Gestione di un sistema di book-crossing: *The WalkingBooks*

Documentazione progettuale

Progetto del corso di Informatica IIIB
A.A. 2018/2019

Paganessi Andrea
Piffari Michele
Villa Stefano

September 5, 2019

Contents

List of Figures

List of Tables

Part I

Iterazione 0

1

Requisiti e specifiche

1.1 Requisiti utente

La startUp bergamasca Book Crossing UniBg desidera mettere a disposizione dei propri utenti un'applicazione Android per poter gestire la libera condivisione di libri all'interno di una vasta community di utenti.

I libri della rete di BookCrossing che l'azienda punta a gestire si possono trovare

- Casualmente (in stazione, su una panchina, in un locale o in un qualsiasi altro luogo): funzionalità *"on the go"*
- Nella zona di scambio ufficiale (*"OCZ UniBg"*: Official Crossing Zone UniBg)

Per il momento, l'unica OCZ gestita direttamente dalla startUp si trova all'interno dell'aula studio del campus di Ingegneria di Dalmine, la quale coincide anche con la sistemazione del server centrale che andrà a gestire i vari interscambi tra gli users.

La startUp richiede che, per usufruire dell'applicativo mobile, i clienti debbano registrarsi fornendo i propri dati quali

- Nome
- Cognome
- Contatto di riferimento (opzionale)
  - Numero telefonico
  - Indirizzo mail
  - Facebook
  - ID Twitter
- Categorie di libri preferite
- Zona di residenza
- Raggio d'azione (inteso come l'area, in km, entro cui l'utente è disposto a spostarsi per incontrare altri utenti con cui scambiarsi libri)

Una volta registratosi, l'utente può partecipare al programma di Book Crossing.

Secondo la politica del book sharing, per rendere disponibile alla comunità uno o più libri che non sono ancora presenti nel network stesso, serve identificarli univocamente, per poterne così tracciare la storia, ovvero ciò che concerne il percorso seguito dal libro, le recensioni lasciate dagli utenti etc.

Prima di procedere con l'identificazione univoca del libro, l'utilizzatore deve inserire i dati del testo (o dei testi) che intende condividere con il resto della community: questo inserimento può avvenire

- In maniera "automatica" tramite scansione del codice ISBN
- In modalità "manuale", nel caso in cui, per esempio, non sia presente il barcode

fornendo i seguenti dati:

- Titolo
- Autore
- Anno di pubblicazione/Edizione
- Categoria

A questo punto il sistema genererà un BCID di 10 caratteri, ovvero un *Book Crossing ID* univoco, il quale dovrà essere riportato sul testo dall'utente.

La vera e propria condivisione avviene nel momento in cui il volume/i viene rilasciato (azione che può avvenire in un secondo momento rispetto alla fase di identificazione), il sistema dovrà acquisire i seguenti dati:

- Luogo di rilascio (con estensione future per un'acquisizione automatica della posizione tramite GPS)
- Ora e data di rilascio

L'app inoltre consiglierà all'utente un luogo di rilascio in cui sia già presente almeno un libro, facilitando così la creazione di cassette virtuali, ovvero di luoghi in cui sono presenti più libri: l'idea è quella quindi di permettere al sistema di creare, in maniera autonoma, dei punti "fissi" di consegna senza dover applicare interventi a livello infrastrutturale.

Successivamente il sistema dovrà notificare gli utenti, interessati al genere del libro rilasciato, della presenza di un nuovo testo, appena rilasciato, che potrebbe interessargli.

In qualsiasi momento è possibile effettuare le seguenti operezioni su ogni libro personalmente condiviso con la rete di sharing:

- Aggiunta di recensione
- Rating del libro

Quando viene trovato un libro (nel gergo definito come *"journal entry"*), il cliente che vuole prelevarlo, dopo aver effettuato il login nell'applicazione, deve inserire nell'apposito menù il BCID del libro che intende acquisire. Il sistema si occuperà poi di informare la community aggiornando lo status del libro raccolto, che diventerà "underReading".

Per quanto concerne invece l'area riservata, ogni utente ha la possibilità di visualizzare informazioni in merito ai libri che:

- ha messo a disposizione della community (**relased**)
- ha ottenuto dalla community (**chased**)
- attualmente possiede

L'utente può effettuare la prenotazione di libri già inseriti nella liste "chased" e "relased" del proprio profilo.

Il sistema deve prevedere anche la possibilità di ricercare un specifico testo e visualizzare i contatti dei lettori del libro al fine di potersi scambiare opinioni e/o pareri in merito al libro stesso. Tale funzionalità di ricerca permette anche la prenotazione del testo ricercato purché lo stesso sia nello stato "under reading". Per soddisfare questa richiesta il sistema provvederà a consigliare, al lettore corrente del libro prenotato, zone di rilascio specifiche al fine di avvicinare tale libro al richiedente, tenendo presente anche la necessità di creare cassette virtuali (come specificato in precedenza).

2

Use cases

2.1 Analisi testuale dei casi d'uso

- ***UC1: Registration***
  - **Descrizione:** registrazione alla rete di Book Crossing.
  - **Attori coinvolti:** utente.
  - **Preconditions:**
    - smartphone dotato di connessione dati;
    - l'utente non è ancora registrato al programma di Book Crossing.
  - **Postconditions:** l'utente è registrato al programma di Book Crossing.
  - **Processo:**
    1. l'utente seleziona "Registrati" nella schermata inziale dell'applicazione;
    2. l'applicazione mostra un form in cui l'utente può inserire:
       a. Nome
       b. Cognome
       c. Data di nascita
       d. Username
       e. Password
       f. Contatto di riferimento
       g. Categorie di libri preferite
       h. Zona di residenza
       i. Raggio d'azione rispetto alla zona di residenza
    3. l'utente inserisce i dati richiesti nel form presentato;
    4. l'utente attende la visualizzazione della conferma di avvenuta registrazione;
    5. l'utente viene reindirizzato alla pagina principale dell'applicazione.
  - **Alternative**
    - **Dati non validi:** se l'utente inserisce dei dati non validi e/o mancanti, l'applicazione mostra un messaggio d'errore permettendo all'utente di modificare i dati non validi.
  - **Estensioni**
- ***UC2: Login***
  - **Descrizione:** accesso alla rete di Book Crossing.
  - **Attori coinvolti:** utente.
  - **Preconditions:**
    - smartphone dotato di connessione dati;

    - l'utente è già registrato al programma di Book Crossing.
  - **Postconditions:** l'utente è loggato nella rete di Book Crossing.
  - **Processo:**
    1. l'utente seleziona "Login" nella schermata iniziale dell'applicazione;
    2. l'applicazione mostra una schermata in cui l'utente inserisce username e password;
    3. l'utente inserisce i dati richiesti nella view presentata;
    4. l'utente attende la verifica della correttezza dei dati inseriti;
    5. l'utente viene reindirizzato alla pagina principale dell'applicazione.
  - **Alternative**
    - **Username e/o password non corretti:** se l'utente inserisce username e/o password non validi, l'applicazione mostra un messaggio d'errore permettendo all'utente di modificare i dati.
  - **Estensioni**

- ***UC3: Logout***
  - **Descrizione:** disconnessione profilo personale.
  - **Attori coinvolti:** utente.
  - **Preconditions:**
    - smartphone dotato di connessione dati;
    - l'utente è già registrato al programma di Book Crossing;
    - l'utente è loggato.
  - **Postconditions:** l'utente non è più loggato nella rete di Book Crossing.
  - **Processo:**
    1. l'utente seleziona "Profilo personale" nella schermata principale dell'applicazione;
    2. l'applicazione mostra una schermata in cui l'utente può visualizzare tutte le proprie informazioni;
    3. l'utente preme il bottone "Logout";
    4. l'utente viene reindirizzato alla pagina di login.

- ***UC4: Book pick-up***
  - **Descrizione:** raccolta di un libro "on the go"[1] o in una OCZ.[2]
  - **Attori coinvolti:** utente.
  - **Preconditions:**
    - smartphone dotato di connessione dati;
    - l'utente ha effettuato l'accesso alla rete di Book Crossing;
    - il libro è stato siglato con il codice BCID.
  - **Postconditions:** Il libro viene associato all'utente.
  - **Processo:**
    1. l'utente seleziona "Raccogli libro" nel menu principale dell'applicazione;
    2. l'applicazione mostra un form in cui l'utente può inserire il BCID del libro;
    3. l'utente inserisce il codice BCID riportato nel libro;
    4. l'utente attende la visualizzazione di una scheda riepilogativa relativa al libro appena aggiunto;

[1] In questo caso il libro condiviso dalla community viene raccolto dall'utente in una qualsiasi zona (come per esempio la stazione, il parco, sale di attesa etc...).

[2] *Official Crossing Zone*: zone riconosciute e fisse in cui la community può liberamente scambiarsi i libri.

5. l'utente verifica la corrispondenza delle informazioni mostrate;
6. l'utente conferma la raccolta.

– **Alternative**

* **BCID inesistente:** se l'utente inserisce un BCID non esistente, l'applicazione mostra un messaggio d'errore permettendo all'utente di modificare il BCID.
* **BCID associato ad un altro utente:** se l'utente inserisce un BCID già in possesso di un'altro utente, l'applicazione mostra un messaggio d'errore permettendo all'utente di modificare il BCID.
* **BCID non corrispondente:** se l'utente, al punto (4), verifica che il libro reale non corrisponde alle informazioni mostrate dall'applicazione, può annullare l'operazione di raccolta.

– **Estensioni**

- ***UC5: Book registration***

– **Descrizione:** registrazione di un libro alla rete di Book Crossing (*journal entry*).

– **Generalizzazione di:**

* aggiunta manuale dei dati del libro (UC8);
* scansione ISBN (UC9).

– **Include:** scrittura BCID (UC10).

– **Attori coinvolti:** utente.

– **Preconditions:**

* smartphone dotato di connessione dati;
* l'utente ha effettuato l'accesso alla rete di Book Crossing;
* il libro non è ancora stato siglato con il codice BCID.

