

Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022



# UNIVERSITÀ DEGLI STUDI DI SALERNO

Corso di Laurea in Informatica, a.a. 2022-23  
 Progetto del corso di Ingegneria del Software  
 prof. A. De Lucia, prof. M. De Stefano  
 Repository GitHub: <https://github.com/giocolella/is-bookbearer-22-23>



- *System Design Document* |  
*Versione 1.0*

Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022

#### **Coordinatore del progetto:**

Nome	Matricola
De Lucia Andrea	
De Stefano Manuel	
Colella Giorgio	0512105946

#### **Partecipanti:**

Nome	Matricola
Colella Giorgio	0512105946

<b>Scritto da:</b>	Colella Giorgio
--------------------	-----------------

## **Revision History**

Data	Versione	Descrizione	Autore

Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022

## Indice

1. INTRODUZIONE.....	4
1.1. Scopo del sistema .....	4
1.2. Ambito del sistema.....	4
1.3. Obiettivi del progetto .....	5
1.4. Riferimenti .....	5
1.5. Definizioni, acronimi ed abbreviazioni .....	5
1.6. Panoramica .....	6
2. Sistema corrente.....	7
3. Sistema proposto .....	7
3.1. Panoramica .....	8
3.2. Decomposizione in sottosistemi.....	9
3.3. Mappatura Hardware/Software .....	12
3.4. Dati persistenti.....	12
3.5. Controllo accessi e sicurezza.....	13
3.6. Controllo globale del software .....	13
3.7. Boundary conditions .....	13
4. Servizi dei sottosistemi .....	14
5. Glossario .....	17

Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022

# 1.Introduzione

## 1.1 Scopo del sistema

Il mondo dell'intrattenimento è diventato estremamente proficuo in relazione a diversi medium come il cinema, videogiochi e libri. Tuttavia a causa di questa grande produzione di proprietà intellettuali non è facile districarsi all'interno di tutte le diverse produzioni odierne e del passato. Ogni consumatore vorrebbe poter spendere bene il proprio tempo libero potendo usufruire sempre di ciò che sta cercando.

Concentrandoci sul mondo delle produzioni letterarie, si sente il **bisogno di capire con facilità se il prodotto che si ha davanti è proprio quello che il consumatore sta cercando e che sia di buona od ottima qualità**. Si vuole inoltre poter avere una lista sempre a portata di mano di titoli che si vorranno leggere in futuro essendo magari già impegnati con altre produzioni.

## 1.2 Ambito del sistema

**Book Bearer è un'applicazione Android** che prende ispirazione dal termine "Ring Bearer" del Signore degli Anelli di Tolkien. La sua funzione è quella di permettere ai lettori più appassionati di scegliere nuovi libri da leggere in base a recensioni della critica professionista o di altri consumatori in modo da poter avere un quadro completo della qualità del prodotto. Esiste inoltre la possibilità di aggiungere recensioni proprie grazie all'iscrizione alla piattaforma e di imbastire un proprio profilo con tanto di "libri già letti" e "libri da leggere in futuro".

Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022

## 1.3. Obiettivi del progetto

### **Obiettivi:**

- Gestione Registrazione;
- Gestione Autenticazione;
- Gestione Area Utente;
- Gestione Liste;
- Gestione Recensioni;
- Gestione Catalogo;
- Interfaccia intuitiva e facile da usare;

### **Criteri di successo:**

- Gestione Registrazione;
- Gestione Autenticazione;
- Gestione Area Utente;
- Gestione Liste;
- Gestione Recensioni;
- Gestione Catalogo.

## 1.4 Riferimenti

- In questo documento si fa riferimento al RAD di Book Bearer.

## 1.5 Definizioni, acronimi ed abbreviazioni

Login = processo che permette l'autenticazione di un utente.

Logout = processo che permette la de autenticazione di un utente.

Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022

RAD = Requirement Analysis Document.

