**Scadenza:  5 ottobre, non versione completa**

***Titolo***

descrizione breve

indice

***obiettivi del progetto***

scopo dell’app, modalità di esecuzione e funzionamento, immagini? per descrizione di layout, sotto obiettivi, descrizione di alto livello

Consistenza tra obiettivi e requisiti, elencare le funzionalità più importanti

***requisiti funzionali***

richieste specifiche, funzioni che ci si aspetta dal sistema

RF1.1, RF 1.2 etc.  coprire casistica se tutto va bene , ma anche casistica negativa o se il processo non va tutto liscio

***requisiti non funzionali***

cosa si aspetta  utente dall'applicazione (tutela privacy e dati sensibili art. 5 comma 1) autorizzazioni, numero di utenti stimati e previsioni di utilizzo

***design del frontend***

Come ci si aspetta che l'applicazione interagisca con l’utente (link nel testo ai requisiti funzionali, segnalibri) interfacce utili per spiegare il progetto

***design del backend***

visione di insieme delle interazioni che la nostra app deve fare (quali database utilizza, quali software sono richiesti)

**Scopo del documento**

Il presente documento riporta l’analisi dei requisiti di sistema del progetto di “nome sito”, in linguaggio naturale.

Il suo scopo è quello di

-       presentare gli obiettivi del progetto;

-       definire i requisiti funzionali e non funzionali;

-       presentare i requisiti di Front-End;

-       presentare i requisiti di Back-End.

***1.    Obiettivi***

Il progetto ha come fine quello di realizzare un’applicazione per la gestione di una biblioteca.

In particolare applicazione permetterà:

a)     Ad un utente autenticato, ovvero ad un utente provvisto di credenziali (mail e password) di poter

-       Poter controllare quali libri sono disponibili per il prestito e quali no

-       Donare libri alla biblioteca

-       Poter richiedere il prestito di un libro dalla biblioteca

-       Di poter lasciare una recensione del libro, una volta riconsegnato

b)    Ad un utente anonimo, cioè ad un utente il quale non ha ancora effettuato l’accesso al sistema, di poter

* Registrare nel sistema

-       Controllare quali libri sono disponibili per il prestito e quali no

-       Poter effettuare una ricerca del libro di proprio interesse in base al suo titolo

-       Poter leggere le eventuali recensioni del libro lasciate da lettori precedenti

Nel caso in cui l’utente anonimo cercherà di effettuare azioni per le quali non ha alcuna autorizzazione, gli verrà mostrato un messaggio d’errore (mettere collegamento ipertestuale e foto)

c) ad un utente amministratore di poter aggiornare il database nelle seguenti casistiche

* rendere disponibile un libro una volta che un utente ha effettuato fisicamente il reso di quest’ultimo
* aggiungere un libro al database dopo che un utente ha effettuato una donazione ed aver portato fisicamente il libro in biblioteca

c)     Di mandare all’utente autenticato, grazie al collegamento tramite Gmail, delle email per le seguenti casistiche

-       Scadenza prossima consegna

-       Multa per mancato reso del libro preso in prestito

-       Conferma del prestito e luogo nel quale si può ritirare il libro

-       Recensione effettuata

-       Donazione del libro correttamente effettuata

**2.   Requisiti Funzionali**

Nel seguente paragrafo vengono elencati i requisiti funzionali (RF) dell’applicazione.

**Requisiti di accesso:**

* **RF 1:**
* Il sistema dovrà avere tre livelli di accesso:

        1.Utente anonimo;

2.Utente verificato;

3.Utente amministratore

**Utente anonimo:**

* **RF 2.1   Registrazione al sistema**

L’applicazione deve permettere all’utente di effettuare il sign in al sistema bibliotecario, in caso di errori nella registrazione l’applicazione li segnalerà e permetterà di effettuare un altro tentativo. La registrazione verrà effettuata tramite: nome, cognome e email (strettamente del tipo nome.cognome@....., (password))

* **RF 2.2   Utilizzo senza verifica**

L’applicazione dovrà permettere all’utente anonimo di utilizzare il sistema per:

* La ricerca di libri nel database della biblioteca.
* Controllare la disponibilità di un dato libro.

L’utente anonimo non potrà in alcun modo usufruire di altre funzionalità presenti dell’applicazione.

**Utente verificato:**

* **RF 3.1     Accesso al sistema**

L’applicazione deve permettere all’utente verificato di accedere alla propria area personale del sistema, con le proprie credenziali (nome,cognome,mail,(password)), in caso di credenziali errate l’applicazione permetterà di effettuare un altro tentativo.

* **RF 3.2 Prenotazione appuntamento**

L’utente sarà in grado di poter prenotare un

appuntamento in biblioteca in modo da poter portare

fisicamente il libro che si vuole donare (rif RF 8)

* **RF 3.3**  **Utilizzo sistema**

L’applicazione dovrà fornire all’utente verificato le seguenti funzionalità:

* Ricerca di libri nel database della biblioteca
* Controllare la disponibilità di un dato libro
* Prenotare libri fino ad un massimo di 5 libri
* Prendere appuntamento per donare propri libri
* Verificare lo stato del reso dei libri noleggiati
* Valutare libri dopo averne fatto il reso

**Utente amministratore**

* **RF 4.1    Accesso**

L’applicazione dovrà permettere l’accesso tramite nome, cognome e codice di verifica all’utente amministratore.

* **RF 4.2    Controllo utenti**

L’applicazione dovrà permettere all’utente amministratore di controllare la situazione di ogni utente verificato.