– **Postconditions:** Il libro viene registrato alla rete di Book Crossing.

– **Processo:**

1. l'utente seleziona "Registra un nuovo libro" nel menu principale dell'applicazione;
2. l'applicazione tenta di iniziare la scansione ISBN mostrando all'utente la fotocamera;
3. l'utente inquadra il codice ISBN del libro per il tempo sufficiente al riconoscimento del codice stesso;
4. l'applicazione mostra all'utente la schermata contente tutti le informazioni del libro;
5. l'utente preme il pulsante "Conferma registrazione", dopo aver verficato rapidamente la coerenza dei dati.

– **Alternative**

* **Aggiunta manuale dei dati:** se, dalla schermata di scansione, l'utente decide di inserire manualmente i dati del libro da registrate, l'applicazione mostra una schermata dove aggiungere manualmente i dati del libro.
* **Scansione fallita:** se la scansione fallisce, l'applicazione riapre la fotocamenra permettendo all'utente di ripetere l'operazione.

– **Estensioni**

- ***UC6: Book research***

– **Descrizione:** ricerca di un libro all'interno della rete di Book Crossing.

– **Attori coinvolti:** utente.

– **Preconditions:**

* smartphone dotato di connessione dati;

    - l'utente ha effettuato l'accesso alla rete di Book Crossing;
    - il libro è presente nella rete di Book Crossing.
  - **Postconditions:** il libro ricercato viene mostrato.
  - **Processo:**
    1. l'utente seleziona “Ricerca libro” nel menu principale dell'applicazione;
    2. l'applicazione mostra all'utente un form da completare in cui inserire i parametri della ricerca;
    3. l'utente inserisce i parametri a cui è interessato;
    4. l'utente preme il pulsante "Cerca" ed attende il completamento;
    5. l'utente visualizza la lista di tutti i libri nella rete, che soddisfano la ricerca.
    6. il sistema verifica la presenza del libro cercato;
    7. in caso di esito positivo, l'applicazione mostra una scheda riassuntiva del libro;
    8. in caso di esito negativo, l'applicazione mostrerà un messaggio di errore.
  - **Alternative**
    - **Parametri non validi:** se l'utente inserisce dei parametri non validi, l'applicazione mostra un messaggio d'errore permettendo all'utente di modificarli.
    - **Ricerca senza risultati:** se la ricerca non va a buon fine, l'applicazione mostra un messaggio all'utente, comunicando che nessun libro presente nella rete soddisfa i parametri di ricerca inseriti.
  - **Estensioni**
    - L'utente può selezionare uno dei libri mostrati dall'applicazione e visualizzare le sue informazioni.

- ***UC7: Info visualization***
  - **Descrizione:** Visualizzazione informazioni
  - **Generalizzazione:** visualizzazione informazioni libri chased (UC13), visualizzazione informazioni libri relase d(UC14) e visualizzazione informazioni libri in possesso (UC15).
  - **Attori coinvolti:** utente.
  - **Preconditions:**
    - smartphone dotato di connessione dati;
    - l'utente ha effettuato l'accesso alla rete di Book Crossing.
  - **Postconditions:** L'applicazione mostra le informazioni desiderate
  - **Processo:**
    1. l'utente seleziona la voce “I miei libri” nel menu principale dell'applicazione;
    2. l'applicazione mostra categorie di informazioni visualizzabili;
    3. l'utente seleziona la categoria che vuole visualizzare;
    4. l'applicazione mostra l'elenco dei libri della categoria selezionata.
  - **Alternative**
    - **Nessun libro in elenco:** se nessun libro è presente nello storico, l'applicazione mostra un messaggio all'utente, comunicando che non è stata ancora effettuata nessuna operazione nella comunità.
  - **Estensioni**

- ***UC8: Personal area visualization***
  - **Descrizione:** Visualizzazione profilo personale utente
  - **Attori coinvolti:** utente.

  - **Preconditions:**
    * smartphone dotato di connessione dati;
    * l'utente ha effettuato l'accesso alla rete di Book Crossing;
  - **Postconditions:** l'applicazione mostra il profilo dell'utente.
  - **Processo:**
    1. l'utente seleziona la voce "Il mio profilo" nel menu principale dell'applicazione;
    2. l'applicazione mostra l'anagrafica, i contatti e le attività svolte dall'utente;
  - **Estensioni:**

- ***UC9: Manual addition of book's data***
  - **Descrizione:** Inserimento manuale di un libro nella rete di Book Crossing
  - **Attori coinvolti:** utente.
  - **Preconditions:**
    * smartphone dotato di connessione dati;
    * l'utente ha effettuato l'accesso alla rete di Book Crossing;
    * il libro non è stato ancora siglato con il codice BCID.
  - **Postcondition:** Viene generato il codice BCID e il libro viene aggiunto alla rete di Book Crossing
  - **Processo:**
    1. facendo riferimento al passo 1 e 2 del UC4, l'utente preme il pulsante "Aggiunta manuale";
    2. l'applicazione mostra un form da compilare con i dati del libro;
    3. l'utente inserisce i dati del libro richiesti e conferma l'operazione;
    4. l'applicazione mostra il codice BCID da trascrivere sul libro;
    5. il sistema aggiunge il libro alla rete di Book Crossing.
  - **Estensioni**

- ***UC10: ISBN scan***
  - **Descrizione:** scansione del codice ISBN tramite fotocamera per ottenere le informazioni in merito al libro da registrare.
  - **Attori coinvolti:** utente.
  - Preconditions:
    * smartphone dotato di connessione dati;
    * l'utente ha effettuato l'accesso alla rete di Book Crossing;
    * il libro possiede il codice ISBN.
  - **Postconditions:** il libro è in possesso dell'utente e non più condiviso con la community.
  - **Processo:**
    1. l'utente seleziona "Registra un nuovo libro" nel menu principale dell'applicazione;;
    2. Viene aperta la fotocamera all'interno dell'applicazione;
    3. l'utente inquadra il codice ISBN finchè il sistema non rileva il barcode.
  - **Alternative**
    * **ISBN non riconsciuto:** il sistema non è in grado di riconoscere l'ISBN inquadrato. Si chiuderà la fotocamera e l'utente verrà reindirizzato alla pagina di inserimento manuale del libro (UC9).
  - **Estensioni**

- ***UC11: Instruction to write BCID code***[3]
  - **Descrizione:** scrittura del codice identificativo sul libro condiviso.
  - **Attori coinvolti:** utente
  - **Preconditions:**
    * smartphone dotato di connessione dati;
    * l'utente ha effettuato l'accesso alla rete di Book Crossing;
  - **Postconditions:** il libro è univocamente riconosciuto del sistema tramite il BCID.
  - **Processo:** l'utente copia il codice BCID sul libro, seguendo le istruzioni mostrate dall'applicazione.
  - **Estensioni**
- ***UC12: View of users contacts***
  - **Descrizione:** l'utente ottiene i contatti che un altro utilizzatore ha deciso di condividere con la community.
  - **Attori coinvolti:** utente.
  - **Preconditions:**
    * smartphone dotato di connessione dati;
    * l'utente ha effettuato l'accesso alla rete di Book Crossing.
  - **Postconditions:** l'applicazione visualizza i contatti a cui l'utente può e/o vuole essere contattato.
  - **Processo:**
    1. l'utente selezione "Ricerca" dal menu principale dell'applicazione;
    2. l'utente seleziona il libro a cui è interessato;
    3. il sistema mostra tutte le informazioni relative al libro (tra cui anche la lista di tutti i lettori che sono stati in possesso del libro in questione);
    4. l'utlizzatore seleziona l'utente che desidera contattare;
    5. il sistema mostra tutte le informazioni di contatto che il cliente terzo ha deciso di condividere con la community.
  - **Alternative:**
    * **Nessun libro trovato:** l'applicazione notifica l'utilizzatore del fatto che la ricerca non sia andata a buon fine.
    * **Utente senza alcun contatto condiviso:** il sistema filtra la visualizzazione della lista degli utenti, visualizzando solo coloro che hanno inserito, durante la fase di registrazione, almeno un contatto o che desiderano essere contattati.
  - **Estensioni**
- ***UC13: Book reservation***
  - **Descrizione:** l'utilizzatore prenota un determinato libro in possesso di un altro lettore.
  - **Attori coinvolti:**
    * utente richiedente (*claimant user*);
    * utente attualmente in possesso del libro richiesto (*owner user*).
  - **Preconditions:**
    * smartphone dotato di connessione dati;
    * l'utente ha effettuato l'accesso alla rete di Book Crossing;
    * il libro che si vuole prenotare deve essere registrato alla rete;

[3] *Book Crossing IDentifier*

    - il libro richiesto deve essere già in possesso di un altro utente.
  - **Postconditions:**
    - Se i due utenti hanno un punto di incontro in comune, si accordano sul luogo di scambio.
    - Se invece non hanno un punto di incontro comune, il libro passerà tra gli utenti che si trovano tra *claimant user* e *owner user*.
  - **Processo:**
    1. l'utente seleziona "Ricerca libro" nel menu principale dell'applicazione;
    2. l'applicazione mostra all'utente un form da completare in cui inserire i parametri della ricerca;
    3. l'utente inserisce i parametri a cui è interessato;
    4. l'utente preme il pulsante "Cerca" ed attende il completamento;
    5. l'utente visualizza la lista di tutti i libri nella rete, che soddisfano la ricerca.
    6. il sistema verifica la presenza del libro cercato;
    7. l'utente va a selezionare il libro all'interno della lista proposta dal sistema;
    8. l'applicazione mostra un riepilogo sulle informazioni del libro, unitamente alla possibilità di prenotare;
    9. l'utente preme il pulsante "Prenota";
    10. l'applicazione mostra una pagina di conferma dell'avvenuta prenotazione.
  - **Alternative**
    - **Libro non prenotabile:** il libro selezionato non è prenotabile poichè non in possesso di un altro utente.
  - **Estensioni**

