limite	23
3.7.1 Avvio del sistema	24
3.7.2 Spegnimento del sistema	25
3.7.3 Fallimento del server	26



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof. C. Gravino

3.7.4 Fallimento dell'applicazione	27
3.7.5 Errore di accesso ai dati persistenti	28
4. Servizi dei Sottosistemi.....	29
5. Glossario.....	32



1. Introduzione

1.1 Obiettivo Sistema

Il nostro gruppo intende fornire un servizio più intuitivo da utilizzare in alternativa al sito fornito dall'università, <https://www.biblioteche.unisa.it/>, per l'utilizzo e la gestione delle risorse (libri e postazioni di studio) delle biblioteche (scientifica e umanistica). L'obiettivo è rendere il processo di prenotazione più veloce e facilitare la ricerca dei libri e delle postazioni che più si adattano alle esigenze degli studenti e dei dottorandi.

1.2 Design Goal

Il sistema intende soddisfare i seguenti Design Goals:

1.2.1 Criteri di performance

- **Tempo di risposta:** UnisaLIB deve minimizzare il response time in seguito ad un input dell'utente. [generalizzazione dei requisiti non funzionali 8-9]
- **Memoria:** UnisaLIB deve disporre di una quantità di memoria sufficiente a memorizzare l'elenco dei libri e delle postazioni offerti dall'ateneo, i prestiti e il loro storico. È richiesto uno spazio di archiviazione di dimensioni discrete, in particolare per la base di dati in quanto è alla base del modulo di intelligenza artificiale per il consiglio di libri all'utente. A



1.2.2 Criteri di dependability

- **Robustezza:** UnisaLIB deve continuare a funzionare anche in condizioni impreviste, come input non validi immessi dall'utente, prevedendo e gestendo le eccezioni.
- **Affidabilità:** UnisaLIB dovrebbe garantire la persistenza delle informazioni e delle transazioni dell'utente. Il sistema deve eseguire le funzioni richieste con continuità riducendo al minimo i malfunzionamenti. [requisito non funzionale 4 riformulato]
- **Disponibilità:** UnisaLIB dovrebbe essere disponibile tutti i giorni della settimana e a tutte le ore, ad eccezioni di eventuali manutenzioni del server o aggiornamenti dell'applicazione.
- **Tolleranza ai guasti:** UnisaLIB deve tollerare i fallimenti dovuti sia al mancato salvataggio di un prestito o di una prenotazione sia alla mancata cancellazione di un libro o di una postazione.
- **Sicurezza:** UnisaLIB non dovrebbe permettere la visualizzazione in chiaro dei dati sensibili degli utenti. Inoltre, il sistema dovrebbe effettuare controlli per impedire agli utenti di accedere alle funzionalità dell'amministratore e viceversa. [generalizzazione del requisito non funzionale 5]

1.2.3 Criteri di costo

- **Costo di sviluppo:** La stima del costo complessivo per la progettazione e lo sviluppo del sistema è di 50 ore per ogni project member (200 ore totali).
- **Costo di deployment:** UnisaLIB dovrà avere un costo di installazione nullo. Inoltre, non sarà necessario nessun training degli utenti per l'utilizzo dell'applicazione.



1.2.4 Criteri di manutenzione

- **Estensibilità:** UnisaLIB dovrà rendere agevole l'inserimento di nuove funzionalità, scaturite da nuovi bisogni degli utenti o da richieste dell'ateneo.
- **Portabilità:** UnisaLIB dovrebbe essere portabile su tutti i dispositivi che supportano l'ambiente Android [requisito non funzionale 10 riformulato]
- **Tracciabilità:** i requisiti alla base di UnisaLIB potranno essere monitorati attraverso una matrice di tracciabilità, in quanto ogni funzionalità sarà riconducibile ad un insieme di requisiti.

1.2.5 Criteri di usabilità

- **Usabilità:** UnisaLIB dovrebbe essere facile ed intuitivo da utilizzare, adattandosi al dispositivo dell'utente [generalizzazione dei requisiti non funzionali 1-2-3]
- **Utilità:** UnisaLIB semplifica il processo di prenotazione di una postazione e di prestito di un libro rispetto alla piattaforma web preesistente e aggiungendo nuove funzionalità (ad es. la lista dei prestiti dell'utente).



1.2 Trade-off

Costo di sviluppo vs Funzionalità

L'obiettivo di rispettare i costi di sviluppo, ovvero non superare le 200 ore di lavoro totali, e quello di implementare tutte le funzionalità richieste sono in contrasto tra loro. Pertanto, si realizzeranno solo le funzionalità con una priorità maggiore per il cliente, posticipando le altre a versioni successive.

Tempo di risposta vs Robustezza

L'obiettivo di minimizzare il tempo di risposta e quello di garantire la robustezza del sistema sono contrastanti. Si intende favorire l'affidabilità al tempo di risposta, permettendo al sistema di eseguire tutti i controlli necessari sugli input pur avendo un possibile aumento del tempo di risposta.

Usabilità vs Funzionalità

Tra l'intuitività del sistema e l'implementazione di tutte le funzionalità si intende favorire la prima. Sarà quindi data precedenza alle funzionalità che il cliente ritiene essere prioritarie.



1.3 Acronimi

RAD: Requirements Analysis Document.

SDD: System Design Document.

DB: DataBase, ovvero “Base di Dati” .

MySQL: È un RDBMS Open Source basato sul linguaggio SQL, composto da un client a riga di comando e un server.

DBMS: DataBase Management System.

JDBC: Java DataBase Connectivity è un connettore (driver) per database che consente l'accesso e la gestione della persistenza dei dati sulle basi di dati da qualsiasi programma scritto con il linguaggio di programmazione Java

HTTP: HyperText Transfer Protocol, è un protocollo a livello applicativo usato come principale sistema per la trasmissione d'informazioni sul web.

APK: L'estensione APK indica un file Android Package. Questo formato di file, una variante del formato .JAR, è utilizzato per la distribuzione e l'installazione di componenti in dotazione sulla piattaforma per dispositivi mobili Android.

ER: In informatica, nell'ambito della progettazione dei database, il modello entità-relazione è un modello teorico per la rappresentazione concettuale e grafica dei dati a un alto livello di astrazione.