UC = Use Case.

UCD = Use Case Diagram.

RF = Requisito Funzionale.

RNF = Requisito Non Funzionale.

PSR = Pseudo Requisito.

CD = Class Diagram.

SD = Sequence Diagram.

SCD = Statechart Diagram.

MP = Mockup

SDD = System Design Document

## 1.6 Panoramica

Book Bearer è un'applicazione Android che consentirà:

- ~ Registrazione di un account, login e cancellazione dell'account;
- ~ Ricerca di libri;
- ~ Possibilità di aggiungere i libri ad una lista di libri già letti o ad una lista di libri letti;
- ~ Possibilità di avere un profilo personale;
- ~ Possibilità di leggere ed aggiungere recensioni a libri;
- ~ Possibilità di visualizzare informazioni dei diversi libri nel catalogo;
- ~ Possibilità di gestione del catalogo.

Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022

## 2. Sistema Corrente

Senza lo sfruttamento delle tecnologie odierne un lettore entrerebbe in una libreria e nel caso fosse interessato ad uno o più libri senza però poterli acquistare in quel preciso momento per diversi motivi (budget momentaneamente mancante, già impegnato in diverse letture, ecc..) dovrebbe prendere carta e penna (che potrebbe anche non avere) e segnare tali libri con la possibilità di perdere la nota cartacea. Nel caso poi in cui decidesse di acquistare un libro il suo sarebbe un acquisto “al buio”, senza cioè sapere nulla sulla qualità del libro non potendo consultare recensioni.

Inoltre, nel caso in cui ci fossero dei metodi per sopperire a queste due mancanze, tali metodi sarebbero separati e non connessi con la possibilità di funzionare insieme.

## 3 Sistema proposto

Si parte quindi a descrivere una soluzione alle mancanze precedentemente illustrate.

### 3.1 Panoramica

Il sistema adotta un’architettura MVP (Model View Presenter). Essa deriva da MVC ed è l’architettura consigliata da Google per lo sviluppo Android. Rispetto ad MVC, MVP rende il testing più semplice ed i cambiamenti al codice più immediati fornendo anche un maggior disaccoppiamento.

Dal punto di vista di un’applicazione Android:

- La View è relativa ai componenti visibili su schermo e quindi quelli con cui l’utente interagisce direttamente. Essa informa il Presenter

Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022

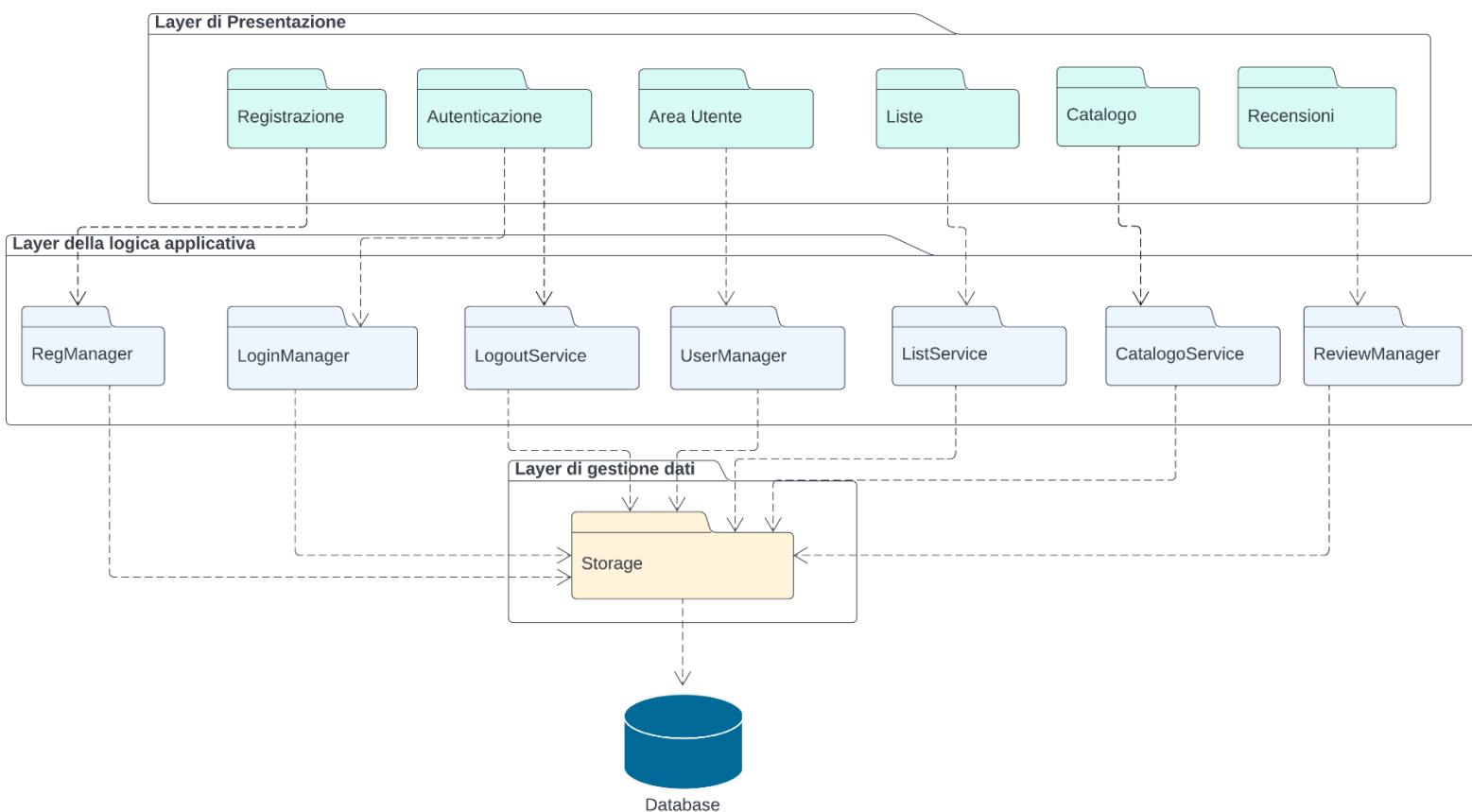
in caso di cambiamenti nell'interfaccia utente ed è rappresentata dalle Activity.

- ~ Il Model si occupa di gestire la logica di business e della comunicazione con il database e il network layer;
- ~ Il Presenter mette in comunicazione la View ed il Model ed è il loro unico modo di scambiarsi informazioni. Esso si occupa di ottenere i dati dal Model, di gestire cosa visualizzare nell'interfaccia utente e degli aggiornamenti di quest'ultima. Si occupa inoltre di gestire eventi ed input dell'utente.

La grande differenza con un tipico MVC è che tra View e Model non vi è alcun collegamento. La View non sa dell'esistenza del Model e viceversa. Il Presenter decide cosa succede quando si interagisce con la View.



## 3.2 Decomposizione in sottosistemi



Registrazione	Sottosistema che fornisce all'utente la funzione di registrazione.
Autenticazione	Sottosistema che fornisce all'utente le funzioni di login e logout.
Area Utente	Sottosistema che fornisce all'utente le funzioni di visualizzazione del profilo, cambio password, cambio e-mail, cambio immagine di profilo ed eliminazione dell'account.
Liste	Sottosistema che fornisce all'utente le funzioni di aggiunta ed

Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022

	eliminazione di libri dalla lista dei libri letti e dalla lista dei libri da leggere.
Recensioni	Sottosistema che fornisce all'utente le funzioni di aggiunta e rimozione di recensioni e lettura di recensioni.
Catalogo	Sottosistema che fornisce al cataloghista le funzioni di aggiunta e rimozione dei libri dal catalogo.