- ***UC14: Chased books informations***
  - **Descrizione:** visualizzazione storico dei libri "raccolti" dall'utente.
  - **Attori coinvolti:** utente.
  - **Preconditions:**
    - smartphone dotato di connessione dati;
    - l'utente ha effettuato l'accesso alla rete di Book Crossing.
  - **Postconditions:** mostrata sulla grafica la lista dei libri raccolti, con relativa data e luogo di "chasing".
  - **Processo:**
    1. l'utente seleziona "Il mio profilo" dal menù principale dell'applicazione;
    2. l'applicazione mostra le informazioni dell'utente e i diversi stati in cui si possono trovare i suoi libri;
    3. l'utente sceglie lo stato "Libri chased";
    4. l'applicazione mostra un riepilogo di tutti i testi ottenuti dalla community di sharing.
  - **Alternative:**
    - **Nessun libro chased:** l'applicazione mostra un messaggio all'utente, comunicando che nessun libro è stato ancora raccolto da lui dalla community.
  - **Estensioni**

- ***UC15: Released books informations***
  - **Descrizione:** visualizzazione storico dei propri libri condivisi con la rete.
  - **Attori coinvolti:** utente.
  - **Preconditions:**

    - ∗ smartphone dotato di connessione dati;
    - ∗ l'utente ha effettuato l'accesso alla rete di Book Crossing.
  - **Postconditions:** mostrata sulla grafica la lista dei libri inseriti nel programma di sharing, con relativa data e luogo di "relase".
  - **Processo:**
    1. l'utente seleziona "Il mio profilo" dal menù principale dell'applicazione;
    2. l'applicazione mostra le informazioni dell'utente e i diversi stati in cui si possono trovare i suoi libri;
    3. l'utente sceglie lo stato "Libri relased";
    4. l'applicazione mostra un riepilogo di tutti i libri rilasciati alla community di sharing.
  - **Alternative:**
    - ∗ **Nessun libro released:** l'applicazione mostra un messaggio all'utente, comunicando che nessun libro è stato ancora rilasciato da lui nella rete di Book Crossing.
  - **Estensioni**

- ***UC16: "Under reading" books informations***
  - **Descrizione:** visualizzazione dei propri libri attualmente "under reading".
  - **Attori coinvolti:** utente.
  - **Preconditions:**
    - ∗ smartphone dotato di connessione dati;
    - ∗ l'utente ha effettuato l'accesso alla rete di Book Crossing.
  - **Postconditions:** l'applicazione mostra la lista dei libri attualmente in possesso dell'utente
  - **Processo:**
    1. l'utente seleziona "Il mio profilo" dal menù principale dell'applicazione;
    2. l'applicazione mostra le informazioni dell'utente e i diversi stati in cui si possono trovare i suoi libri;
    3. l'utente sceglie lo stato "Libri in possesso";
    4. l'applicazione mostra un riepilogo di tutti i libri attualmente in possesso.
  - **Alternative:**
    - ∗ **Nessun libro in lettura:** l'applicazione mostra un messaggio all'utente, comunicando che nessun libro è in suo possesso al momento.
  - **Estensioni**

- ***UC17: Book relased***
  - **Descrizione:** l'utente libera un libro.
  - **Attori coinvolti:** utente.
  - **Preconditions:**
    - ∗ smartphone dotato di connessione dati;
    - ∗ l'utente ha effettuato l'accesso alla rete di Book Crossing;
    - ∗ libro rilasciato già registrato al sistema di Book Crossing.
  - **Postconditions:** il libro passa dallo stato "under reading" a quello "Available".
  - **Processo:**
    1. l'utente seleziona "Il mio profilo" dal menù principale dell'applicazione;
    2. l'applicazione mostra le informazioni dell'utente e i diversi stati in cui si possono trovare i suoi libri;
    3. l'utente sceglie lo stato "Libri in possesso";

4. l'applicazione mostra un riepilogo di tutti i libri attualmente in possesso;
5. l'utente preme sul testo che intende rilasciare;
6. l'utente conferma il rilascio.
7. l'applicazione mostra una conferma di avvenuto rilascio del testo selezionato.

– **Alternative:**
  * **Nessun libro in lettura:** l'applicazione mostra un messaggio all'utente, comunicando che nessun libro è in suo possesso al momento.
  * **Segnale GPS non trovato:** l'applicazione avvisa l'utente di attivare il GPS del dispositivo.

– **Estensioni**

2.2 Use Case Diagram

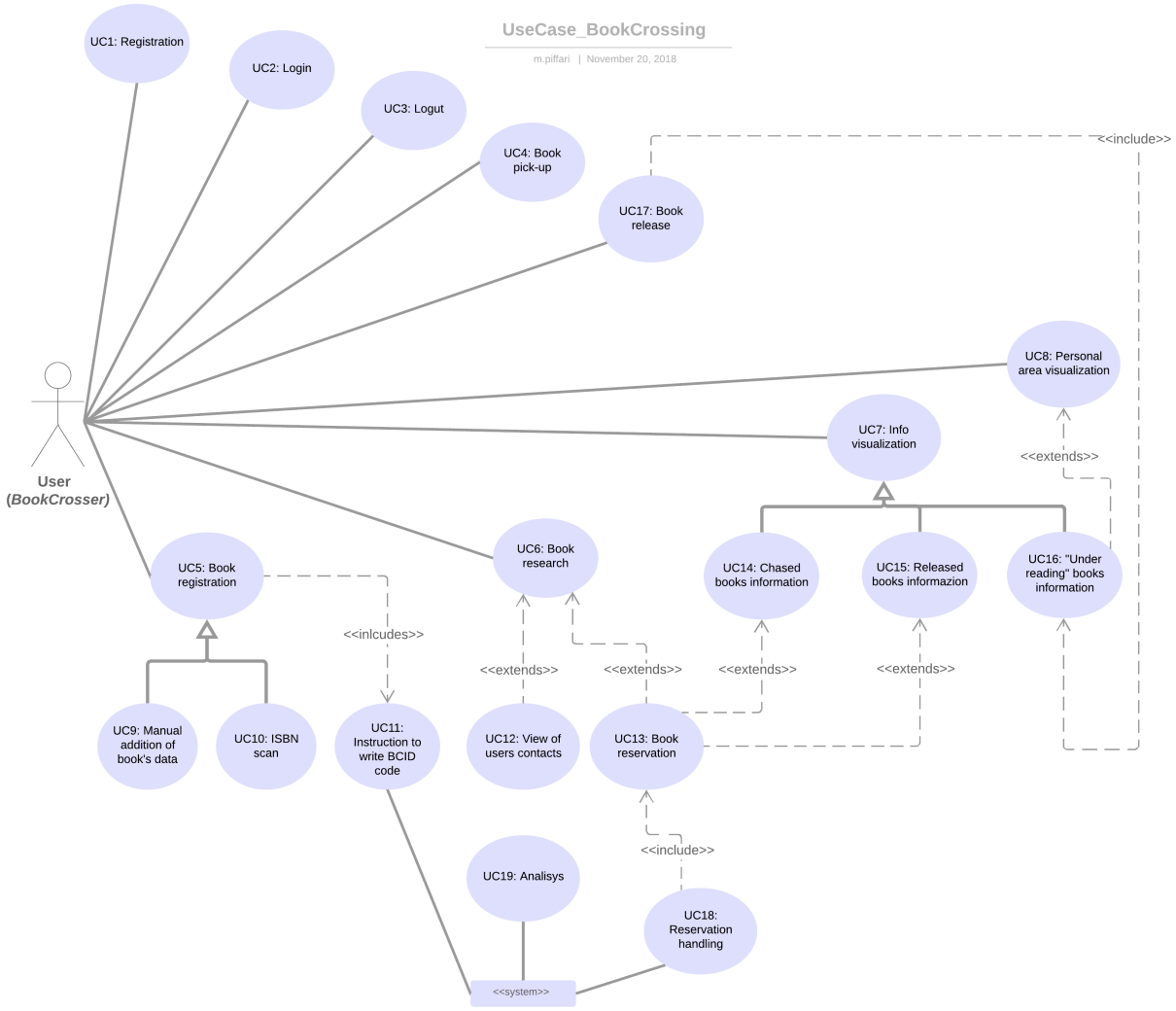

Figure 2.1: Use cases diagram

2.3 Funzionalità richieste

Table 2.1: Panoramica requisiti funzionali progettuali

Nome requisito	ID requisito	Tipologia	Priorità	Requisiti padre	Requisiti figli
Raccolta libro	UR1	funzionale	alta		UR2
Login utente	UR2	funzionale	alta	UR1	UR3, UR4
Registrazione utente	UR3	funzionale	alta	UR2	
Aggiunta libro	UR4	funzionale	alta	UR2	UR7, UR9
Ricerca libro	UR5	funzionale	media	UR2	UR10
Prenotazione libro	UR6	funzionale	bassa	UR2	
Visualizzazione info libri chased	UR7	funzionale	bassa	UR2	
Visualizzazione info libri released	UR8	funzionale	bassa	UR2, UR4	
Rilascio libro	UR9	funzionale	alta	UR2, UR4	
Visualizzazione contatti utenti	UR10	funzionale	bassa	UR2, UR5	
Visualizzazione profilo personale	UR11	funzionale	media	UR2	

Table 2.2: Descrizione requisiti non funzionali progettuali

Nome requisito	ID requisito	Descrizione
Controllo geolocalizzazione	UR12	Requisito non funzionalità che permette di verificare che la posizione GPS salvata del libro corrisponda, con margine d'accettazione, alla posizione in cui si trova l'utente nel momento in cui vuole raccogliere un libro trovato "on the go".

2.4 Stati del libro

La starup si impone l'obbiettivo di andare a gestire lo scambio di libri all'interno della rete di book-crossing: come visto all'interno dei diversi use-cases, ogni libro, nel corso della propria vita all'interno della community, passa di mano in mano, attraversando diverse zone. A questo movimento fisico, corrisponde anche un continuo cambio di stato da parte del libro stesse: possiamo riassumere con una *"Finite State Machine"* il percorso che un generico libro segue durante la sua vita.

INSERIRE PICCOLO SCHEMA DEGLI STATI

Riassumendo gli stati di un libro, possono essere:

- Out of the network
- Available
- Under reading
- Released

- Reserved
- Traveling

3

Architettura

3.1 Deployment diagram

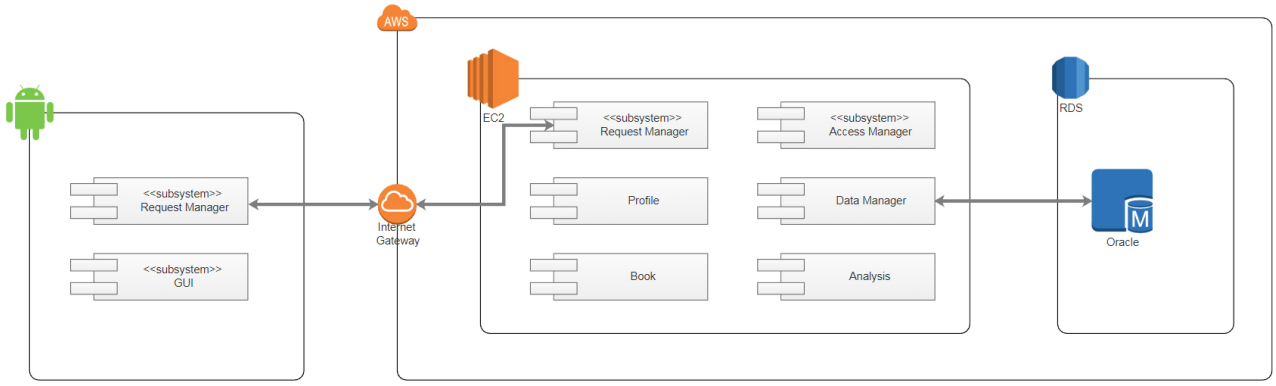

Figure 3.1: Deployment Diagram

3.2 Architecture Envisioning

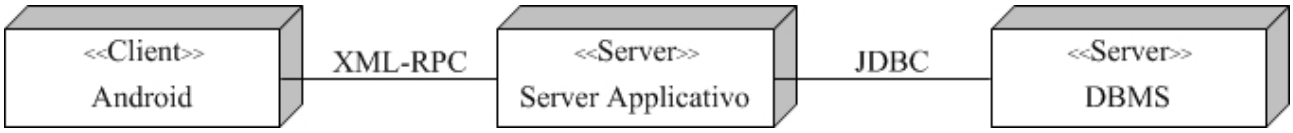

Figure 3.2: Architecture Envisioning

In figura 3.1 e 3.2 sono mostrati il Deployment Diagram e l'Architecture Envisioning del sistema progettato per lo sviluppo dell'applicazione di Book Crossing. Si può osservare che si tratta di un'architettura ***Three Tiers***:

1. A sinistra si individua il client, ovvero il dispositivo Android con il quale è possibile interfacciarsi. Al suo interno quindi si può osservare la presenza di un componente relativo all'interfaccia grafica e uno relativo alla gestione delle richieste per invio e ricezione di dati con il server;

2. Nella parte centrale individuiamo gli altri due layer dell'architettura: server EC2 e Database Relazionale RDS. il fatto di utilizzare Amazon Web Services consente di avere questi due elementi a bordo di un unico strato.

Per quanto riguarda la comunicazione tra i vari layer si sfruttano protocolli e librerie fornite sempre dall'ambiente Amazon Web Services. Nel caso della comunicazione tra smartphone e server EC2 si sfrutta Amazon SDK(nota con link), mentre per interfacciarsi con il Database RDS si sfrutta il connettore JDBC.

Part II

Iterazione 1

4

Architettura

4.1 Architettura software

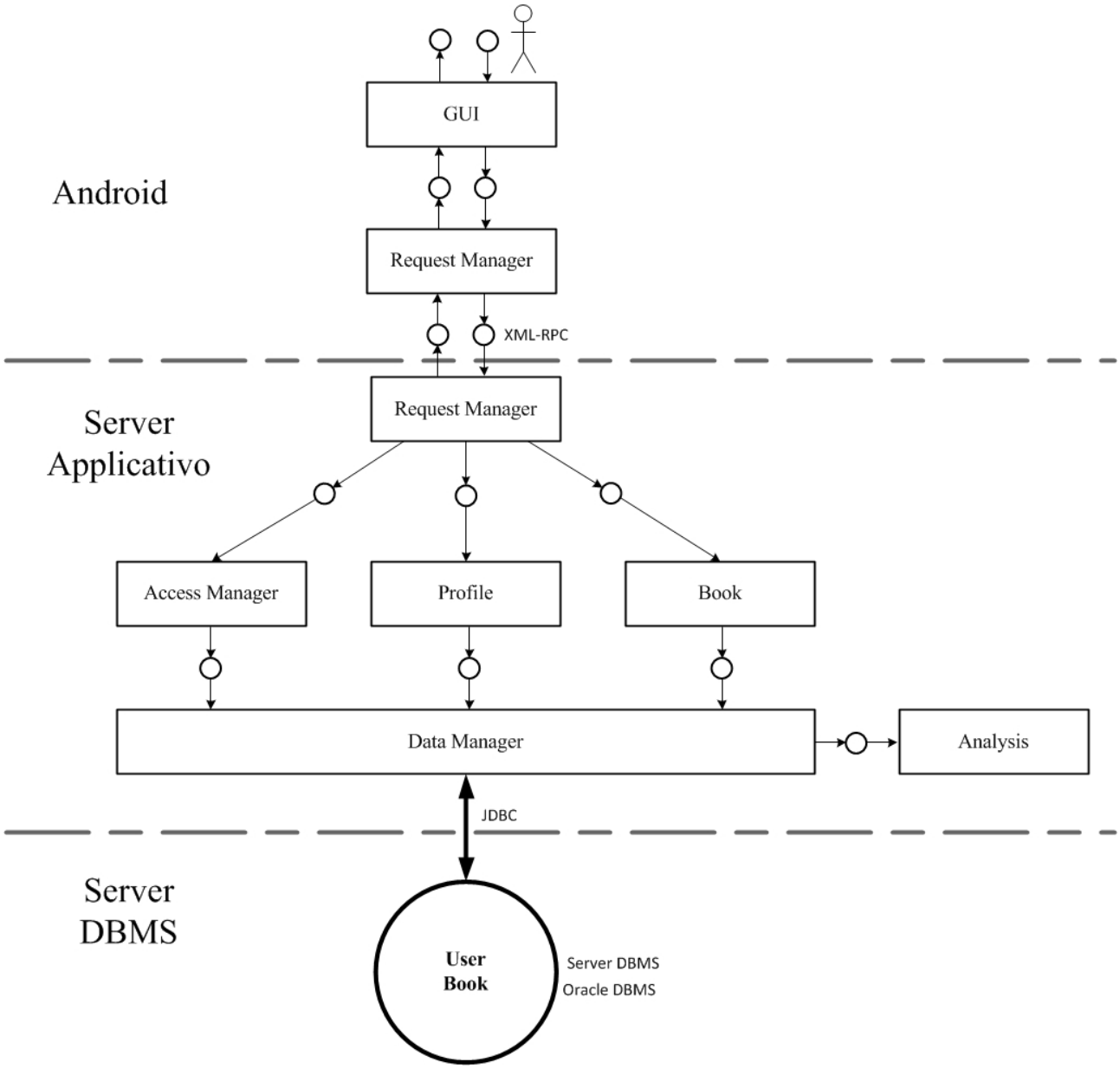

Figure 4.1: Architettura Software

Nella figura 4.1 è mostrata l'architettura software modellizzata attraverso una rete di Petri. Innanzitutto si può già osservare chè è stata definita seguendo il modello architetturale MVC:

- A monte è prevista una parte riservata all'interfaccia grafica, attraverso la quale sarà possibile inviare e ricevere informazioni dal server applicativo. Si vede, infatti, che è prevista una comunicazione bidirezionale tra dispositivo Android e Server.
- Al centro sono rappresentate tutte le richieste a cui è in grado di rispondere. Queste, quindi, saranno funzioni implementate lato Server.
- Infine, è prevista una banca dati persistente, in questo caso un database relazione, al quale il Server Applicativo accede sia per operazioni di lettura che di scrittura, sempre con lo scopo di far fronte alle richieste provenienti a monte.

Si può quindi constatare che non si trattano di strati tra loro indipendenti, poichè il flusso dei dati li coinvolge tutti.

4.2 Logical view

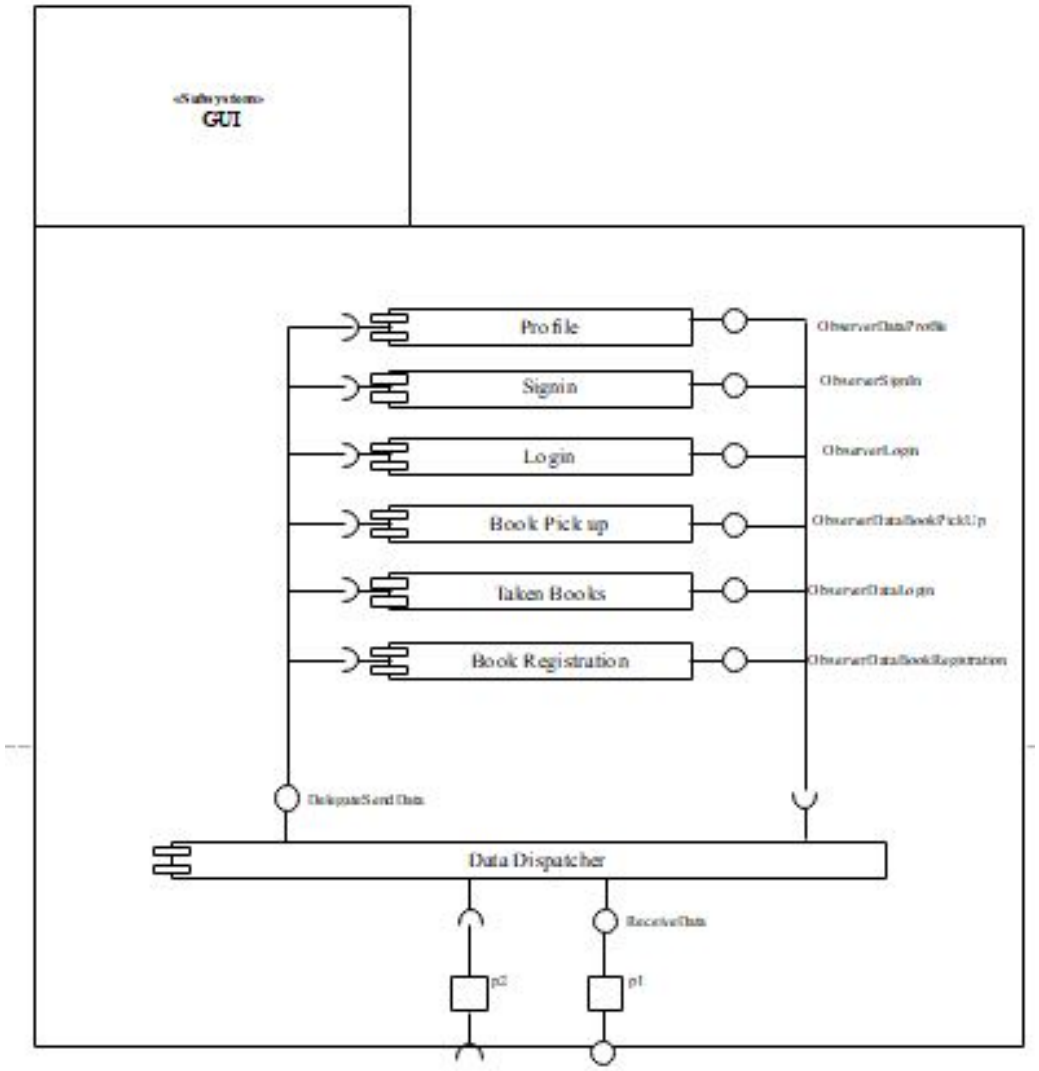

Figure 4.2: Logical view - GUI subsystem

Nelle figure 4.2, 4.3, 4.4 e 4.5 è mostrata in dettaglio la Logical View del sistema progettato. Si può osservare che segue il modello definito attraverso il pattern archietteturale Model View Controller, dal momento che vengono individuati tre strati, ciascuno dei quali con le seguenti caratteristiche:

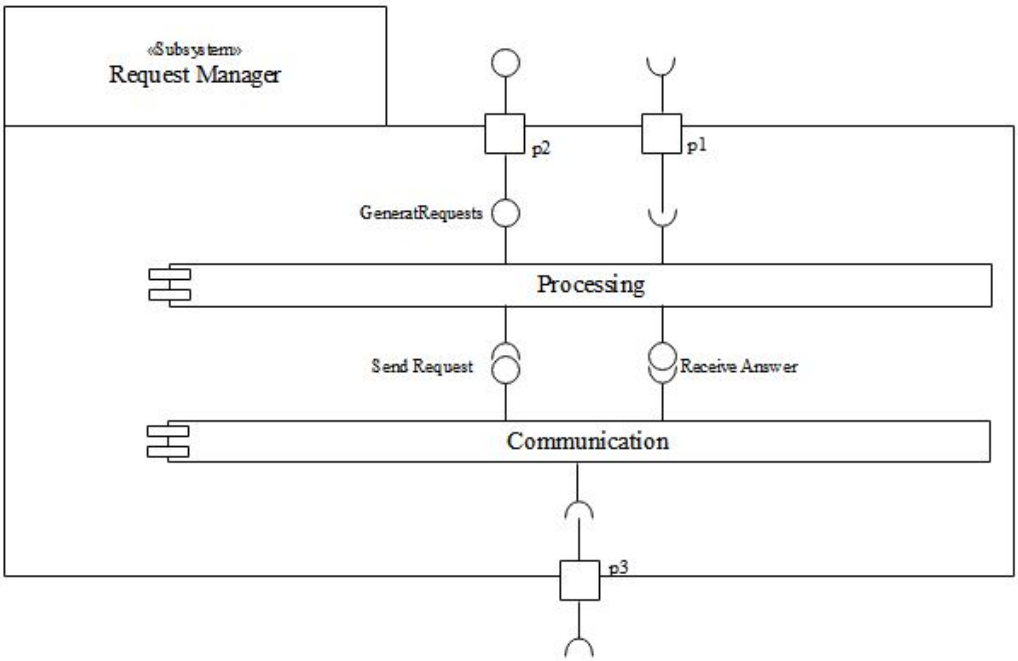

Figure 4.3: Logical view - Request Manager subsystem

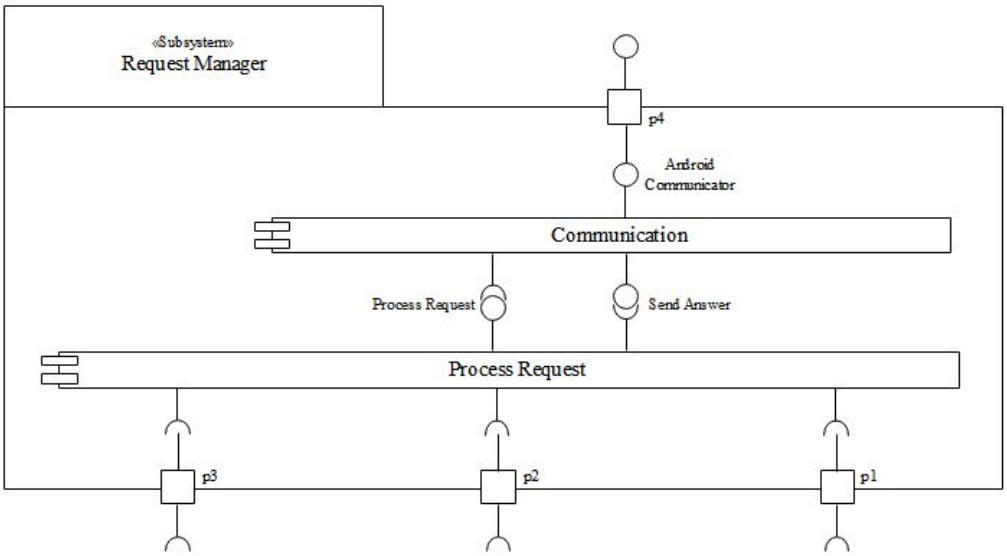

Figure 4.4: Logical view - Request Manager subsystem server side

- **Subsystem *"GUI":*** rappresenta l'interfaccia grafica con la quale l'applicazione si presenterà. Ciascun componente fa riferimento ad ogni view che l'applicazione può mnostrare, e che quindi corrisponde a diffenti casi d'uso dell'applicativo stesso, come l'accesso alla rete di Book Crossing (Login) o registrazione di un libro.

  Questi componenti saranno quindi ovviamente allocati direttamente sul dispositivo mobile: ogni singolo component (*fragment*) avrà legato ad esso, in maniera intrinseca, anche un file *.xml*, il quale permette di definirne l'interfaccia grafica, ovvero come sono posizionati gli elementi visivi.

  TODO: spiegazione sintetica delle funzionalità di ogni singolo fragment

- **Subsystem *"Request Manager":*** ha il compito di gestire le richieste provenienti da ciascun componente descritto nel subsystem *GUI"*. Al suo interno sono indicati i componenti attraverso i quali si risponde alle richieste provenienti dal dispositivo mobile.

  Una parte di questo *manager* sarà disposta a bordo del dispositivo, permettendo una preelabo-

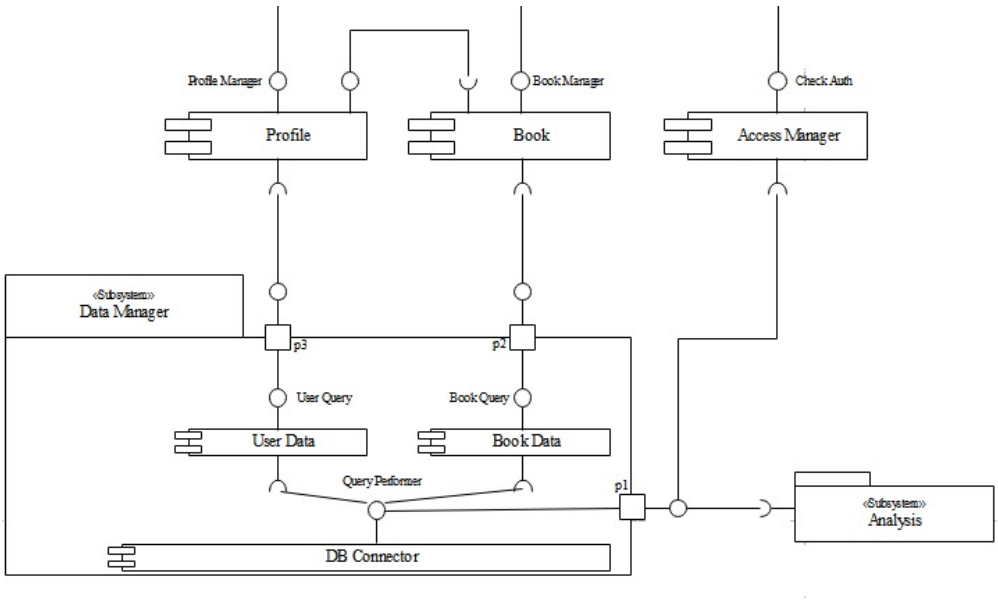

Figure 4.5: Logical view - Server functionality

razione e assemblamento delle richieste; questo subsystem avrà però anche un implementazione *server side*, che gli permetterà di ricevere tali richieste, rieditarla, secondo quello che è la necessità della richiesta, e poi interacciarsi, in un verso o nell'altro, con la parte di persistenza dei dati.

- **Subsystem *"Data Manager":*** ": rappresenta la comunicazione con il Database. Sono quindi indicati i componenti con i quali il sistema si interfaccerà con la banca dati dell'architettura.

Si vede quindi come ogni parte dell'archietettura abbia un compito ben definito: la parte relativa al subsystem *GUI (Graphic User Interface)*, si occupa di gestire l'interazione con l'utente ricevendo e/o mostrando i dati forniti e/o richieste dall'utilizzatore stesso; al suo interno quindi non trovermo codice di logica applicativa ma solamente componenti di gestione *UI*. Esso rappresenta quindi la parte di ***view***. Il subsystem *request manager* invece si occupa di controllare il flusso di dati dall'applicazione al server e viceversa; rappresenta quindi la sezione di ***controller*** dove è contenuta la *low logic* dell'applicazione. Infine, il terzo ed ultimo componente della struttura MVC, è rappresentato dal subsystem *data manager*, il quale permette di interfacciarsi direttamente con la base di dati, astrendo tutte le operazione di controllo di accesso al database stesso. Il modello architetturale MVC è stato poi applicato anche successivamente per la progettazione delle componenti previste per ciascun elemento dell'architettura.

4.3 Analisi dei componenti

Come già presentato in precedenza, per la definizione dei componenti si è deciso di seguire il pattern architetturale MVC. Le componenti che si è deciso di sviluppare durante la prima iterazione sono:

- **Componente *Book registration:*** La componente *Book registration* fa riferimento all' UC5, ovvero alla funzione di aggiunta di un libro alla rete di Book Crossing. La componente si presenta nel seguente modo:
  - *GUI:* Interfaccia grafica utilizzata per registrare un libro alla rete di Book Crossing. Verranno quindi messe a disposizione una serie di funzionalità per aggiungere un nuovo libro e ottenere il relativo BCID;
  - *Model:* Si fa carico di ricevere le informazioni relative al libro e sfruttando la parte Data restituisce alla parte GUI il BCID con il quale siglare il libro;
  - *Data:* Le informazioni relative al libro che si vuole aggiungere sono memorizzate nel Database RDS, associandolo all'utente per ulteriori funzionalità.
- **Componente *Book reservation:*** La componente *Book reservation* fa riferimento all' U13, ovvero alla funzione di prenotazione di un libro, passando prima per un'operazione di ricerca dello stesso all'interno della community. La componente si presenta nel seguente modo:
  - *GUI:* Interfaccia grafica utilizzata per poter procedere con la prenotazione di un libro della rete di Book Crossing. In questo caso sarà possibile compiere tale azione attraverso una procedura di ricerca del libro, oppure navigando nella propria sezione personale del profilo;
  - *Model:* Si fa carico di gestire la coda relativa alla prenotazione di un determinato libro, in modo da poterle soddisfare. La gestione è demandata ad un algoritmo presentato nelle sezioni successive. Fa affidamento alla parte Data per poter risalire ai dati dei richiedenti;
  - *Data:* Le informazioni relative al libro che si vuole prenotare e agli utenti richiedenti sono memorizzate nel Database..

4.4 Observer-delegate pattern: la nostra implementazione

L'applicativo, oltre a seguire un design MVC, per la separazione dei componenti, implementa anche un pattern di comunicazione ad eventi, meglio conosciuto come *pattern observer-delegate*.

TODO: breve spiegazione pattern

Nel nostro applicativo abbiamo introdotto un pattern ad eventi custom, che permettesse così di mantenere separata l'implementazione delle informazioni ottenuto da ogni singolo fragment, come si vede nella figura 4.6. Ogni singola interfaccia implementa un'interfaccia comune: questa scelta è stata

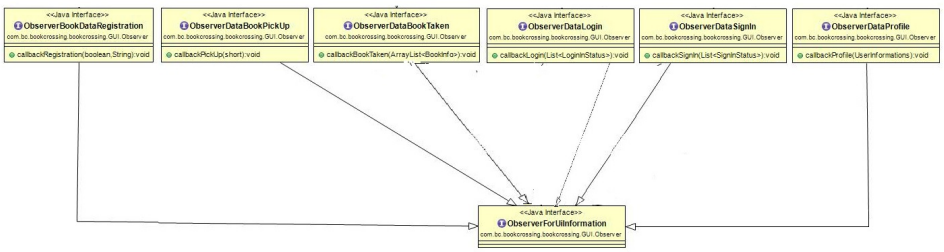

Figure 4.6: Interfaccie per la ricezione degli eventi

adotta per rendere del tutto generico il tipo di observer che va a registrarsi per gli eventi forniti dal delegate: infatti, come si può vedere nell'interfaccia *DelegateSendData* (figura 4.7), chi è interessato ad ottenere informazioni specifiche, si registra come un oggeto di tipo *ObserverForUiInformation*. Sarà poi compito del delegate fornire le corrette informazioni, richiamando le corrette callback.

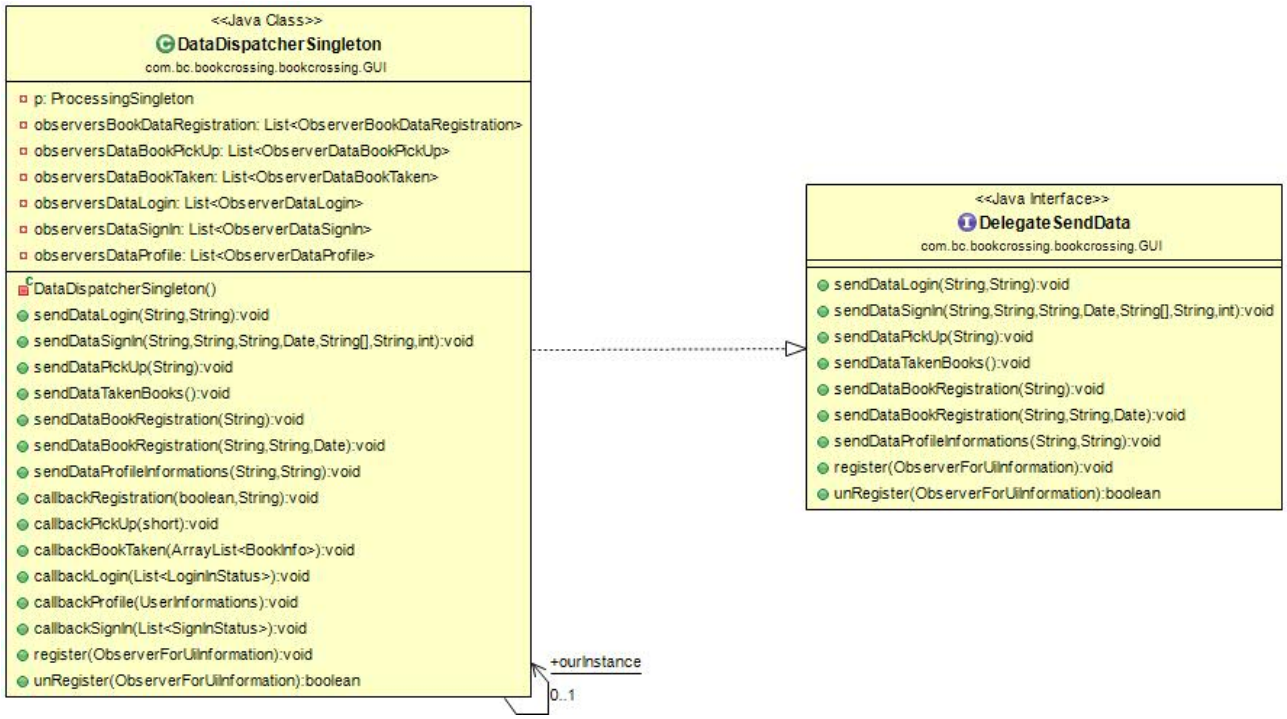

Figure 4.7: Struttura del componente *DataDispatcher*

Si può quindi vedere che, nella struttura predisposta, il *DataDispatcher* lavora come se fosse un **repository**, ovvero un contenitore di informazioni, che è in grado di fornire a tutti gli elementi che si registrano e che ne richiedono la ricezione.

Questi elementi, allo stadio implementativo attuale, sono rappresentati da tutti i fragment, che necessitano di ricevere/inviare informazioni per soddisfare l'interazione con l'utente: come si può notare nelle figure 4.8- 4.8- 4.13- 4.9- 4.10- 4.11- 4.12, ogni singolo fragment è predisposto per ricevere le informazioni di cui ha nisogno, tramite la chiamata delle callback da parte del dispatcher stesso.

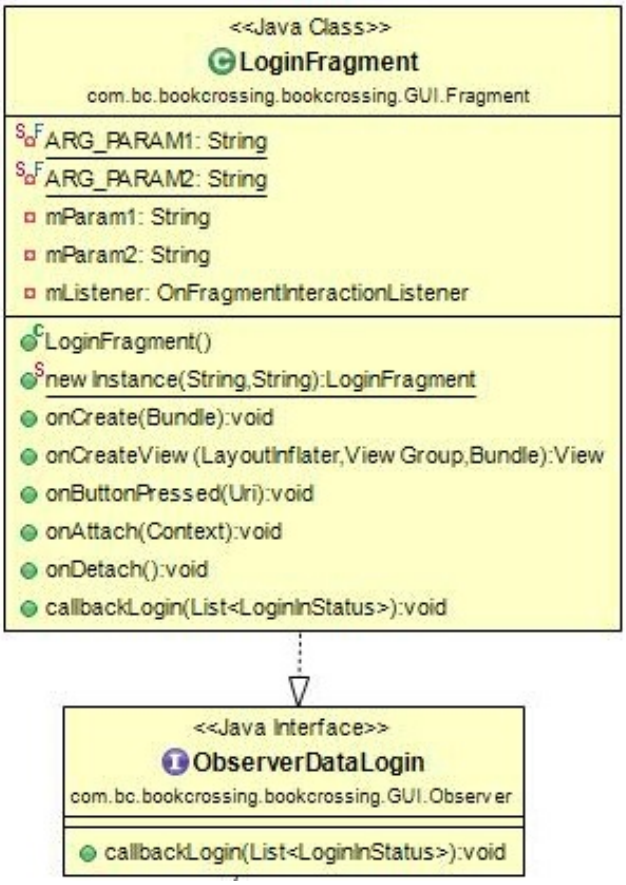

Figure 4.8: Struttura del componente *LoginFragment*

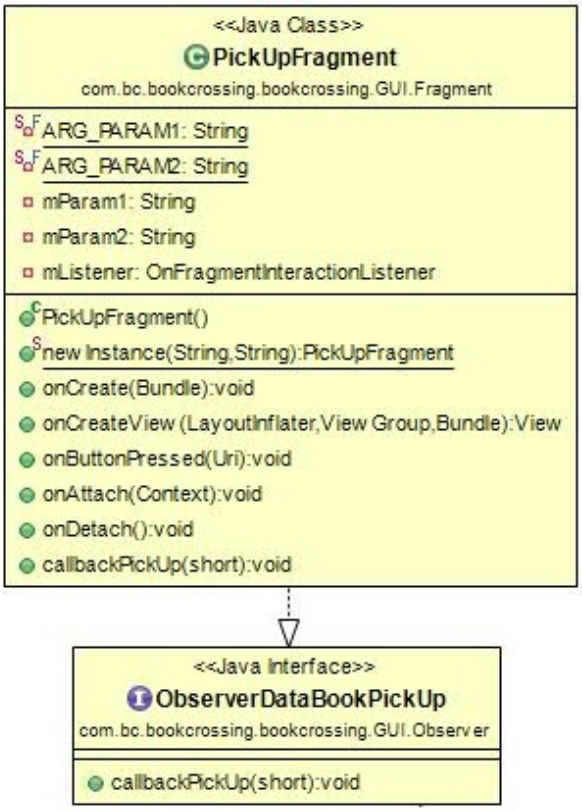

Figure 4.9: Struttura del componente *PickUpFragment*

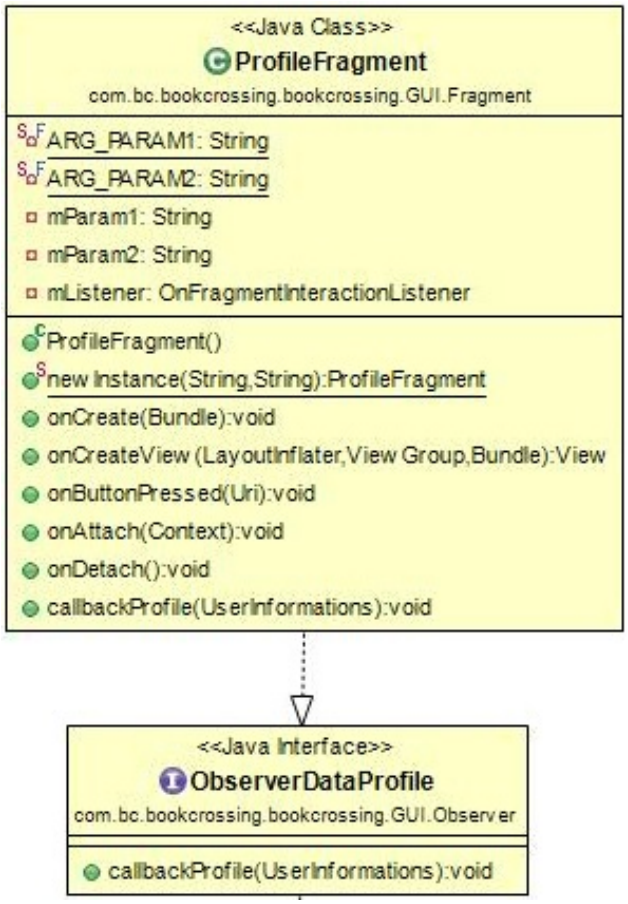

Figure 4.10: Struttura del componente *ProfileFragment*

4.5 Implementazione comunicazione server: il framework *Netty*

4.6 Parte algoritmica: *reservation handler*

La gestione delle prenotazioni dei libri è una delle parte innovative introdotte dalla start up: questo servizio mira a sfruttare la flessibilità della community di sharing, basata sull'ideale di open-source, per comunque offrire un seervizio mirato ed attento alle necessità del lettore. Ogni utente, purchè sia registrato all'interno del servizio di Book-sharing, può prenotare un determinato libro che si trova nello stato "Under reading".

Andiamo ad evidenziare gli attori coinvolti in questa operazione:

- **Lettore [L]:** esso rappresenta l'utente, registrato nella community, che possiede il libro oggetto della prenotazione. Indichiamo con:
  - r_L**:** raggio d'azione del lettore;
  - z_0**:** zona di residenza (espressa come coordinate puntuali).
- **Prenotanti** [P_i **con** $i = 1, ..., N$]**:** rappresentano l'insieme degli utenti, tutti interessati ad uno specifico libro in possesso dell'utente **L**. Nello specifico l'indice *i-esimo* indica l'ordine temporale con cui sono giunte le prenotazioni per lo specifico libro. Oltre a questa informazione, ogni utente, rappresentando un generico utente della community, avrà fornito, al momento della registrazione, le seguenti informazioni:
  - r_i^P **con** $i = 1, ..., N$**:** raggio d'azione del lettore;
  - z_i^P **con** $i = 1, ..., N$**:** zona di residenza.

  L'algoritmo può essere scomposto in due macro-blocchi:

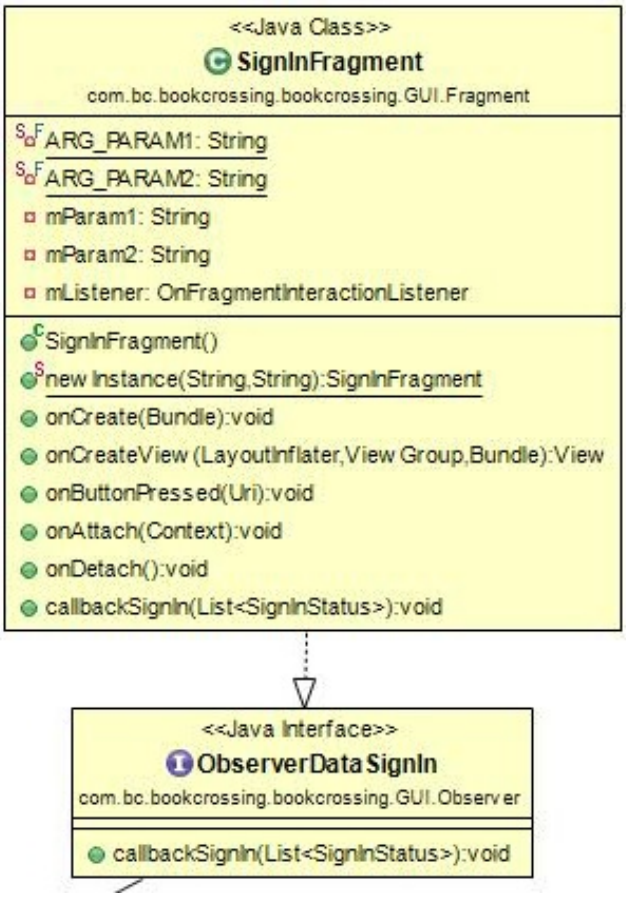

Figure 4.11: Struttura del componente *SignInFragment*

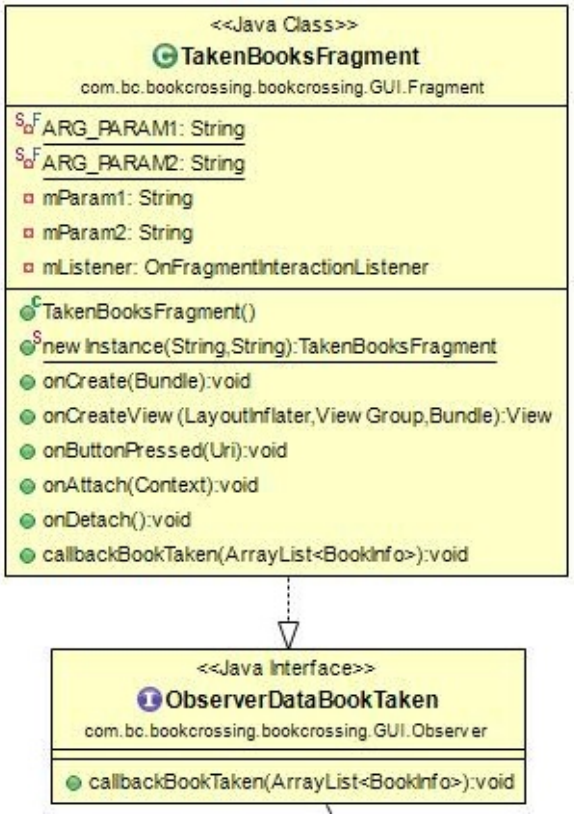

Figure 4.12: Struttura del componente *TakenBooksFragment*

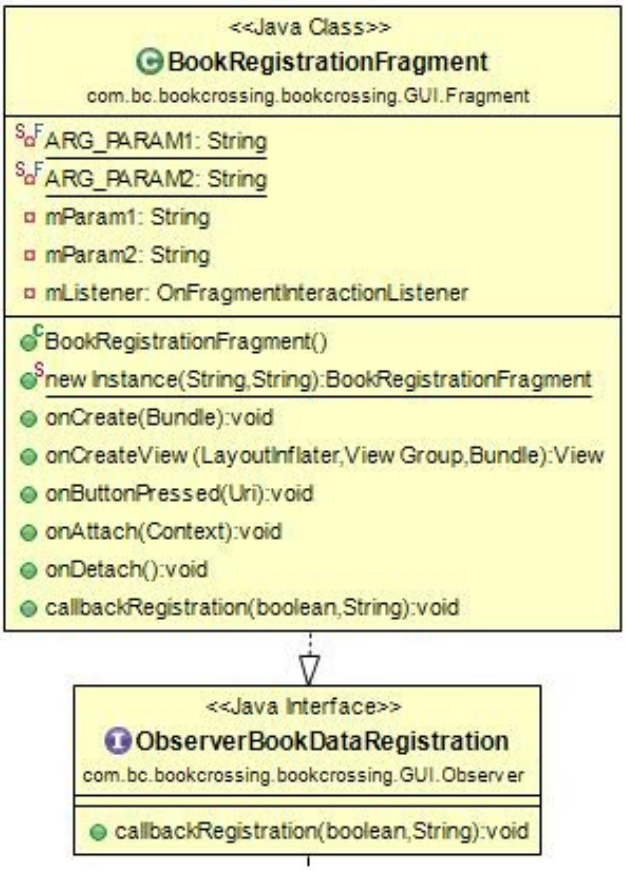

Figure 4.13: Struttura del componente *BookRegistrationFragment*

– **Step 0:** questo fase viene realizzata nel momento in cui il sistema inizia ad analizzare tutti gli utenti che hanno effettuato una prenotazione per un determinato libro che si trova nello stato di "Under reading".

  Tutti gli N prenotanti P_i vengono ordinati in base alla distanza dal lettore L, indipendentemente da quello che è l'ordine temporale con cui è stata effettuata la prenotazione: la quantità di cui si terrà (come si può vedere nella figura 4.14) conto sarà quindi la distanza

$$\left|z_i^p - z_0\right| \tag{4.1}$$

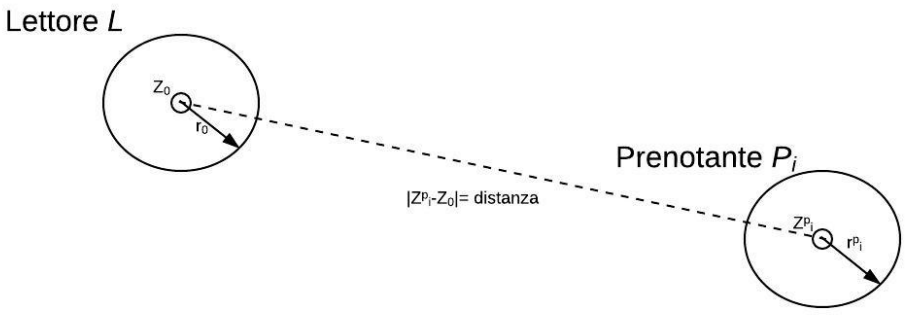

Figure 4.14: Distanza tra lettore e prenotante i-esimo

Questo ordinamento corrisponde quindi sostanzialmente a creare una *priority queue* in cui si va ad assegnare una maggiore priorità all'utente la cui zona di residenza è più vicina a quella del lettore in possesso del libro richiesto.

– **Step 1:** in questo macro-blocco andiamo effettivamente ad applicare l'algoritmo *smart* per poter soddisfare, nella maniera migliore, le esigenze di ogni utente della community.

L'idea di base è che, se lettore e prenotante hanno possibilità di incontrarsi, ovvero se i loro raggi d'azione si sovrappongono, essi potranno accordarsi direttamente sul luogo dello scambio, rendendo *"safety"* il passaggio del libro: questo scambio avverrà ovviamente in una zona all'interno dell'intersezione dei raggi d'azione, come mostrato in figura 4.15.

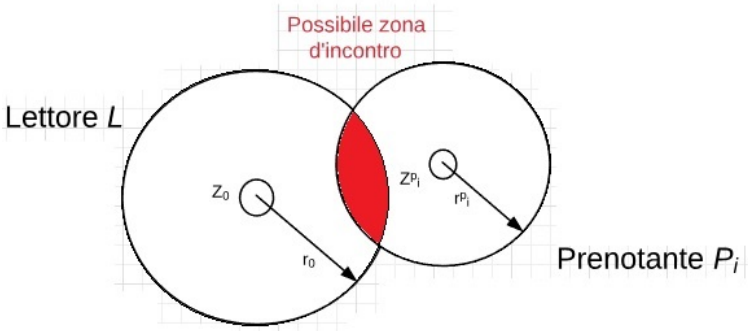

Figure 4.15: Zona d'incontro tra lettore e prenotante

Nel caso in cui invece, i due utenti interessati non abbiano la possibilità di trovare un luogo comune in cui potersi scambiare il libro direttamente, si avrà che, la rete di users appartenenti alla community farà da tramite, per portare il libro *"coast-to-coast"*. Quindi, tramite un semplice pseudo-codice, possiamo scrivere il nostro algoritmo come

Algorithm 1: Algoritmo di gestione della prenotazione

```
Data: Users and their informations
Result: Optimum path from reader to reserver
Step 0 (intilization);
if Distanza <= 0 then
    Trova un punto d'incontro nell'unione delle delle area;
    Notifica gli utenti di dove potersi scambiare direttamente il libro;
else
    Crea la rete di utenti che faranno da tramite tra lettore e prenotante;
    Ricercare il cammino ottimo (il libro si muoverà hand-to-hand);
end
```

Nello specifico il calcolo della distanza avverrà tramite la funzionalità *VerificaPuntoIncontro(Lettore, Prenotante)* (TODO: inseire funzione effettiva), la quale andrà a verificare che:

$$|z_i^p - z_0| - r_0 - r_i <= 0 \tag{4.2}$$

ovvero che i raggi d'azione si sovrappongano o meno.

Nel caso in cui i due utenti non abbiano possibili punti d'incontro (Distanza >= 0), dobbiamo selezionare gli utenti tramite il quale il libro in questione potrà viaggiare: l'idea base è quindi quella di costruirsi un'area circolare di centro pari alla metà della congiungente del punto z_0 (zona di residenza lettore) e z_i^p (zona di residenza prenotante). Andremo poi a selezionare tutti gli utenti che si trovano all'interno di questa circoscrizione. In passi

sequenziali, possiamo scrivere:

Algorithm 2: Creazione del percorso tra lettore e prenotante

Data: Zona di residenza e raggio d'azione di utente lettore e prenotante
Result: Elenco di utenti attraverso cui il libro dovrà spostarsi *hand-to-hand*
Il raggio della circoscrizione di utenti coinvolti sarà pari alla distanza

$$\bar{Z} = \frac{1}{2}|z_i^p - z_0|$$

for *Tutti gli utwnti* z_i^U *nella community* **do**
  if $(Distanza(z_0, z_i^U)) <= \bar{Z}$ *oppure* $(Distanza(z_i^P, z_i^U)) <= \bar{Z}$ **then**
    Seleziono l'utente z_i^U e lo inserisco nella lista (*HandToHandUsers*) dei possibili utenti che potrebbero partecipare attivamente al prestito;
  end
end
Creiamo il collegamento tra gli users della lista *HandToHandUsers* il cui raggio d'azione si sovrappone;

Quindi, alla fine dello step 1, avremo individuato tutti gli utenti i quali potrebbero partecipare attivamente alla realizzazione di un prestito: il risultato ottenuto sarà quindi come quello in figura 4.16.

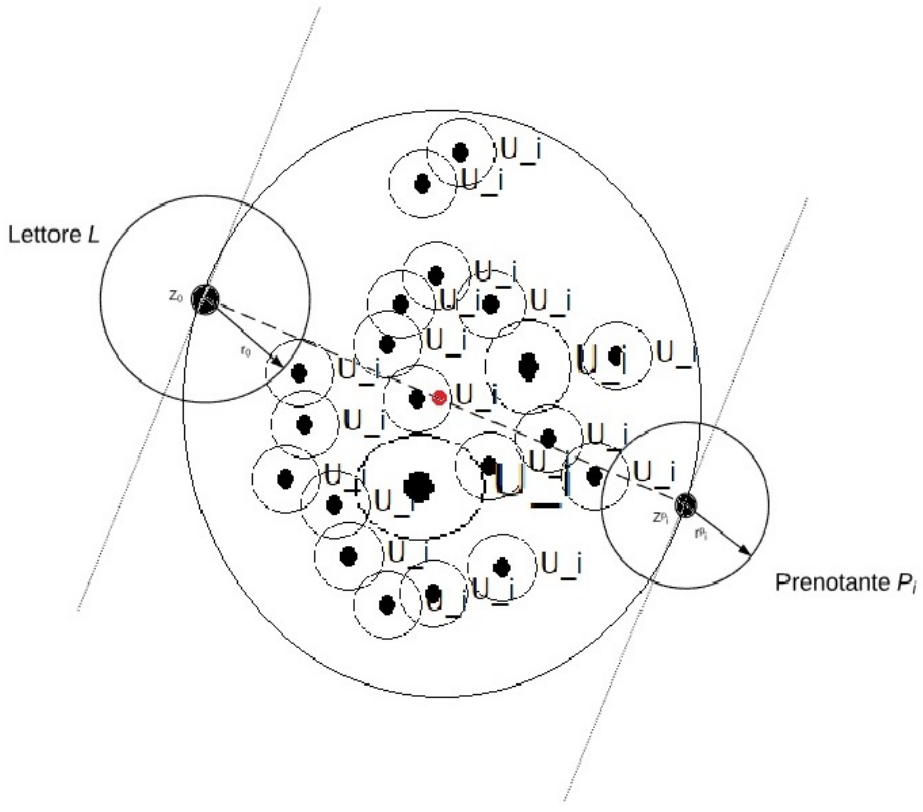

Figure 4.16: Rete di utenti che potrebbero essere attivi nel prestito

//TODO: move this comment to documentation

// Insieme di candidati: users == overlapping users // Funzione obbiettivo: minimizzare i km che il libro deve fare / minimizzare il numero di utenti tra cui fare // il passamano –> la scelgo in maniera epsilon - greedy

// Epsilon for epsilon greedy algorithm: during the path calculation for the reservation // we

choose usually the next user, as the nearest one. // Using the greedy algorithm, with a probability that depends on the epsilon variable value, we choose the user // with the biggest radius. // epsilon lower –> algorithm stronger (always foound path to me) // epsilon higher –> algorithm found sorthest path