* **RF 4.3    Resi**

L’applicazione dovrà permettere all’utente amministratore di confermare i resi degli utenti e rendere i libri di nuovo disponibili sull’archivio.

* **RF 4.4    Donazioni**

L’applicazione dovrà permettere all’utente amministratore di aggiornare l’archivio libri dopo le  donazioni effettuate, come esplicitato in RF 8.

**Sistema di ricerca**

* **RF 5.1 Barra ricerca**

L’applicazione dovrà fornire un sistema di ricerca sull’archivio della biblioteca tramite una barra di ricerca usando titolo del libro oppure nome dell’autore, nel caso non ci fossero riscontri nel database verrà mostrato un messaggio di ricerca fallita.

* **RF 5.2   Filtri di ricerca**

L’applicazione fornirà all’utente dei filtri di ricerca sull’archivio libri, per categoria, autore, ordine alfabetico.

**Sistema di prenotazione**

* **RF 6.1    Verifica disponibilità**

L’applicazione dovrà permettere di verificare la

disponibilità al noleggio di ogni libro

* **RF 6.2    Prenotazione**

L’applicazione dovrà permettere all’utente la prenotazione, di uno o più libri fino ad un massimo di 5, fornirà anche un codice univoco per il ritiro.

**Notifiche**

* **RF 7**

L’applicazione dovrà fornire un sistema di notifica per le comunicazioni con l’utente

* **RF 7.1    Notifiche di conferma prenotazione**

L’applicazione dovrà inviare una notifica di conferma per ogni libro prenotato

* **RF 7.2    Notifiche di conferma reso**

L’applicazione dovrà  inviare una notifica di conferma per ogni reso effettuato

* **RF 7.3    Notifiche di scadenza noleggio**

L’applicazione dovrà  inviare una notifica reminder a cinque giorni dal termine del noleggio di un libro

* **RF 7.4    Notifiche appuntamento donazioni**

L’applicazione dovrà inviare una notifica per ogni appuntamento di donazione libri confermato, con data, ora e luogo

**Donazioni**

* **RF 8**

1. L’applicazione dovrà permettere la donazione di libri compilando un form contenente: titolo, autore.
2. Verificare se il libro in donazione sia presente o meno nell'archivio ed inserirlo.
3. Da definire come le donazioni non vanno in porto.

**Valutazioni**

* **RF 9**

L’applicazione dovrà fornire uno strumento di valutazione dei libri, con una scala da uno a cinque, che si attiverà ogni volta che verrà confermato un reso all’utente.

**MongoDB**

* **RF 10**

L’applicazione dovrà utilizzare i sistemi di MongoDB per lo sviluppo dell’archivio libri della biblioteca e del database utenti.

**Logout**

* **RF 12**

L’applicazione dovrà permettere all’utente verificato e amministratore di effettuare il logout in qualsiasi momento

-utente anonimo. ok

-ut autenticato.ok

-notifica.ok

-verifica accesso(cambio preferenze notifica).

-verifica sull’archivio(ricerca ).ok

-login automatico.

-logout.ok

-contatti.

-database mongodb.ok

-sistema di recensioni.ok

-noleggio/reso.ok

-donazioni ok

-registrazione (strong pass, nome, cognome…)ok

**3.   Requisiti Non Funzionali**

Nel seguente paragrafo vengono descritti i requisiti non funzionali (RNF) dell’applicazione

**Privacy**

* **RNF 1**

L’applicazione deve rispettare la vigenti norme riguardanti la Privacy dell’individuo (art. 5, comma 1),  e le norme imposte dal GDPR.

Ad esempio:

1. Il trattamento dati dovrà essere coerente con l’utilizzo che l’applicazione richiede.
2. In nessun caso potranno essere divulgati i dati personali degli utenti, quali : nome, cognome, email; senza previa autorizzazione
3. Sotto richiesta degli utenti, i dati personali  dovranno essere eliminati dal sistema

**Linguaggio**

* **RNF 2**

L’applicazione dovrà essere fornita sia in lingua italiana e inglese per permettere al meglio la sua comprensione tra tutti gli utenti. I contenuti all’interno dell’applicazione dovranno essere gli stessi per ciascuna delle due lingue.

**Scalabilità**

* **RNF 3**

L’applicazione dovrà garantire l'utilizzo sincrono di un numero elevato di utenti. E’ previsto un aumento significativo del numero di utenti, progressivo dalla data di lancio e proporzionale alla sua diffusione.

**Prestazioni**

* **RNF 4**

L’applicazione deve garantire un tempo massimo di avvio di 3 secondi e essere scorrevole in fase di utilizzo delle varie funzionalità.

**Facilità d’uso**

* **RNF 5**

L'applicazione deve  essere intuitiva all’uso da parte degli  utenti anche senza esperienza e/o senza bisogno di richiedere assistenza.

**Compatibilità**

* **RNF 6**

L’applicazione dovrà essere compatibile con i browser più utilizzati (Chrome, firefox, safari, etc..) dalla versione più attuale (2023) in poi.

In caso di eventuali aggiornamenti dell’applicazione la compatibilità con i browser  non verrà influenzata.

**Sicurezza**

* **RNF 7**

L’applicazione dovrà fornire un livello di sicurezza tale che gli utenti non possano accedere ad informazioni sulle attività di altri utetnti all’interno dell’applicazione, del tipo: prenotazioni effettuate, codici univoci prenotazioni, notifiche di sistema.