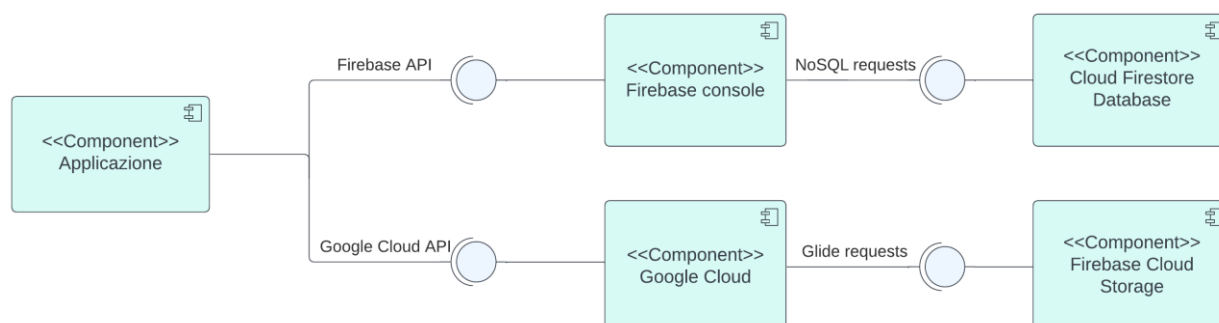
Storage	Lo Storage è un sottosistema che permette di comunicare con il Database. Esso esiste per ottenere basso accoppiamento: in questo modo cambiare tipo di database diventa più facile perché non devo modificare tutti i sottosistemi che lo usano ma solo lo Storage.
---------	---

### 3.3 Mappatura Hardware/Software

Il sistema utilizza un'architettura serverless integrata in un sistema di Cloud Messaging basato sui servizi Firebase e Google Cloud. Firebase inoltre fornisce un database NoSQL chiamato Firestore. Firebase permette di portare parte della logica applicativa nel client rendendo lo sviluppo più semplice e le policy di sicurezza fornite da Firestore permettono di ottenere un'architettura serverless. Infine Google Cloud fornisce uno storage in cui è possibile memorizzare immagini senza appesantire il database.

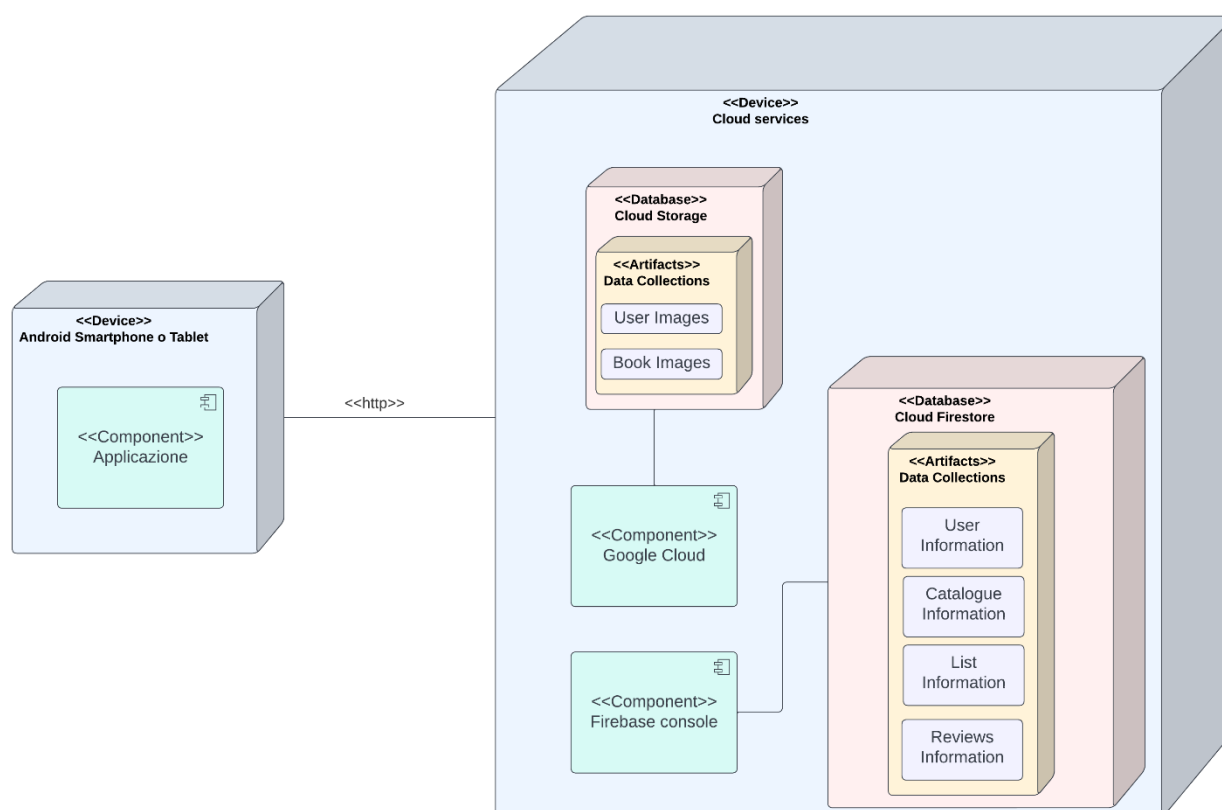
Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022

