

# REQUIREMENTS - GROUP 17C

## ***BOOKMANIA USER MICROSERVICE***

### **Contents**

1.	<i>Terminology</i>	1
2.	<i>Functional requirements</i>	1
2.1.	Must Haves	1
2.2.	Should Haves	2
2.3.	Could Haves	2
2.4.	Won't Haves	2
3.	<i>Non-functional requirements</i>	2

### **Terminology**

1. **Regular User**: they can add books to their bookshelf, follow other users, write book reviews
2. **Admin**: they have multiple responsibilities, including adding, editing, and removing books from the overall collections
3. **Author**: they can add their own books, authors are a special type of user
4. **System**: the user microservice

### **Functional requirements**

#### **Must Haves**

- A client must be able to create a user account
- A user must contain a username, a unique email address and a password.
- A user must be able to manage their profile (name, bio, profile picture, location, favourite book genres, favourite book)
- Admins must be able to add, edit and remove books from the overall collection.
- Admins must be able to remove, deactivate and add users.
- A user shall not be able to edit books or other users' information/account
- An author must be able to prove their author identity
- An author must be able to add their own books to the system

## Should Haves

- A user should be able to edit privacy preferences.
- A user should be able to edit notification settings.
- A user should be able to modify account security settings.
- A user should be able to add books(that already exist in the system) to their bookshelf
- A user should be able to write book reviews.
- A user should be able to discover other users based on name, interests, other books.
- The system should implement features to deactivate or delete user accounts upon request, ensuring compliance with privacy regulations and user preferences.

## Could Haves

- A user should be able to discover other users based on connections.
- A user can follow other users/authors
- A user can unfollow other users/authors
- The system can collect and provide analytics on user engagement, login activity, popular genres, and other relevant metrics
- The system can improve user experience and platform performance using analytics.

## Won't Haves

- The system will currently not have a two-factor authentication option.
- The system will currently not display notifications.

## Non-Functional requirements

- The system shall be modular, adding functionality should be easy
- The system shall be easily integrated with other systems
- The system shall be developed using Java 15
- The system shall be developed using Spring Boot
- The builds shall be realized using Gradle
- A user shall not be allowed to use the same email address as another user
- A username shall start with an alphabetic character and not include special characters (e.g., punctuation marks, \$, /, -, \_, etc.)