



**THOMPSON RIVERS UNIVERSITY**

SENG 3210– Applied Software Engineering

Books R' Us

# Deliverable 1 – Book Recommender System

Sarah Hill (T00731037)

Ivailo Stoychev (T00728133)

Natasha (T00714769)

February 15, 2026

# Table of Contents

<b>Table of Contents.....</b>	<b>2</b>
<b>List of Figures.....</b>	<b>2</b>
<b>List of Tables.....</b>	<b>2</b>
<b>1. Introduction.....</b>	<b>3</b>
<b>2. Design Problem.....</b>	<b>3</b>
2.1. Problem Definition.....	3
2.2. Design Requirements.....	3
2.2.1. Objectives.....	3
2.2.2. Functional Requirements.....	4
2.2.3. Non-Functional Requirements.....	4
2.2.4. Constraints.....	5
<b>3. Solution.....</b>	<b>5</b>
3.1. Solution 1.....	5
3.2. Solution 2.....	5
3.3. Final Solution.....	5
3.3.1. Features and the software architecture.....	6
3.3.2. The system interfaces.....	6
3.3.3. The user interface design.....	6
3.3.4. The requirements traceability matrix.....	6
3.3.5. Environmental, Societal, Safety, and Economic Considerations.....	6
<b>4. Teamwork.....</b>	<b>7</b>
4.1. Meeting 1.....	7
4.2. Meeting 2.....	7
4.3. Meeting 3.....	7
4.4. Meeting 4.....	7
<b>5. Conclusion and Future Work.....</b>	<b>8</b>
<b>6. References.....</b>	<b>8</b>
<b>7. Appendix.....</b>	<b>8</b>

## List of Figures

## List of Tables

# **1. Introduction**

In Thompson Rivers University, one of the key aspects of attending such a campus is its ability for you to connect more personally with your instructors. It's also a campus that prides itself in its arts and literature-based classes. To bridge these facets, Books R' Us is intended to be a dedicated book searching, recommendation-based platform for the many classes in TRU to utilize in order to facilitate a more collaborative, unique learning journey by letting students choose what books to read in their classes, and review books that they have read as well.

For a universal reach to all students and faculty, Books R' Us is a mobile application that you can install onto any modern smartphone seamlessly, utilizing a very intuitive and easy-to-use interface that enables all students to be able to voice their opinions and recommend books to their instructors straight from their phones. It invites a more personalized experience on campus, incentivizes more interaction between all members, and provides a fulfilling educational journey for everyone.

## **2. Design Problem**

### **2.1. Problem Definition**

Books R' Us is organizing a Summer Study Challenge that allows students to propose books for group reading. Due to the wide range of opinions among participants, selecting a suitable book has become challenging. To address this issue, Books R' Us plans to design a mobile application called ReadForFun to recommend books. ReadForFun will allow students to rate, recommend, and search for books remotely, while administrators will be able to add, edit, and delete books from the database.

### **2.2. Design Requirements**

#### **2.2.1. Objectives**

- Provide an efficient platform for students to search, rate, and discover books with minimal delay.
- Offer a user-friendly and intuitive interface that allows users to easily navigate and perform tasks without technical difficulty.

- Deliver complete functionality for both students and administrators, including book searching, rating, recommendations, and database management.
- Ensure reliable and secure handling of user data, ratings, and administrative actions.

## 2.2.2. Functional Requirements

### 2.2.2.1. Book Search & Viewing

- The system shall allow users to search books by title.
- The system shall display book details (title, author, description, average rating).
- The system shall display a list of recommended books.

### 2.2.2.2. Rating & Recommendation

- The system shall allow users to rate books on a predefined scale (1–5).
- The system shall store user ratings in the database.
- The system shall calculate the average rating of each book.
- The system shall recommend books based on the highest calculated average ratings.

### 2.2.2.3. Administrator Controls

- The system shall allow administrators to add new books.
- The system shall allow administrators to edit book details.
- The system shall allow administrators to delete books from the database.

## 2.2.3. Non-Functional Requirements

### 2.2.3.1. Performance

- The system shall load search results within 3 seconds.
- The system shall support at least 100 concurrent users without degradation.

### 2.2.3.2. Security & Data Integrity

- The system shall implement authentication to protect user accounts.

- The system shall use security rules to prevent unauthorized modification.
- Users shall only be able to modify their own ratings.
- Only administrators shall have write access to book records.

#### 2.2.3.3. Usability

- The application shall provide a simple and intuitive user interface.
- Error messages shall be clear and informative.

#### 2.2.3.4. Modifiability

- The UI components shall be modular to allow easy expansion.
- The database schema shall allow adding new attributes.

#### 2.2.3.5. Compatibility

- The system shall support Android API level 19 (KitKat) and above.

#### 2.2.3.6. Reliability

- The system shall not crash due to null or invalid inputs.

#### 2.2.4. Constraints

- Must be created using Android Studio.
- Must support Android API Level 19 (KitKat).
- Must be completed by March 30th.

## 3. Solution

### 3.1. Solution 1

Books are ranked based on the highest calculated average rating (1-5). The top 10 list is generated from books with the highest averages.

Advantages included:

- Low implementation complexity
- Would work efficiently with real-time Firebase updates.
- Transparent and user-friendly.
- Minimal database queries because only average values need to be retrieved and sorted.

Disadvantages:

- Books with only 1 rating could appear unfairly low or not evenly rated if high.
- It does not account for rating count reliability.
- Small changes in rating would dramatically change the top 10 list.

## 3.2. Solution 2

Books are ranked according to average rating, but they have a minimum number of ratings, eg. 2, before being eligible for the Top 10 dashboard.

Advantages:

- Reduces bias from single ratings.
- Still simple to implement, since it would only require tracking the rating count as well as the average.
- Maintains real-time compatibility and user reliability.

Disadvantages:

- Slightly increased logic complexity.
- Still not fully personalized since rankings are global instead of tailored to each user.
- Delayed visibility for new books since it may take time for them to appear on the top 10 list.

## 3.3. Solution 3

Books are recommended based on similarity between users' rating behaviors (e.g., collaborative filtering).

Advantages:

- Higher personalization since users would receive recommendations aligned with their interests.
- Modern user experience since personalized recommendations are comparable to platforms like Netflix or Pinterest.

Disadvantages:

- High implementation complexity since it requires more advanced data processing.
- Longer development timeline.
- Increased back-end processing may affect performance.

## **3.4. Final Solution**

3.4.1. Features and the software architecture

3.4.2. The system interfaces

3.4.3. The user interface design

3.4.4. The requirements traceability matrix

3.4.5. Environmental, Societal, Safety, and Economic Considerations

3.4.5.1. Environmental considerations

3.4.5.2. Societal considerations

3.4.5.3. Safety considerations

3.4.5.4. Economic considerations

3.4.5.5. Limitations

## 4. Teamwork

### 4.1. Meeting 1

**Time:** February 11, 2026, 12:00 pm

**Agenda:** Distribution of Project Tasks

Team Member	Previous Task	Completion State	Next Task
Sarah	N/A	N/A	Requirements
Ivailo	N/A	N/A	Introduction
Natasha	N/A	N/A	Solutions

### 4.2. Meeting 2

**Time:** February 11, 2026, 12:00 pm

**Agenda:** Distribution of Project Tasks

Team Member	Previous Task	Completion State	Next Task
Sarah	N/A	N/A	
Ivailo	N/A	N/A	
Natasha	N/A	N/A	

### 4.3. Meeting 3

**Time:** February 11, 2026, 12:00 pm

**Agenda:** Distribution of Project Tasks

Team Member	Previous Task	Completion State	Next Task
Sarah	N/A	N/A	
Ivailo	N/A	N/A	
Natasha	N/A	N/A	

### 4.4. Meeting 4

**Time:** February 11, 2026, 12:00 pm

**Agenda:** Distribution of Project Tasks

Team Member	Previous Task	Completion State	Next Task
Sarah	N/A	N/A	



Ivailo	N/A	N/A	
Natasha	N/A	N/A	

## **5. Conclusion and Future Work**

## **6. References**

## **7. Appendix**