## Component Diagram:



Glide è una libreria esterna che permette di gestire con facilità l'ottenimento delle immagini dallo storage di Firebase.

## Deployment Diagram:



Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022

## 3.4 Dati Persistenti

Fare riferimento al documento “Dati Persistenti”.

## 3.5 Controllo accessi e Sicurezza

	Utente	Catalogo	Libro	Liste	Recensione
Ospite	Registrazione e Login	deny	deny	deny	deny
Iscritto	Login e Logout	Lettura	Lettura	Lettura e Scrittura	Lettura e Scrittura
Cataloghista	Login e Logout	Lettura e Scrittura	Lettura e Scrittura	deny	deny

I dati sensibili subiranno una crittazione. L'autenticazione sarà implementata utilizzando funzioni già esistenti di Firebase e verrà utilizzato il protocollo HTTPS.

## 3.6 Controllo Globale del Software

Il sistema Android predilige un control flow di tipo parzialmente event-driven: consideriamo i diversi listener `onClick()` oppure il ciclo di vita delle Activity formato da eventi che triggerano operazioni. Tuttavia ogni `onClick()` è associato ad una specifica View e quindi il loro scope è ridotto e la centralizzazione è impossibile.

Inoltre Firestore è asincrono quindi serviranno delle Callback per sapere quando saranno pronti i risultati richiesti.

Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022

## 3.7 Boundary Conditions

### Start-up

Per lo start-up del sistema è necessario l'avvio di un'infrastruttura remota per cloud messaging che fornirà accesso ad un database. L'interfaccia iniziale del sistema è costituita da un layout con pulsanti per la registrazione o login.

### Shutdown

Lo shutdown è essenzialmente gestito dal ciclo di vita delle Activity e viene eseguito con il susseguirsi dei metodi onPause(), onStop() e onDestroy(). Verrà aggiunta della logica per assicurare la consistenza dei dati.

### Errori

Nel caso in cui il dispositivo si spenga a causa della batteria scarica l'applicazione verrà chiusa improvvisamente e non sono previsti metodi per la gestione di tale problema.

Nel caso di crash inaspettati del sistema non sono previsti accorgimenti per assicurare la consistenza dei dati.

Nel caso in cui si venga disconnessi da Internet non si assicura il funzionamento dell'applicazione.

Nel caso di sovraccarico del database il problema verrà gestito nel minor tempo possibile una volta individuato dagli sviluppatori.