-flessibilità evolutiva

-tempo limite login automatico

-sicurezza

**4.  Design del Front-end**

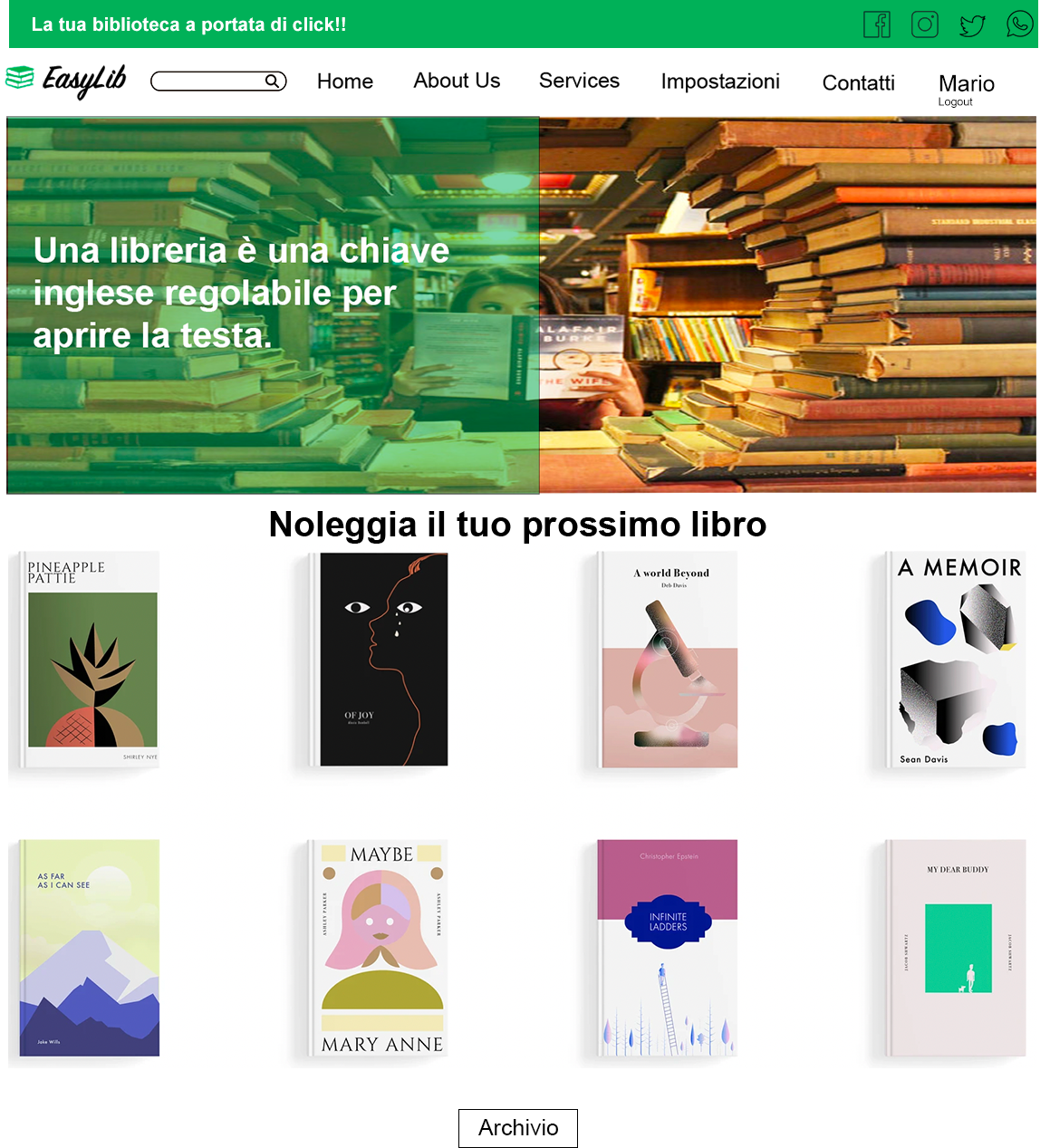
In questo paragrafo verranno riportati alcuni mock-up relativi alle schermate della web-app con l’obiettivo di rappresentare come il sistema si interfaccerà all’utente. Le pagine che presenteremo sono

* + Homepage
  + Pagina di Login
  + Pagina di Registrazione
  + Contatti
  + Impostazioni
  + Archivio
  + Modulo di inserimento libro in archivio

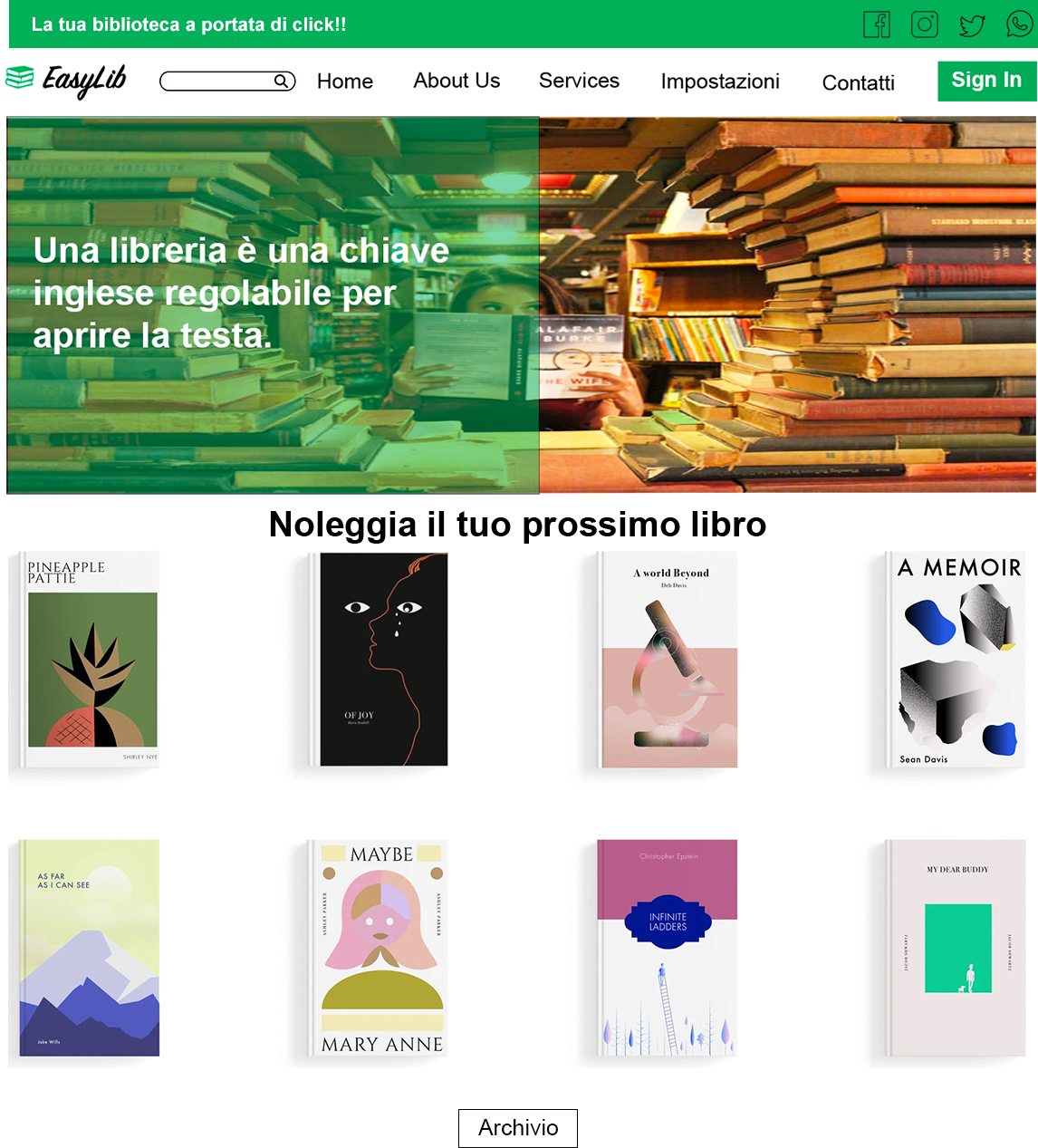
Homepage

Il sistema deve presentare all’utente, appena entrato nel sito, una pagina con una piccola descrizione del progetto e alcuni esempi di libri disponibili al noleggio. Inoltre vi dovrà essere un pulsante il quale, una volta premuto, permetterà all’utente di poter vedere tutti i libri disponibili per il noleggio. In cima alla pagina saranno presenti diverse scritte ognuna delle quali, una volta cliccatoci sopra, permetteranno di raggiungere le rispettive pagine web.

Una volta che l’utente abbia eseguito l’eventuale login, in cima a destra, verrà mostrato il suo nome e al di sotto di quest’ultimo il tasto di logout.



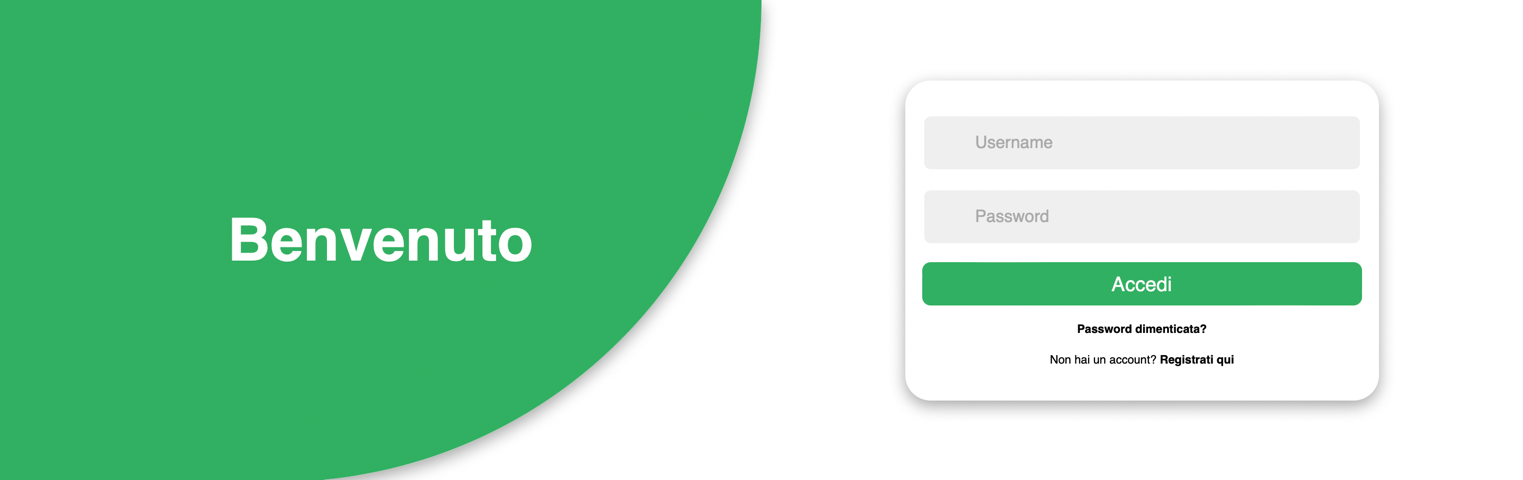
**Figura 4.1:** Homepage utente non loggato



**Figura 4.2:** Homepage utente loggato

Pagina di Login

La pagina di login è accessibile cliccando il tasto ‘Login’ presente in alto ad ogni pagina. La pagina alla sua sinistra mostrerà anche un messaggio di ‘Benvenuto’.