ISBN: International Standard Book Number, è lo standard internazionale del mondo del libro che permette di riconoscere in modo univoco attraverso un codice un libro o un prodotto creato per essere usato come libro.

Url: Uniform Resource Locator, indirizzo di una risorsa espresso in modo univoco e con una forma utilizzabile dal browser;



1.4 Riferimenti

- **Slide del corso**, presenti sulla piattaforma e-learning.
- **Libro di testo**: Object-Oriented Software Engineering (Using UML, Patterns, and Java) Third Edition. Autori: Bernd Bruegge & Allen H. Dutoit.
- **Requirement Analysis Document**: UL_RAD_V1.1
- **Statement of Work**

1.5 Organizzazione del Contenuto

Il documento illustra la struttura dell'applicazione UnisaLIB. Si apre con una presentazione dei design goals che il sistema deve soddisfare, con un compromesso per quelli contrastanti, e una panoramica sugli acronimi e riferimenti utilizzati nel documento per comprenderlo appieno. Viene presentata l'architettura del sistema corrente, nei suoi aspetti principali ed evidenziandone le criticità. In seguito, verrà mostrata l'architettura del sistema proposto, in cui sarà gestita la decomposizione in sottosistemi, il mapping hardware/software, la gestione dei dati persistenti, il controllo degli accessi, il controllo del flusso globale del sistema e le condizioni limite. Verranno spiegati nei dettagli tutti i servizi che offrono i sottosistemi. Il documento si conclude infine con un glossario.



2. Architettura del Sistema Corrente

Al momento l'unico modo per poter prendere in prestito libri o prenotare una postazione è attraverso il sito www.biblioteche.unisa.it, il quale fornisce accesso a queste funzionalità appoggiandosi ad altri due siti: www.eu-campania.hosted.exlibrisgroup.com per la funzione di prestito dei libri e www.easycourse.unisa.it/PortalePlanning per la funzione di prenotazione postazioni.

L'accesso a queste funzioni è molto ostico per gli studenti che si avvicinano inizialmente alla piattaforma, in quanto risulta essere molto dispersiva e farragginosa. L'autenticazione all'interno dei siti sopracitati spesso presenta degli errori che rendono più lento o impossibile l'accesso ai servizi.

Il sito ha un tempo di risposta molto elevato, non mette in evidenza le sue funzioni principali rendendo le procedure prolisse e poco intuitive, inoltre manca di un servizio di profiling utente che permetta di consigliare agli utenti libri in base ai loro prestiti pregressi.

Non avendo accesso alla documentazione ufficiale, dopo un'analisi del sito, supponiamo che esso si basi sul modello MVC. Viene utilizzato un server per la gestione delle richieste finalizzate alla prenotazione delle postazioni. Inoltre, è chiaro l'utilizzo di un database per garantire la persistenza del catalogo dei libri.



3. Architettura del Sistema Proposto

3.1 Panoramica

Il sistema che proponiamo è un'applicazione mobile; una soluzione che, a nostro parere, è ottima per rendere più accessibili i servizi forniti dalle applicazioni web delle biblioteche dell'Università degli Studi di Salerno. L'obiettivo che il sistema si pone è di rendere più semplice e immediata la prenotazione delle postazioni di studio all'interno delle biblioteche di ateneo, di rendere più accessibile la funzione di prestito dei libri e aggiungere funzioni che non sono presenti nell'applicazione web originale.

Il sistema, in particolare, è diviso in:

- Lato client, contenente l'interfaccia utente tramite la quale è possibile richiedere i servizi di prenotazione delle postazioni e di prestito dei libri.
- Lato server, contenente l'accesso ai dati persistenti e i servizi che eseguono le richieste provenienti dal client.

L'architettura che verrà utilizzata nell'implementazione della piattaforma è il modello MVP: un'alternativa per le applicazioni Android all'architettura three-tier, una particolare architettura software e hardware di tipo multi-tier per l'esecuzione di un'applicazione web che prevede la suddivisione dell'applicazione in tre diversi moduli o strati dedicati rispettivamente alla interfaccia utente, alla logica funzionale e alla gestione dei dati persistenti.



3.2 Decomposizione in sottosistemi

3.2.1 Decomposizione in Layer

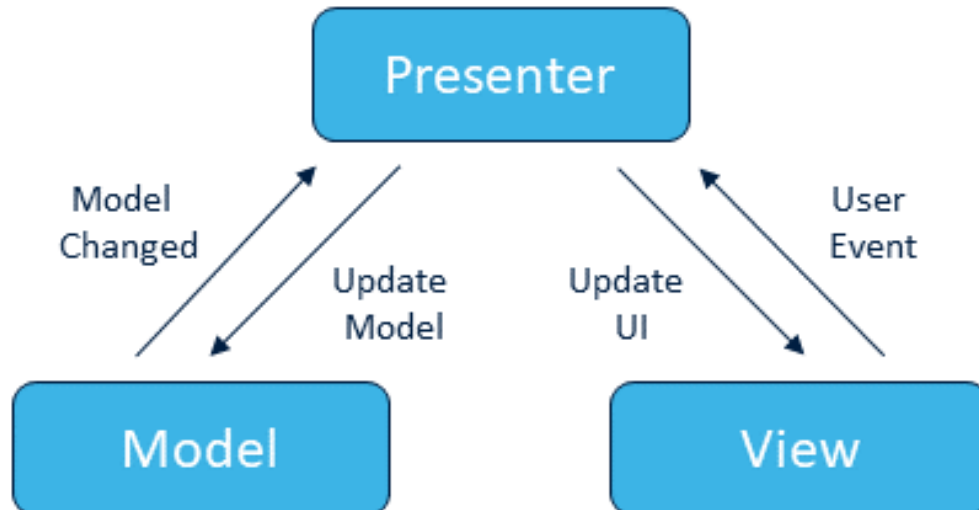
Il sistema segue l'architettura MVP (Model-View-Presenter). La scelta è ricaduta sull'MVP perché permette il riuso del codice e favorisce un approccio “test-driven” in quanto, essendo le componenti isolate, sono più semplici da testare separatamente. Di seguito riportiamo le linee guida alla base dell'architettura MVP:

- Activity, Fragment e CustomView rappresentano la parte View dell'applicazione
- Il Presenter è responsabile dell'ascolto delle interazioni dell'utente, gli aggiornamenti del Model e della View
- Generalmente, View e Presenter sono in relazione uno ad uno
- Una classe Presenter gestisce una sola View alla volta
- Bisogna definire ed implementare delle interfacce per comunicare tra View-Presenter e Presenter-Model
- Il Presenter è responsabile della gestione di tutte le attività di background
- Le classi Model e View non possono avere un riferimento dell'altra

Il sistema è suddiviso in tre livelli:

- **Model:** si occupa della gestione e dello scambio dei dati tra i sottosistemi; fornisce i metodi per accedere ai dati utili all'applicazione.
- **View:** lavora con il Presenter per mostrare i dati, gestisce elementi di interfaccia grafica e si occupa dell'interazione con utenti e agenti.
- **Presenter:** fa da tramite tra il Model e la View. Prende i dati dal Model, li elabora e li restituisce alla View. Gestisce lo stato della View e aggiorna il Model in base agli input dell'utente.

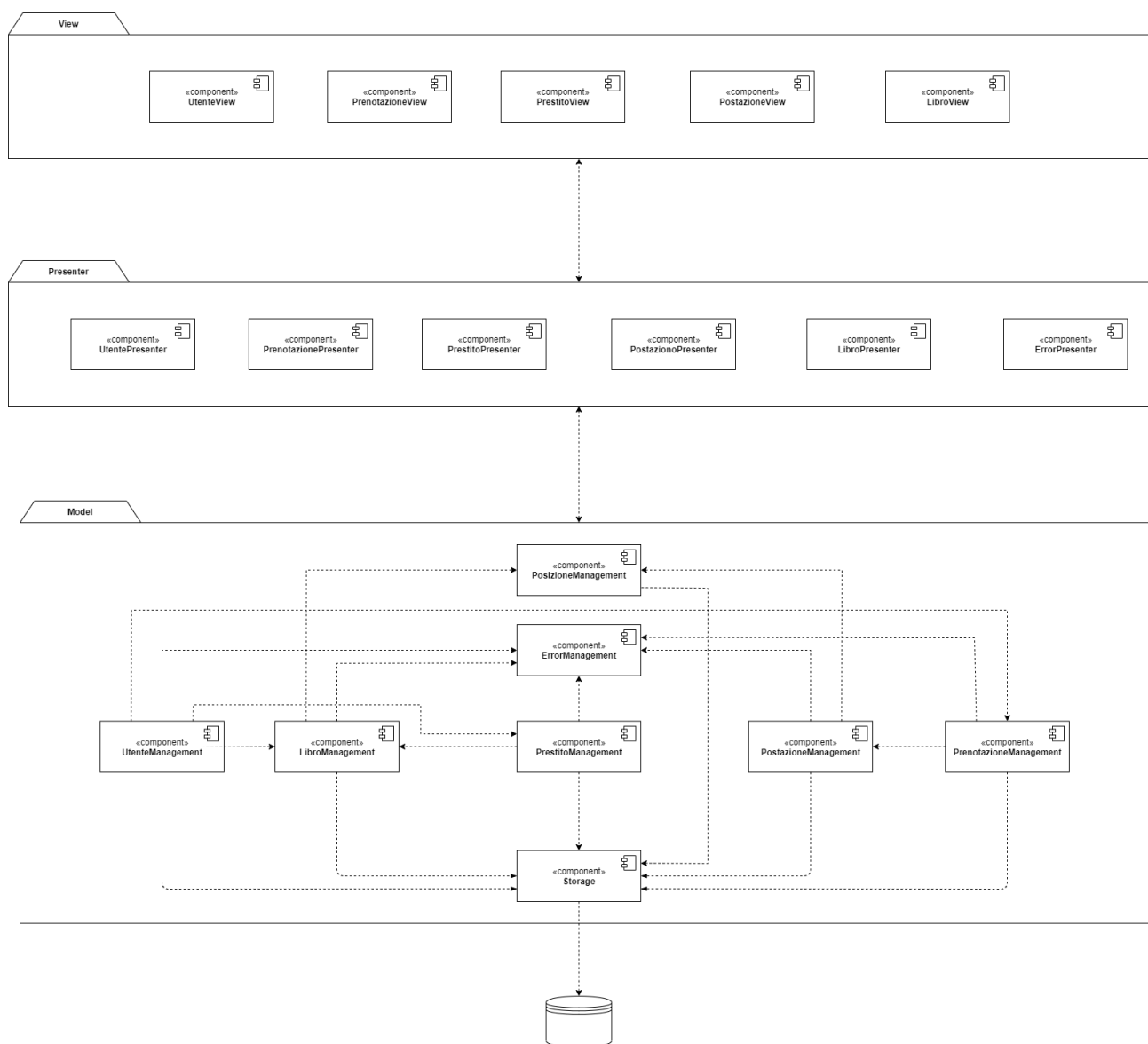
Fonte: <https://www.journaldev.com/14886/android-mvp>.



3.2.2 Decomposizione in Sottosistemi

Dopo un'analisi effettuata sul sistema, abbiamo deciso di suddividerlo nei seguenti sottosistemi in modo tale da utilizzare un'architettura aperta per motivi di efficienza. Si è deciso di gestire i singoli componenti con basso accoppiamento ed elevata coesione in modo tale da garantire, in caso di successive modifiche, il minor numero di aggiornamenti da apportare tra tutti i sottosistemi.

Il sistema si compone di diciotto componenti:



Autori: Daniele Donia, Pasquale Somma



Il livello **View** prevede due sottosistemi:

- **UtenteView**: sottosistema che definisce le interfacce che mettono a disposizione tutte le funzioni dedicate agli utenti e agli admin.
- **PrenotazioneView**: sottosistema che definisce le interfacce per la funzionalità di prenotazione delle postazioni.
- **PrestitoView**: sottosistema che definisce le interfacce per la funzionalità di prestito dei libri.
- **PostazioneView**: sottosistema che definisce le interfacce per le funzionalità di gestione delle postazioni.
- **LibroView**: sottosistema che definisce le interfacce per la funzionalità di gestione dei libri.

Il livello **Presenter** prevede sei sottosistemi:

- **UtentePresenter**: sottosistema che gestisce l'accesso degli utenti, la lista dei prestiti, la lista degli interessi e la lista delle prenotazioni.
- **PrenotazionePresenter**: sottosistema che gestisce le prenotazioni effettuate dagli utenti. Permette agli Utenti di prenotare le postazioni.
- **PrestitoPresenter**: sottosistema che gestisce i prestiti effettuati dagli utenti. Permette agli Utenti di prendere in prestito i libri.
- **PostazionePresenter**: sottosistema che gestisce le postazioni presenti nelle biblioteche di ateneo. Permette l'aggiunta, l'eliminazione e il blocco di una postazione da parte dell'amministratore.
- **LibroPresenter**: sottosistema che gestisce i libri all'interno delle biblioteche di ateneo. Permette la visualizzazione dei dati di un libro agli Utenti e l'aggiunta, eliminazione e modifica dei libri all'amministratore.
- **ErrorPresenter**: sottosistema che gestisce tutti i possibili errori all'interno del sistema.



Il livello **Model** prevede sette sottosistemi:

- **UtenteManagement**: Sistema che si occupa dell'accesso ai dati degli Utenti.
- **PrenotazioneManagement**: Sistema che si occupa dell'accesso ai dati delle prenotazioni.
- **PrestitoManagement**: Sistema che si occupa dell'accesso ai dati dei prestiti.
- **PostazioneManagement**: Sistema che si occupa dell'accesso ai dati delle postazioni.
- **PosizioneManagement**: Sistema che si occupa dell'accesso ai dati delle posizioni.
- **LibroManagement**: Sistema che si occupa dell'accesso ai dati dei libri.
- **ErrorManagement**: Sistema che si occupa della definizione degli errori.
- **Storage**: Sistema che immagazzina i dati in modo permanente.

3.3 Mapping hardware/software

Il sistema proposto utilizzerà una struttura hardware costituita da un server centrale e da un dispositivo Android (smartphone o tablet), compatibile con più versioni di Android OS.

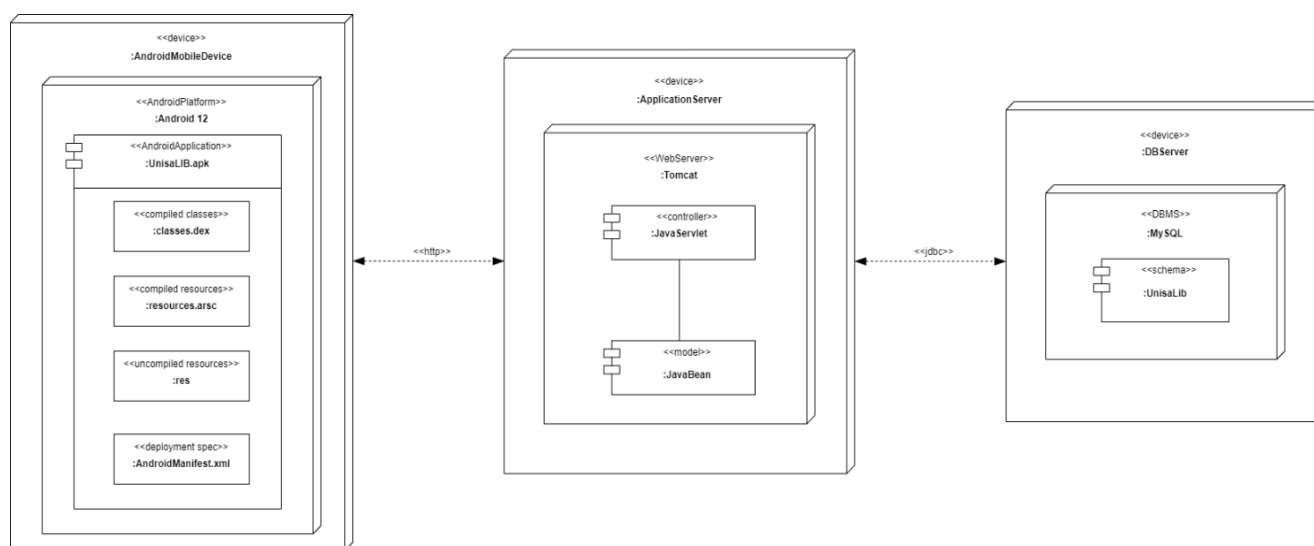
Nell'Application server, in cui risiedono gli oggetti di tipo control ed entity, il client scambia informazioni con lo storage. La comunicazione client-server avverrà tramite protocollo HTTP.

Il server deve essere necessariamente online ed avere una componente software DBMS per trattare grosse quantità di dati. Tramite il server sarà garantita una ripartizione delle varie operazioni consentite ai client a seconda del tipo di autenticazione (utente o admin).

L'autenticazione sarà validata lato server per confermare l'accesso al client.

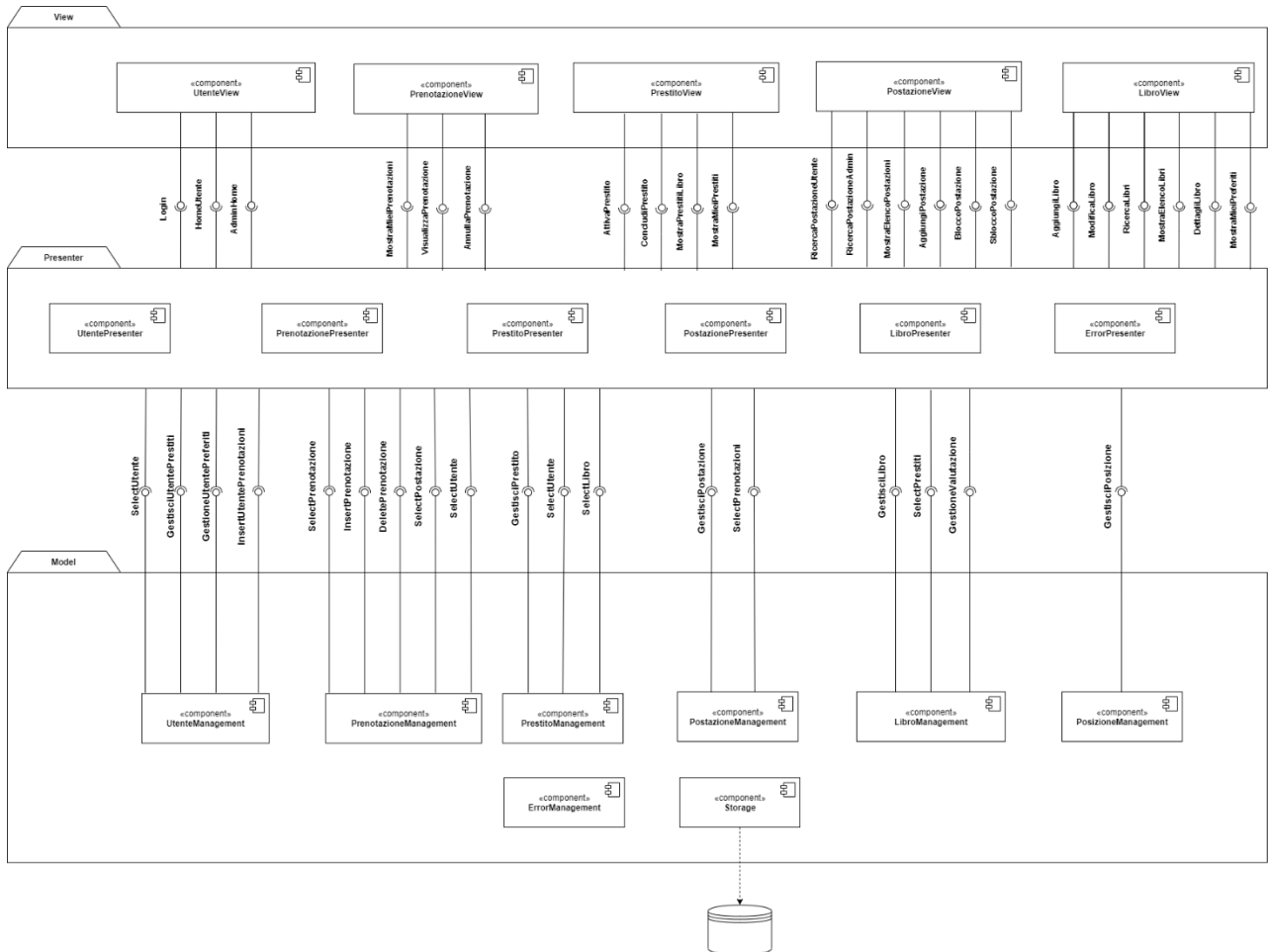
3.3.1 Deployment diagram

Il deployment diagram proposto descrive il sistema in termini di risorse hardware con le relative relazioni. Gli utenti potranno accedere e interagire con UnisaLIB dal proprio client, ossia un dispositivo Android sul quale è installata l'omonima applicazione (UnisaLIB.apk). Il controller poi, una volta validato l'accesso, avrà il ruolo di interagire con il Database tramite JDBC a seconda degli eventi generati dall'utente, richiedendo e prelevando ed eventualmente modificando le opportune risorse.



Autore: Pasquale Somma, Daniele Donia

3.3.2 Component diagram



Autore: Daniele Donia, Gennaro Rascato

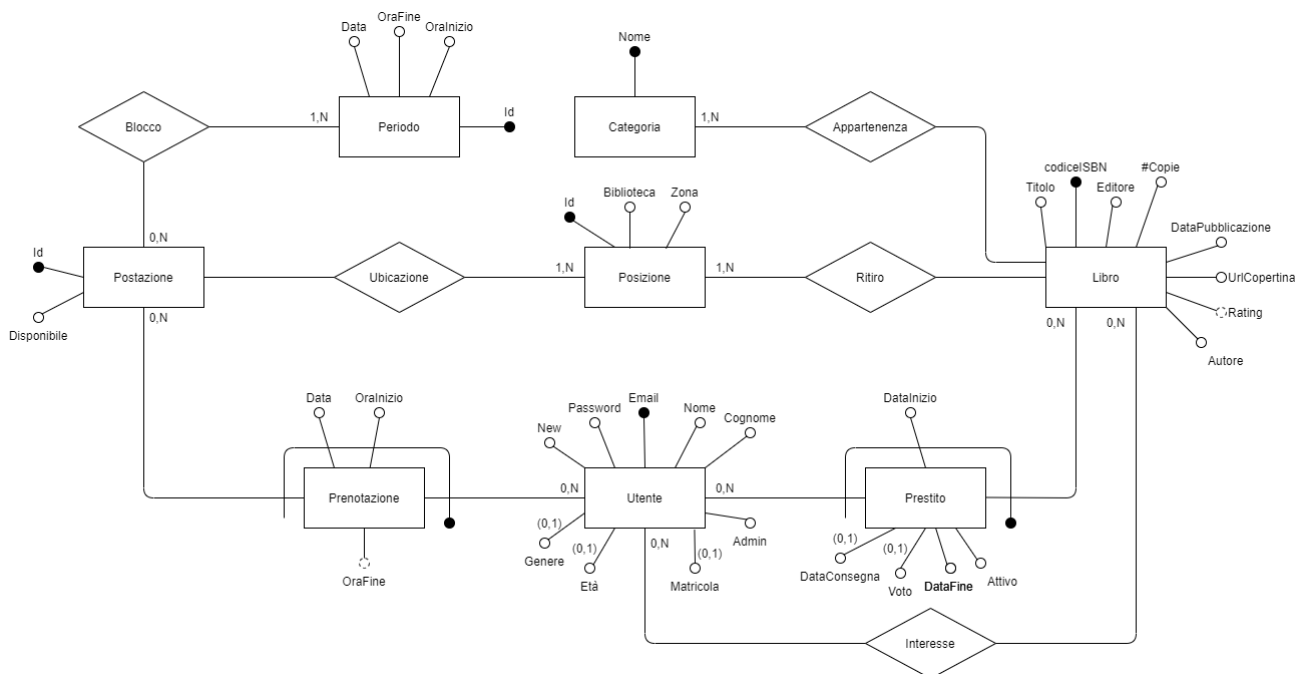
3.4 Gestione dati Persistenti

3.4.1 Modello concettuale

Rispetto al class diagram del RAD si è deciso di unire l'entità UtenteUNISA all'entità Amministratore poiché molto simili negli attributi, di conseguenza abbiamo introdotto il boolean "Admin" per distinguere un UtenteUNISA da un Amministratore.

Inoltre, abbiamo deciso di rendere Categoria un'entità, in modo da facilitare la visualizzazione delle categorie da parte dell'utente, in particolare durante la fase di ricerca libri.

In seguito alle nostre modifiche abbiamo definito il seguente diagramma ER:



Autore: Daniele Donia



3.4.2 Dizionario dei Dati

Entità	Descrizione	Attributi	Identificatore
Utente	Utente in possesso di credenziali istituzionali	Email Password Nome Cognome Admin Matricola New Genere Età	Email
Libro	Libro appartenente al catalogo delle biblioteche di Ateneo	Codice ISBN Titolo Editore #Copie DataPubblicazione UrlCopertina Rating Autore	Codice ISBN
Prestito	Prestito di un libro preso da un Utente	DataInizio DataFine DataConsegna Voto Attivo	DataInizio Email
Postazione	Postazione di studio all'interno della biblioteca d'Ateneo	Id Disponibile	Id
Periodo	Periodo temporale	Id Data OralInizio OraFine	Id
Posizione	Luogo all'interno di una delle due biblioteche di ateneo	Id Biblioteca Zona	Id
Categoria	Classe contraddistinta da un particolare criterio	Nome	Nome

Relazione	Descrizione	Componenti	Attributi
Interesse	Interesse di un utente ad un libro	Utente, Libro	N/A
Appartenenza	Appartenenza di un libro ad una categoria	Libro, Categoria	N/A



Ritiro	Ritiro di un libro in una posizione	Libro, Posizione	N/A
Ubicazione	Ubicazione di una postazione	Postazione, Posizione	N/A

Autore: Daniele Donia

3.4.3 Tabella dei volumi

Nome	Costrutto	Volumi
Utente	E	35100
Categoria	E	100
Posizione	E	100
Libro	E	200000
Periodo	E	35
Postazione	E	200
Prestito	E	1000000
Prenotazione	E	7000
Interesse	R	1000000
Ritiro	R	1000000
Appartenenza	R	200000
Ubicazione	R	200
Blocco	R	7000

Autore: Pasquale Somma

3.4.4 Mapping

Utente (Email, Password, Nome, Cognome, Admin, New, Matricola*, Genere*, Eta*)

Libro (CodiceISBN, Titolo, Editore, NCopie, DataPubblicazione, UrlCopertina, Rating, Autore, *Categoria*, *Posizione*)

Prestito (DataInizio, Libro, Utente, DataFine, DataConsegna*, Voto*, Attivo)

Postazione (Id, Disponibile, *Posizione*)

Periodo (Id, Data, OraInizio, OraFine)

Posizione (Id, Biblioteca, Zona)

Categoria (Nome)

Prenotazione (Data, OraInizio, Postazione, Utente, OraFine)

Interesse (Utente, Libro)

Blocco (Postazione, Periodo)

Autore: Sabato Celentano



3.5 Controllo accesso e sicurezza

3.5.1 Global access table

Autore: Gennaro Rascato, Sabato Celentano

Oggetto Attore	Postazione	Libro	Prenotazione	Prestito	Blocco
Utente	getInfoPostazione()	isDisponibile() getInfoLibro() addMieiInteressi() removeMieiInteressi()	createPrenotazione() getPrenotazione() annullaPrenotazione()	createPrestito() getInfoPrestito() annullaPrestito()	
Admin	blocco() sblocco() addPostazione() getElencoPostazioni()	updateInfoLibro() getInfoLibro() getListaPrestiti() addLibro() deleteLibro()	getPrenotazione()	setDataConsegna() setRitirato()	create() destroy()

3.6 Controllo flusso globale del sistema

Le funzionalità fornite dalla piattaforma UnisaLIB richiedono una continua interazione da parte dell'utente, pertanto, il controllo del flusso globale del sistema è “event-driven”, guidato dagli eventi.

3.7 Condizioni limite

In questa sezione sono analizzate le condizioni limite del sistema. Nello specifico sono mostrati i casi d'uso per l'avvio, lo spegnimento e il fallimento del sistema e per l'errore di accesso ai dati persistenti.



3.7.1 Avvio del sistema

Identificativo <i>UC_CL1</i>		<i>Avviare il sistema</i>	<i>Data</i>	<i>23/01/2022</i>
			<i>Vers.</i>	<i>0.00.001</i>
			<i>Autore</i>	<i>Donia Daniele</i>
Descrizione		<i>Lo UC definisce le operazioni necessarie per avviare il sistema</i>		
Attore Principale		Amministratore È interessato a mettere in funzione il sistema affinché sia fruibile agli utenti		
Attori secondari		NA		
Entry Condition		L'amministratore accende il database server		
Exit condition On success		L'amministratore ha avviato il sistema correttamente		
Exit condition On failure		L'amministratore non avvia il sistema		
Extension point		NA		
Generalization of		NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Amministratore:	Esegue il comando per avviare l'application server		
2	Sistema:	Instaura una connessione JDBC con il database server		
3	Sistema:	Controlla che dati persistenti non siano danneggiati		
4	Sistema:	Rende fruibili i servizi agli utenti		
Scenario/Flusso di eventi di ERRORE: la connessione JDBC non va a buon fine				
2.1	Sistema:	Visualizza un messaggio di avviso all'amministratore per comunicare che la connessione non è riuscita		
2.2	Amministratore:	Riavvia il server		
Scenario/Flusso di eventi di ERRORE: i dati persistenti sono danneggiati o corrotti				



3.1	Sistema:	Visualizza un messaggio di avviso all'amministratore per comunicare che i dati sono danneggiati
3.2	Amministratore:	Restaura la buona qualità dei dati
3.3	Amministratore:	Riavvia il server

3.7.2 Spegnimento del sistema

Identificativo <i>UC_CL2</i>		<i>Spegnere il sistema</i>	<i>Data</i>	<i>23/01/2022</i>
			<i>Vers.</i>	<i>0.00.001</i>
			<i>Autore</i>	<i>Donia Daniele</i>
Descrizione	<i>Lo UC definisce le operazioni necessarie per lo spegnimento del sistema</i>			
Attore Principale	Amministratore È interessato a terminare l'attività del server			
Attori secondari	NA			
Entry Condition	L'amministratore accede al server			
Exit condition On success	L'amministratore ha spento il sistema correttamente			
Exit condition On failure	L'amministratore non ha spento il sistema			
Extension point	NA			
Generalization of	NA			
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Amministratore:	Esegue il comando per spegnere l'application server		
2	Sistema:	Chiude la connessione JDBC con il database		
3	Sistema:	Termina l'esecuzione dell'application server e visualizza un messaggio dello spegnimento del server		
4	Amministratore:	Esegue il comando per terminare l'attività del database server		
5	Sistema:	Termina l'esecuzione del database server		



3.7.3 Fallimento del server

I fallimenti del sistema possono essere causati da malfunzionamenti hardware o errori software. Lato server, può verificarsi un'interruzione dell'alimentazione, a cui si può rimediare con un generatore di emergenza così da poter effettuare un backup dei dati e attendere il ripristino dell'alimentazione. Può verificarsi anche un errore di connessione tra l'application server e il database server. Dal momento che possono verificarsi guasti alla memoria del database, si effettuano backup periodici dei dati, in modo da poterli recuperare in seguito.

Identificativo <i>UC_CL3</i>	<i>Gestire il fallimento del sistema</i>	<i>Data</i>	<i>23/01/2022</i>
		<i>Vers.</i>	<i>0.00.001</i>
		<i>Autore</i>	<i>Somma Pasquale</i>
Descrizione	<i>Lo UC definisce le operazioni necessarie per la gestione del fallimento del server</i>		
Attore Principale	Amministratore È interessato a gestire il fallimento del server in caso di malfunzionamenti hardware o errori software		
Attori secondari	NA		
Entry Condition	Il sistema riscontra un fallimento		
Exit condition On success	L'amministratore ha riavviato correttamente il sistema		
Exit condition On failure	L'amministratore non ha gestito il fallimento		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Amministratore:	Ricerca la causa del fallimento e individua una soluzione	
2	Amministratore:	Riavvia il sistema utilizzando il caso d'uso UC_CL1	
4	Sistema:	Comunica che il riavvio è avvenuto correttamente	



3.7.4 Fallimento dell'applicazione

I fallimenti dell'applicazione possono essere causati da bug nel codice.

Identificativo <i>UC_CLA</i>	<i>Gestire l'errore di accesso ai dati</i>	<i>Data</i>	<i>23/01/2022</i>
		<i>Vers.</i>	<i>0.00.001</i>
		<i>Autore</i>	<i>Celentano Sabato</i>
Descrizione	<i>Lo UC definisce le operazioni necessarie per la gestione del fallimento dell'applicazione</i>		
Attore Principale	Utente È interessato ad usufruire dei servizi dell'applicazione		
Attori secondari	NA		
Entry Condition	Il sistema riscontra un fallimento		
Exit condition On success	L'utente riesce ad accedere nuovamente all'applicazione		
Exit condition On failure	L'utente non riesce ad accedere all'applicazione		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Sistema:	Effettua una chiusura forzata dell'applicazione	
2	Sistema:	Notifica all'utente la presenza di un errore	
3	Utente:	Segnala l'errore all'amministratore del sistema	



3.7.5 Errore di accesso ai dati persistenti

Identificativo <i>UC_CL5</i>	<i>Gestire l'errore di accesso ai dati</i>	<i>Data</i>	<i>23/01/2022</i>
		<i>Vers.</i>	<i>0.00.001</i>
		<i>Autore</i>	<i>Rascato Gennaro</i>
Descrizione	<i>Lo UC definisce le operazioni necessarie per la gestione dell'errore di accesso ai dati persistenti</i>		
Attore Principale	Amministratore È interessato a gestire l'errore di accesso ai dati persistenti		
Attori secondari	NA		
Entry Condition	Il sistema riscontra un errore nell'accedere ai dati		
Exit condition On success	Il sistema riesce ad accedere ai dati persistenti		
Exit condition On failure	Il sistema non riesce ad accedere ai dati persistenti		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Sistema:	Comunica tramite una notifica l'impossibilità di accedere ai dati persistenti	
2	Amministratore:	Verifica la qualità dei dati e risana eventuali problemi	
3	Amministratore	Termina il sistema utilizzando il caso d'uso UC_CL2	
4	Amministratore:	Riavvia il sistema utilizzando il caso d'uso UC_CL1	
5	Sistema:	Comunica che il riavvio è avvenuto correttamente	



4. Servizi dei Sottosistemi

Le **View** rappresentano le interfacce che gestiscono l'interfaccia grafica e gli eventi generati dall'interazione dell'utente con il sistema

InterfacciaUtenteUNISA fornisce dieci servizi all'interfaccia Presenter:

- *Login*: mostra l'interfaccia tramite cui l'utente effettua l'accesso all'applicazione.
- *UtenteHome*: mostra le funzioni di prestito libro e prenotazione postazione, inoltre fornisce l'accesso ad un menù che permette la visualizzazione dei propri dati e del tasto di logout.
- *MostraMieiPrestiti*: mostra i prestiti dell'utente.
- *MostraMieiPreferiti*: mostra i libri nei preferiti dell'utente.
- *MostraMiePrenotazioni*: mostra le prenotazioni effettuate dall'utente.
- *RicercaLibri*: mostra la barra di ricerca dei libri e le categorie dei libri.
- *MostraElencoLibri*: mostra l'elenco dei libri risultati dalla ricerca.
- *DettagliLibro*: mostra i dettagli di un libro.
- *RicercaPostazioneUtente*: mostra il form per la ricerca di una postazione in biblioteca.
- *MostraElencoPostazioni*: mostra la griglia delle postazioni disponibili.



InterfacciaAdmin fornisce undici servizi all'interfaccia Presenter:

- *Login*: mostra l'interfaccia tramite cui l'admin effettua l'accesso all'applicazione.
- *AdminHome*: mostra le funzionalità di gestione libri e gestione postazioni, inoltre fornisce l'accesso ad un menù che permette la visualizzazione dei propri dati e del tasto di logout.
- *RicercaLibri*: mostra la barra di ricerca dei libri e le categorie dei libri.
- *MostraElencoLibri*: mostra l'elenco dei libri risultanti dalla ricerca.
- *AggiungiLibro*: mostra il form per l'aggiunta dei libri.
- *ModificaLibro*: mostra il form di modifica dei libri.
- *RicercaPostazioniAdmin*: mostra il form per la ricerca di una postazione da parte dell'admin.
- *MostraElencoPostazioni*: mostra un elenco con le postazioni risultanti dalle ricerche dell'admin.
- *AggiungiPostazione*: mostra il form per l'aggiunta delle postazioni all'admin.
- *BloccoPostazione*: mostra l'interfaccia per inserire i dati per il blocco delle postazioni.
- *SbloccoPostazione*: mostra l'elenco delle postazioni bloccate e l'opzione per sbloccarle.

I **Model** rappresentano la logica applicativa, quindi tutte le operazioni che si possono effettivamente eseguire sui dati a disposizione del sistema.

UtenteManagement fornisce quattro servizi all'interfaccia Presenter:

- *SelectUtente*: permette l'accesso ai dati persistenti relativi agli utenti.
- *GestioneUtentePrestiti*: permette l'accesso ai dati, l'inserimento e l'eliminazione dei prestiti dell'utente.
- *GestioneUtentePreferiti*: permette l'accesso ai dati, l'inserimento e l'eliminazione dei libri preferiti dall'utente.
- *GestioneUtentePrenotazione*: permette l'accesso alla lista delle prenotazioni dell'utente.



PrenotazioniManagement fornisce cinque servizi all'interfaccia Presenter:

- *SelectPrenotazione*: permette l'accesso ai dati persistenti relativi alle prenotazioni.
- *InsertPrenotazione*: permette l'inserimento di nuove prenotazioni.
- *DeletePrenotazione*: permette l'eliminazione di una prenotazione.
- *SelectPostazione*: permette l'accesso ai dati di una postazione relativa ad una prenotazione.
- *SelectUtente*: permette l'accesso ai dati di un utente relativo ad una prenotazione.

PrestitiManagement fornisce tre servizi all'interfaccia Presenter:

- *GestionePrestito*: permette l'accesso ai dati, l'inserimento e l'eliminazione di un prestito.
- *SelectUtente*: permette l'accesso ai dati di un utente relativo ad un prestito.
- *SelectLibro*: permette l'accesso ai dati di un libro relativo ad un prestito.

PostazioniManagement fornisce due servizi all'interfaccia Presenter:

- *GestionePostazione*: permette l'accesso ai dati, l'inserimento e l'eliminazione di una prenotazione.
- *SelectPrenotazioni*: permette l'accesso ai dati delle prenotazioni relative ad una postazione.

LibriManagement fornisce tre servizi all'interfaccia Presenter:

- *GestioneLibro*: permette l'accesso ai dati, l'inserimento e l'eliminazione di un libro.
- *SelectPrestiti*: permette l'accesso ai dati dei prestiti relativi ad un libro.
- *GestioneValutazione*: permette l'inserimento e l'accesso ai dati riguardanti le valutazioni.



5. Glossario

- Admin:** termine inglese per indicare amministratore;
- Amministratore:** funzionario addetto alla gestione dei servizi delle biblioteche dell'Università degli Studi di Salerno;
- Application Server:** una tipologia di server che fornisce l'infrastruttura e le funzionalità logiche di supporto, sviluppo ed esecuzione di applicazioni;
- Blocco:** indisposizione di una postazione di studio;
- Component Diagram:** diagramma che ha lo scopo di rappresentare la struttura interna del sistema software modellato in termini dei suoi componenti principali e delle relazioni fra di essi;
- Deployment Diagram:** diagramma di distribuzione che permette la rappresentazione dell'architettura fisica del sistema (hardware e software) a diversi livelli di dettaglio;
- Form:** parte dell'interfaccia utente di un'applicazione che consente all'utente di inserire e inviare all'application server uno o più dati digitati dallo stesso;
- Mobile Application:** applicazione software dedicata ai dispositivi di tipo mobile, quali smartphone o tablet, tipicamente progettata e realizzata in maniera più leggera in termini di risorse hardware;
- Layer:** un gruppo di sottosistemi che provvede a realizzare servizi possibilmente utilizzando servizi forniti da altri layer;
- Login:** procedura attraverso la quale un utente registrato può accedere alla piattaforma. Richiede l'inserimento della e-mail universitaria e di una password;
- Logout:** procedura attraverso la quale un utente registrato può disconnettersi dalla piattaforma;
- Posizione:** collocazione di una postazione di studio o del punto di ritiro di un libro;
- Posto/Postazione:** luogo della biblioteca attrezzato per lo studio e prenotabile dagli studenti;
- Prenotazione:** impegno a occupare e diritto ad aver riservato una postazione di studio della biblioteca;
- Prestito:** contratto con cui l'utente riceve dalla biblioteca libro impegnandosi a restituirlo alla data prestabilita;



- Requisito funzionale:** requisito che definisce una funzione di un sistema di uno o più dei suoi componenti;
- Requisito non funzionale:** requisito che descrive gli aspetti del sistema che non sono direttamente legati alle funzionalità del sistema;
- Responsive:** tecnica di web design per la realizzazione di siti in grado di adattarsi graficamente in modo automatico al dispositivo coi quali vengono visualizzati, riducendo al minimo la necessità dell'utente di ridimensionare e scorrere i contenuti;
- Utente:** fa riferimento ad UtenteUnisa ed Amministratore, che in certe situazioni effettuano le stesse operazioni;
- Web Application:** applicazione accessibile e fruibile dall'utente tramite un browser web con una connessione attiva;
- Zona:** sezione della biblioteca.