## 4.Servizi dei sottosistemi

Registrazione	
Servizio	Descrizione
validateRegAll(...)	Permette di controllare la correttezza delle informazioni fornite dall'ospite.

Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022

createNewUser(...)	Permette di rendere un ospite un iscritto con l'aggiunta di nuove funzionalità.
--------------------	---

Autenticazione	
Servizio	Descrizione
validateLoginAll(...)	Permette di controllare la validità delle credenziali fornite dall'ospite.
isCataloghist(...)	Permette di controllare se l'utente che sta cercando di autenticarsi è un cataloghista
deleteSession(Utente utente)	Permette di eliminare l'utente dalla sessione de autenticandolo.

Area Utente	
Servizio	Descrizione
Getters	Permettono di ottenere una specifica informazione dall'utente come il nome utente, l'email, ecc...
Setters	Permettono di modificare una specifica informazione dall'utente come il nome utente, l'email, ecc...
validateFormatPassword(...)	Permette di controllare che la vecchia password e la nuova password fornite abbiano il giusto formato.
validateFormat(...)	Permette di controllare che la vecchia password e la nuova mail fornite abbiano il giusto formato.
checkOldPass(...)	Permette di controllare che la password vecchia fornita corrisponda.

Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022

Modifica e-mail,password ed immagine di profilo	Permette la modifica di e-mail, password ed immagine di profilo.
deleteUser(String nomeUtente)	Permette di eliminare l'account di un utente.
deleteLists(String nomeUtente)	Permette di eliminare le liste dei libri già letti e da leggere di un utente.

Liste	
Servizio	Descrizione
getReadBook(String ISBN)	Permette di ottenere un libro dalla lista dei libri già letti in base ad un ISBN specificato.
insertAR(Libro libro)	Permette di aggiungere un libro alla lista dei libri già letti.
removeARBook(String ISBN)	Permette di rimuovere un libro dalla lista dei libri già letti.
getToBeReadBook(String ISBN)	Permette di ottenere un libro dalla lista dei libri da leggere in base ad un ISBN specificato.
insertTBRBook(Libro libro)	Permette di aggiungere un libro alla lista dei libri da leggere.
removeTBRBook(String ISBN)	Permette di rimuovere un libro dalla lista dei libri da leggere.
searchBook(String titolo)	Permette di cercare un libro in base al titolo.

Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022

Catalogo	
Servizio	Descrizione
getBookList(String titolo)	Permette di ottenere una lista di libri in base ad un titolo specificato.
getBook(String ISBN)	Permette di ottenere un libro in base ad un ISBN specificato.
addBook(Libro libro)	Permette di aggiungere un libro al catalogo.
checkInfo(...)	Permette di controllare che le informazioni fornite per l'aggiunta di un libro al catalogo siano corrette.
removeBook(String ISBN)	Permette di rimuovere un libro dal catalogo in base ad un ISBN specificato.

Recensioni	
Servizio	Descrizione
Getters	Permettono di ottenere informazioni di una recensione come il punteggio e la descrizione.
Setters	Permettono di modificare informazioni di una recensione come il punteggio e la descrizione.
checkIfSecond(...)	Controlla se un utente sta cercando di aggiungere una seconda recensione ad un libro.
valiateReviewFields(...)	Permette di controllare che le informazioni fornite per l'aggiunta di una recensione abbiano il formato corretto.
removeReview(...)	Permette di rimuovere una recensione.



Progetto: Book Bearer	Versione: 1.0
Documento: System Design Document	Data: 20/12/2022

createnewReview(...)	Permette di aggiungere una recensione ad un libro.
getReviewList(String ISBN)	Permette di ottenere tutte le recensioni di un libro in base ad un dato ISBN.

## 5. Glossario

Serverless = senza l'uso di un server.

Callback = funzione trasmessa ad un'altra come parametro.