**Università degli Studi di Salerno**

**Corso di Ingegneria del Software**

**BookPoint**

**SDD**

**Versione 1.0**

Partecipanti:

|  |  |
| --- | --- |
| **Nome Cognome** | **Matricola** |
| **Massimo Giordano** | **0512104480** |
| **Giovanni Buonocore** | **0512104612** |
| **Gennaro Teodoro** | **0512104876** |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autore** |
|  |  |  |  |

Sommario

[1 Introduzione 2](#_Toc533693297)

[1.1 Purpose of the system 2](#_Toc533693298)

[1.2 Design Goals 2](#_Toc533693299)

[1.3 Definition, acronyms, and abbreviations 2](#_Toc533693300)

[1.4 References 2](#_Toc533693301)

[1.5 Overview 2](#_Toc533693302)

[2 Current software architecture 3](#_Toc533693303)

[3 Proposed software architecture 3](#_Toc533693304)

[3.1 Overview 3](#_Toc533693305)

[3.2 Subsystem decomposition 3](#_Toc533693306)

[3.3 Hardware/software mapping 3](#_Toc533693307)

[3.4 Persistent data management 3](#_Toc533693308)

[3.5 Access control and security 3](#_Toc533693309)

[3.6 Global software control 3](#_Toc533693310)

[3.7 Boundary conditions 3](#_Toc533693311)

[4 Subsystem services 3](#_Toc533693312)

# Introduzione

## Purpose of the system

## Design Goals

**Tempi di risposta:** Il sistema dovrà essere performante e reattivo in ogni situazione e garantire tempi di risposta ragionevoli per l'utente, in modo che egli non percepisca il sistema come bloccato e non provi frustrazione durante il suo utilizzo.

**Validazione input:** il sistema deve essere robusto, ovvero ogni volta che l’utente compila degli input di testo, ad esempio il login/registrazione, l’input sarà controllato in modo da non permettere la sottomissione di input non ammesso.

**Sicurezza:** quando l’utente si registra o cambia la password, quest’ultima viene criptata e salvata all’interno del database, così da non permettere, in nessun modo di risalire alla password se non attraverso il codice di decifratura.

**Criteri di costi (Tempo):** il sistema deve essere consegnato entro il secondo appello di is.

**Criteri di mantenimento:** Il codice deve essere comprensibile e facile da modificare in caso di futuri cambiamenti. Per far questo si userà Javadoc per commentare i metodi, il che aiuterà l’individuazione del metodo da modificare.

**Usabilità:** Il sito deve essere semplice dal punto di vista dell’usabilità. Con semplice si intende che il sito deve guidare l’utente in tutte le azioni che fa, aiutandolo attraverso un’interfaccia semplice con immagini, testi e bottoni, in modo da minimizzare il numero di errori che un utente può commettere.

## Definition, acronyms, and abbreviations

## References

Vedere RAD

## Overview

Il seguente documento di System Design (SDD) mostra i dettagli tecnici del sistema BookPoint.

Il documento inizia con un’ introduzione generale del sistema seguito dagli obiettivi di design che il sistema si propone di raggiungere. Viene discussa la suddivisione del sistema in sottosistemi e definito il mapping hardware/software in modo da assegnare i sottosistemi ad uno specifico hardware.  
Verranno descritti il controllo degli accessi e i problemi di sicurezza legati al sistema.

Infine, saranno discussi il controllo generale del software con i relativi problemi di concorrenza e le boundary conditions trattando gli scenari iniziali e la gestione del sistema.

# Current software architecture

# Proposed software architecture

## Overview

In seguito verrà mostrata la suddivisione del sistema in sottosistemi con i relativi compiti che ogni sottosistema dovrà offrire e il mapping hardware/software del sistema.

Successivamente vengono descritti i dati persistenti con il relativo schema relazionale, verrà discusso il controllo degli accessi con la descrizione degli utenti del nostro sistema, il controllo generale del software e le boundary contitions.

## Subsystem decomposition

Interfaccia Utente: Fornisce all’utente l’interfaccia per interagire col sistema.

GestioneAccount: Fornisce i servizi per gestire il proprio account, quali: gestione delle carte di credito, gestione degli indirizzi e cambio password.

GestioneLibro: Fornisce i servizi per ricercare, modificare e aggiungere dei libri.

GestioneCarrello: Gestisce il carrello, permettendone la modifica e la persistenza.

