Bookflex

Documentation

Prepared:

Christian Huber K12010223Tymofii Bereznytskyi K123480

Table of Contents

[1. Goal of system 3](#_Toc169616391)

[2. Requirements 5](#_Toc169616392)

[a) Functional requirements 5](#_Toc169616393)

[b) Non-functional requirements 6](#_Toc169616394)

[c) AI-related requirements 7](#_Toc169616395)

[3. Use case descriptions 9](#_Toc169616396)

[a) Main use cases 9](#_Toc169616397)

[b) Supporting use cases 14](#_Toc169616398)

[4. Use case diagram 19](#_Toc169616399)

[5. Traceability matrix 20](#_Toc169616400)

[6. Domain model 21](#_Toc169616401)

[7. System architecture 22](#_Toc169616402)

[a) Diagram 22](#_Toc169616403)

[b) Description 22](#_Toc169616404)

[8. Design questions 25](#_Toc169616405)

# Goal of system

Main goal is book recommendation basing on previous read books and user’s mood. It is done by analyzing user’s preferences and books which were previously read by user.

**Part - I. Goals of the system**

**1) Non-AI related**

a) An application for the books search

b) Books review (readers feedback)

**2) AI-related**

a) Books recommendation basing on previous read books

b) Collecting and analyzing data about users` mood

c) Organizing books into subcategories which could be interesting for the user

**Part - II. Stakeholders**

**1) