以下是对该图书管理系统的需求分析：

\*\*一、功能需求\*\*：

- \*\*图书添加功能\*\*：

- 目的：允许用户向图书馆系统中添加图书信息。

- 实现：

- 调用 `addBook` 方法时，程序会使用 `Scanner` 类从控制台接收用户输入。

- 提示用户输入图书的名称和作者信息。

- 接收用户输入的图书名称和作者，创建 `Book` 类的对象存储这些信息。

- 将创建的 `Book` 对象添加到 `ArrayList<Book>` 集合 `books` 中。

\*\*二、数据存储需求\*\*：

- \*\*图书信息存储\*\*：

- 目的：存储图书的基本信息。

- 实现：

- 定义 `Book` 类，包含 `name` 和 `author` 属性，用于存储图书的名称和作者。

- 使用 `ArrayList<Book>` 集合 `books` 作为存储图书信息的数据结构，存储多个 `Book` 对象。

\*\*三、用户交互需求\*\*：

- \*\*信息输入提示\*\*：

- 目的：引导用户输入所需信息。

- 实现：

- 在 `addBook` 方法中，通过 `System.out.println` 输出提示信息，提示用户输入图书名称和作者信息。

\*\*四、可扩展性需求\*\*：

- \*\*功能扩展\*\*：

- 目的：为系统后续功能的添加预留空间。

- 实现：

- 目前仅提供了添加图书的功能，后续可添加删除图书、修改图书信息、查询图书、借阅图书、归还图书等功能。

\*\*五、性能需求\*\*：

- \*\*基本性能保障\*\*：

- 目的：在小规模使用场景下保证系统性能。

- 实现：

- 对于少量图书的添加和存储，使用 `ArrayList<Book>` 可以满足性能需求，但对于大规模图书管理，可能需要考虑更高效的数据存储结构。

\*\*六、可靠性需求\*\*：

- \*\*输入验证\*\*：

- 目的：确保输入信息的有效性和可靠性。

- 实现：

- 当前代码未对用户输入进行验证，后续可添加对图书名称和作者信息的长度、格式等验证，避免存储不合法或不合理的数据。

\*\*七、可维护性需求\*\*：

- \*\*代码结构优化\*\*：

- 目的：提高代码的可读性和可维护性。

- 实现：

- 可以添加更多注释，解释 `Book` 类和 `LibraryManagementSystem` 类的功能。

- 将功能模块细化，例如将输入输出、数据处理、业务逻辑等部分拆分为不同的方法或类，提高代码的模块化程度。

\*\*八、异常处理需求\*\*：

- \*\*异常处理\*\*：

- 目的：处理系统运行中可能出现的异常情况。

- 实现：

- 当前代码未对可能出现的异常进行处理，如 `Scanner` 输入异常、集合操作异常等，后续可添加相应的异常处理代码，确保系统在异常情况下的稳定性。

\*\*九、安全性需求\*\*：

- \*\*数据安全保障\*\*：

- 目的：确保系统数据的安全性。

- 实现：

- 对于用户输入的数据，目前没有安全防护措施，在后续扩展（如存储到数据库）时，需考虑防止 SQL 注入等安全问题，对用户输入进行安全处理。