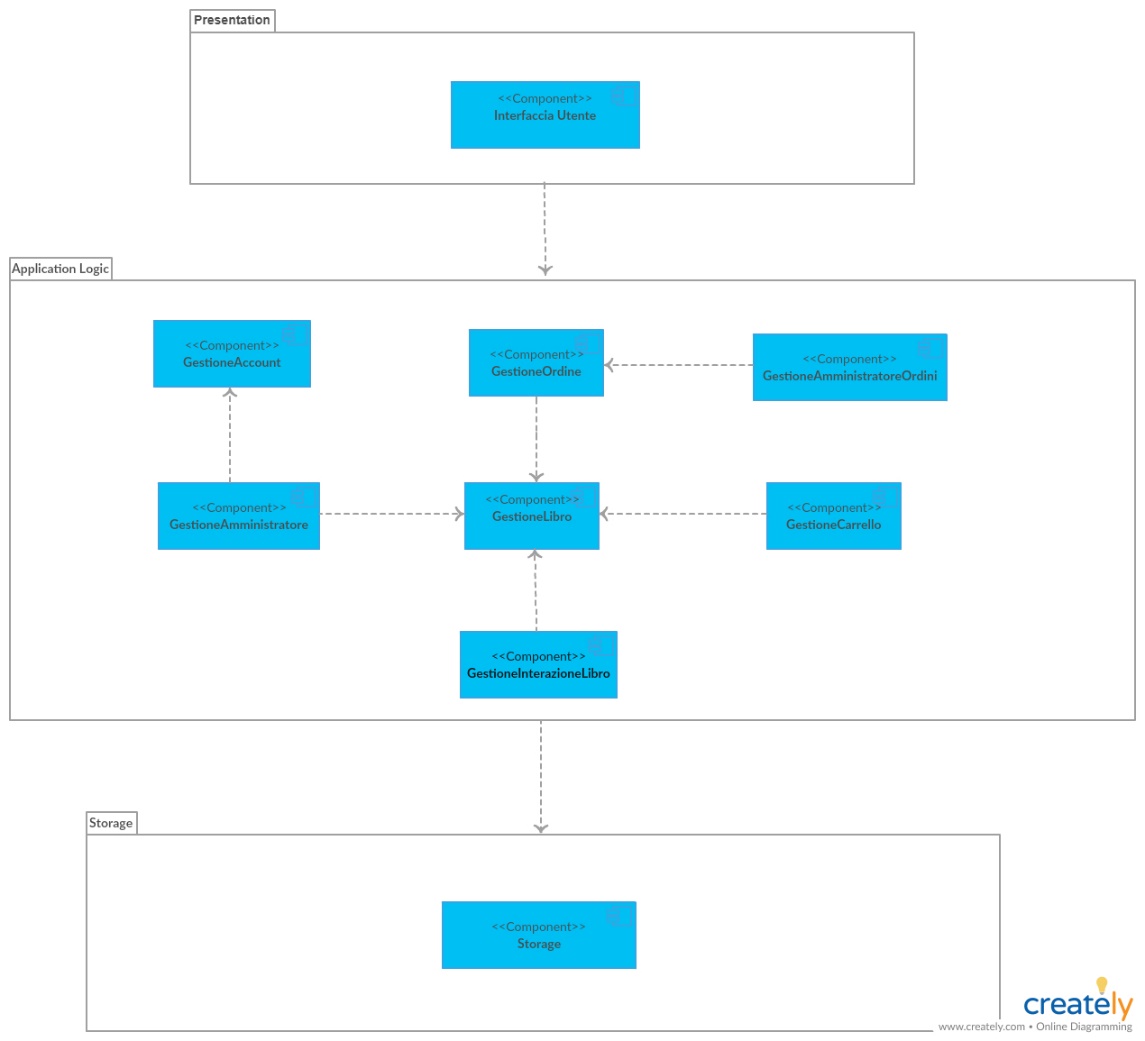
GestioneOrdine: Permette di consultare lo storico degli ordini, visualizzare la fattura, ricercare un ordine e di modificarne lo stato e la data di consegna.

GestioneAmministratore: Fornisce all’amministratore gli strumenti necessari per aggiungere, modificare e rimuovere un libro, per ricercare un account e modificarne il tipo.

GestioneAmministratoreOrdini: Fornisce all’amministratore gli strumenti necessari per ricercare un ordine e modificarne lo stato e la data di consegna.

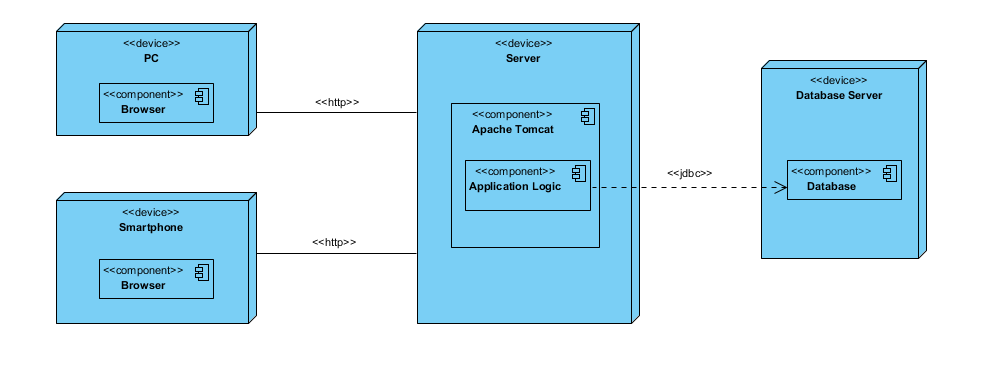
GestioneInterazioneLibro: Fornisce i servizi per aggiungere o rimuovere una recensione e per aggiungere o rimuovere un libro dai preferiti.

Storage: gestisce il salvataggio, la modifica e il recupero dei dati dal database.



## Hardware/software mapping

BookPoint utilizza un’architettura Client/Server.   
Il WebServer è rappresentato da Apache Tomcat 9 ed è situato su una singola macchina, la logica del sistema è costituita da Java Servlet mentre l’interfaccia utente è realizzata utilizzando pagine JSP (Java Servlet Page). Il Client è rappresentato dal Web Browser utilizzato dall’utente.  
La comunicazione tra i nodi è rappresentata da richieste e risposte http tra client e server, e da query in JDBC tra server e database.



## Persistent data management

## Access control and security

## Global software control

## Boundary conditions

# Subsystem services