**Figura 4.3:** Schermata di Login

Pagina di Registrazione

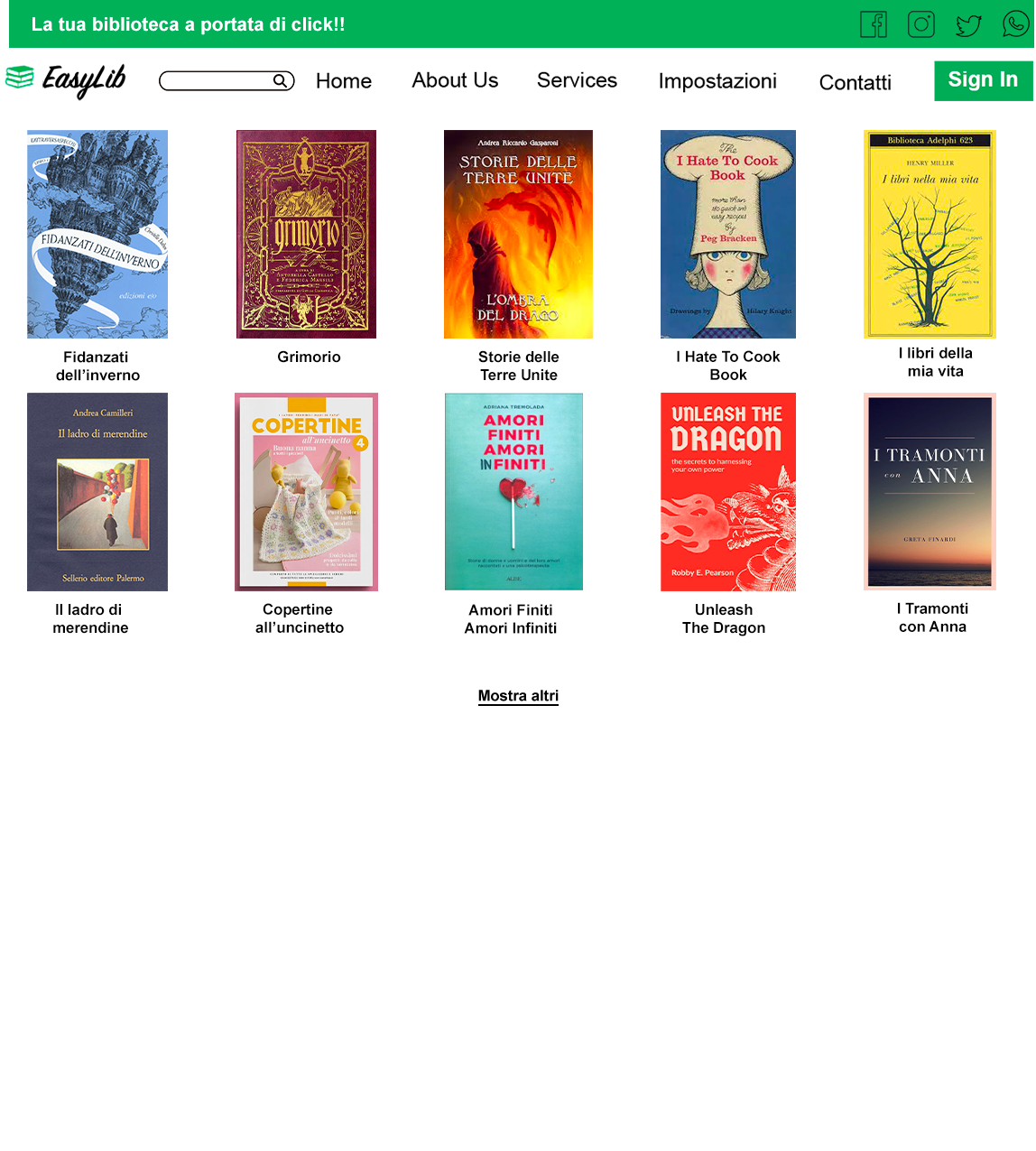
Se una volta cliccato il tasto ‘Sign In’ l’utente vuole registrarsi per unirsi al sistema bibliotecario, gli sarà fornita la possibilità premendo la scritta ‘Registrati’. Una volta fatto ciò la pagina di login, attraverso un’animazione, diventerà la pagina di registrazione in modo da permettere all’utente di registrarsi.



**Figura 4.4:** Schermata di Regisrazione

Archivio

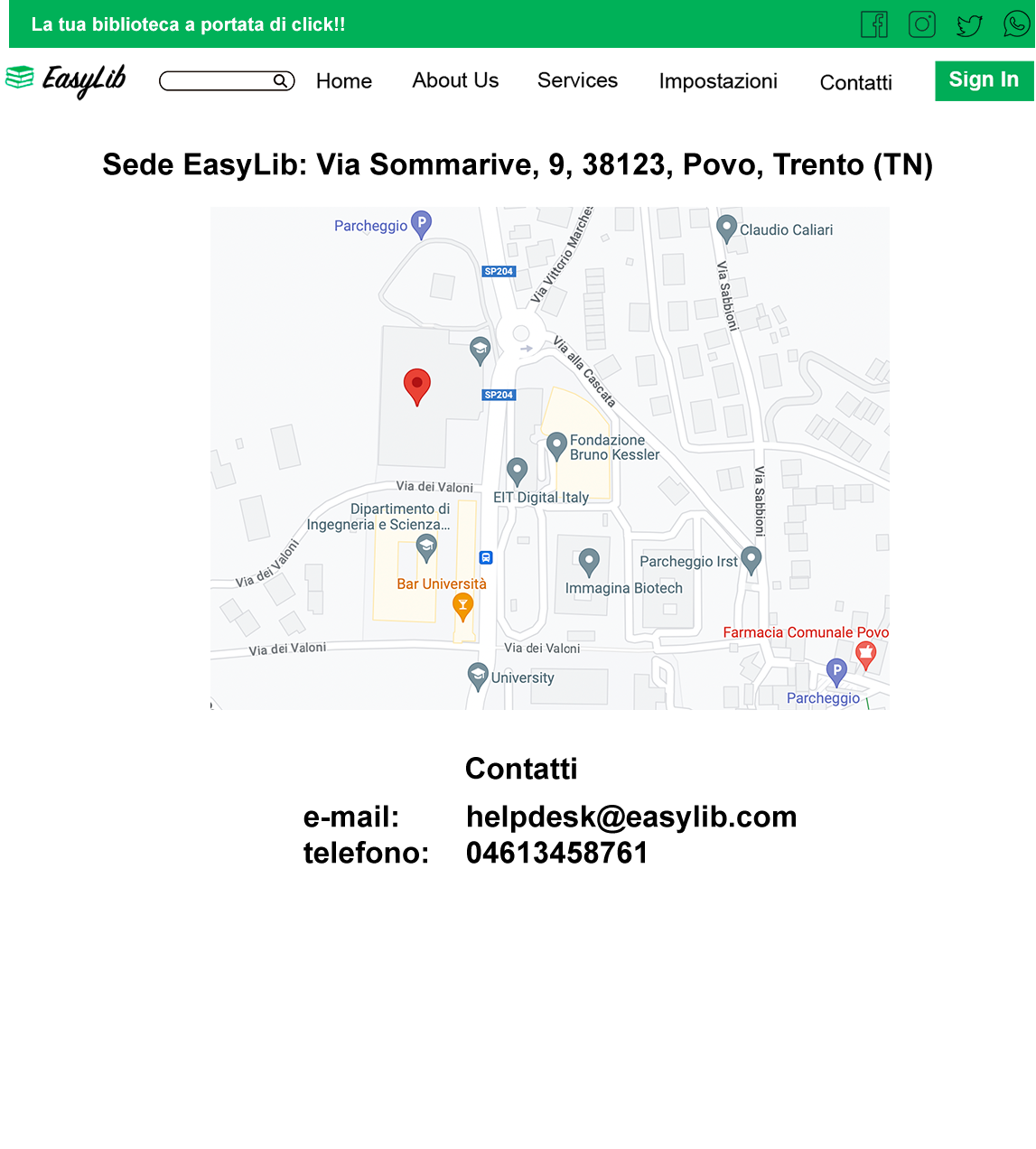
Una volta che l’utente abbia premuto sul tasto ‘Archivio’ presente nella homepage, verrà aperta una pagina ‘archivio’ nella quale verranno mostrati ulteriori libri disponibili per il noleggio. Premendo il testo ‘Mostra più’ verrano mostrati ulteriori libri, se disponibili.



**Figura 4.5:** Archivio

Contatti

Questa pagina contiene una mappa che fa alla posizione della Biblioteca ed mail e numero di telefono per potersi mettere in contatto con la libreria.

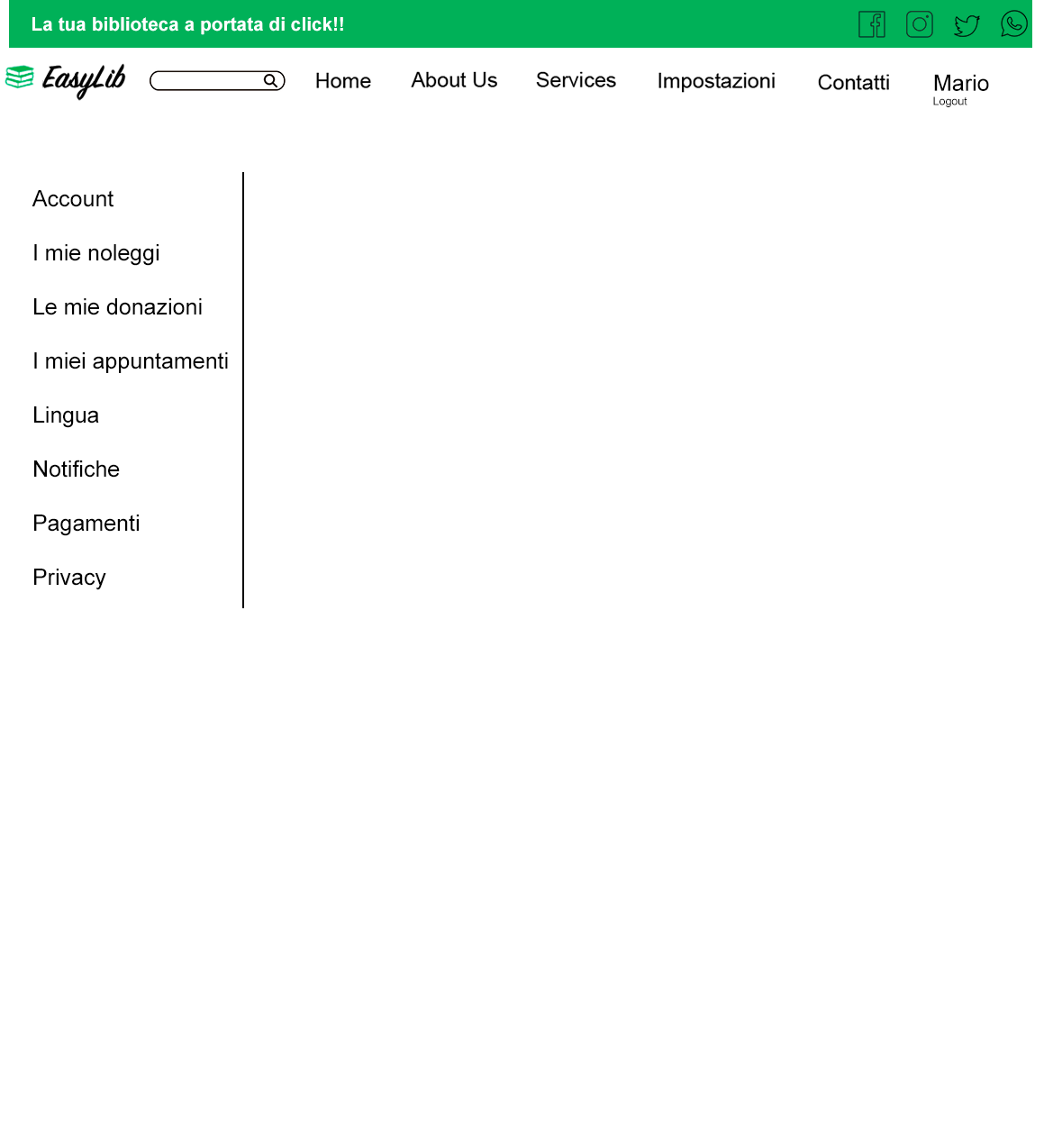


**Figura 4.6:** Contatti

Impostazioni

La pagina contiene sulla sinistra diverse scritte, ognuna delle quali porterà alla sezione apposita. In particolare, nel caso di un utente loggato, dovranno essere presenti le sezioni

* + Account
  + I miei noleggi
  + Le mie donazioni
  + I miei appuntamenti
  + Lingua
  + Notifiche
  + Pagamenti
  + Privacy

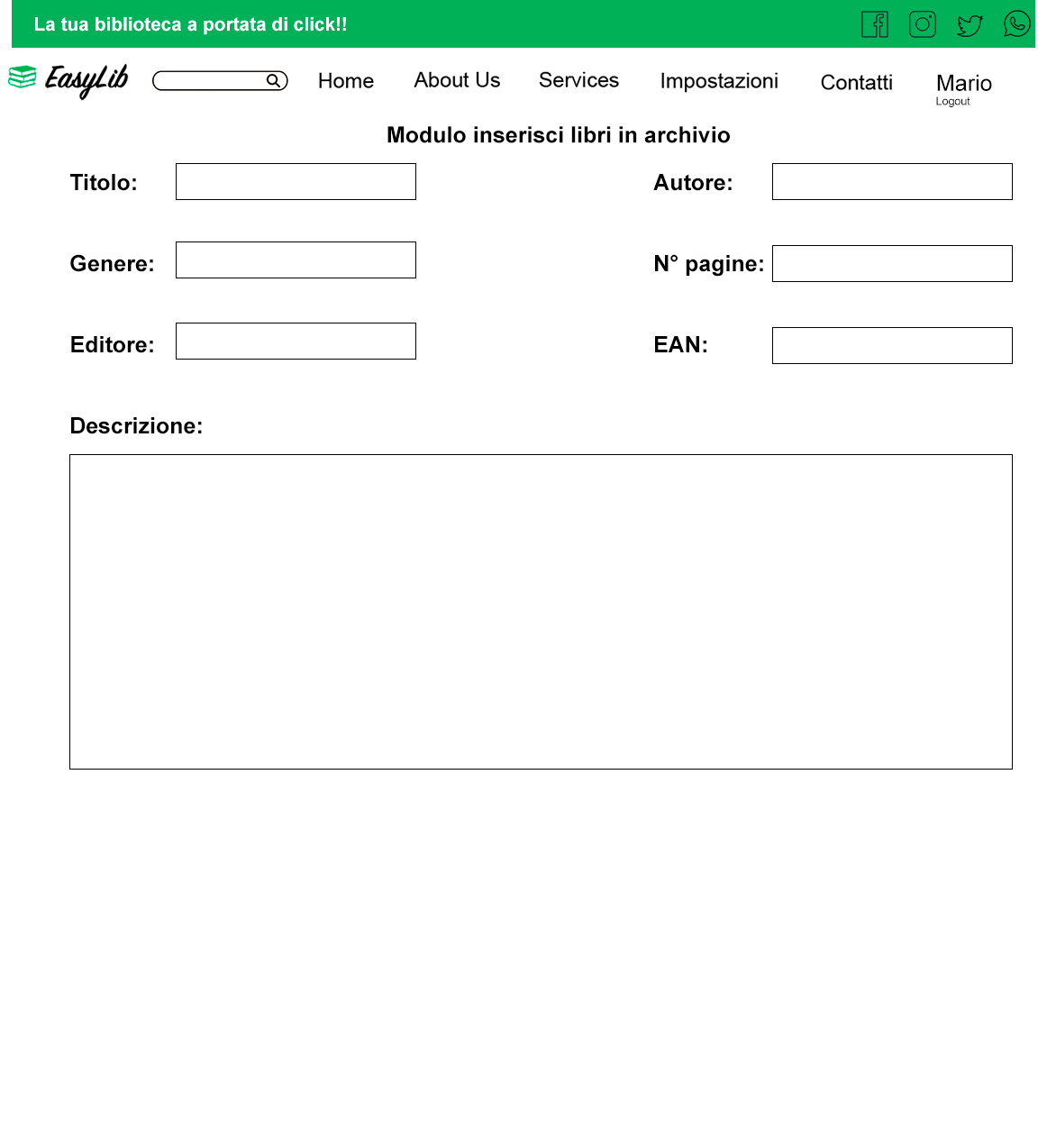


**Figura 4.7:** Impostazioni

Modulo inserisci libri in archivio

Affinchè possa essere inserito un libro nell’archivio, si dovrà riempire un modulo online nel quale saranno richieste le informazioni necessarie. In particolare verranno richieste le info

* + Titolo
  + Autore
  + Genere
  + N° pagine
  + Editore
  + EAN
  + Descrizione



**Figura 4.8:** Modulo inserisci libro in archivio

**5.  Design del Back-end**

Nel presente paragrafo vengono elencati i componenti con cui l’applicazione interagirà esternamente e il loro utilizzo, compreso sottostante un semplice diagramma esplicativo (Figura 5.1).

**Gmail API**

La API di Gmail verrà utilizzata per l’accesso all’applicazione e per la ricezione di avvisi riguardanti notifiche in app.

**MongoDB**

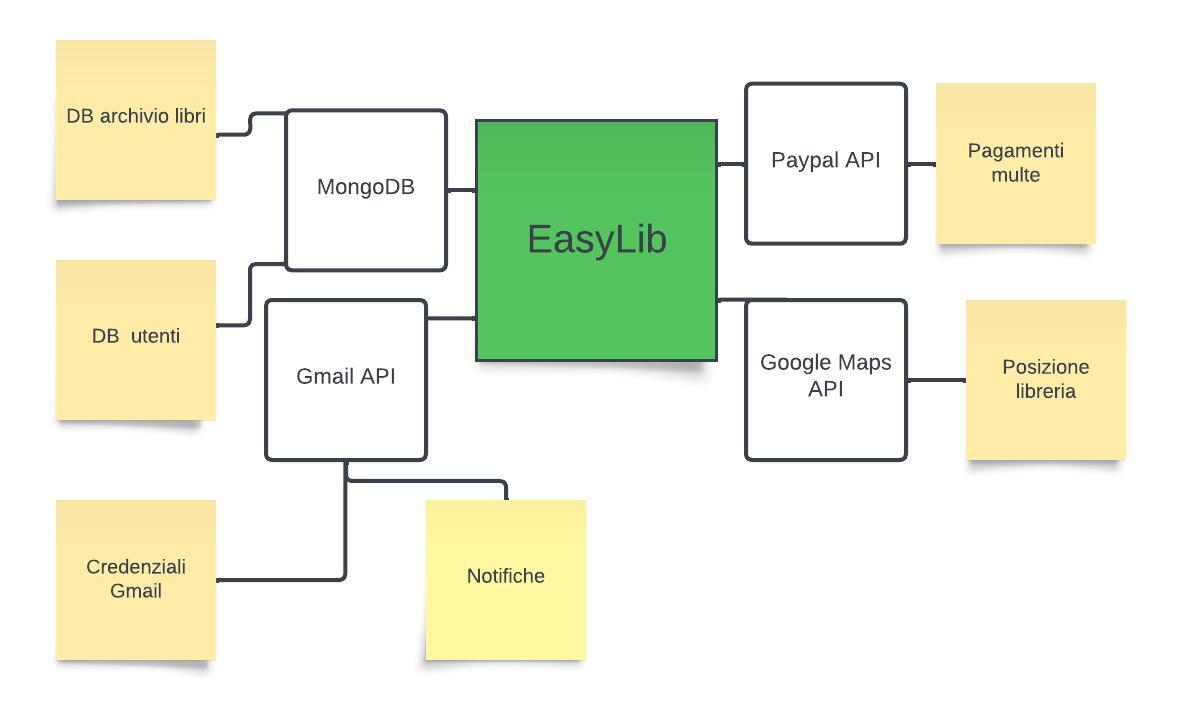
L’applicazione utilizzerà il database di MongoDB per l’archivio virtuale dei libri della biblioteca e per l’archivio degli utenti verificati.

**PayPal API**

Le multe riguardanti libri non restituiti entro il tempo limite verranno pagate tramite l’utilizzo di PayPal.

**Google Maps API**

Per permettere agli utenti di trovare più facilmente la sede fisica della biblioteca l’applicazione utilizza l’API di Google Maps.



**Figura 5.1:** diagramma del back-end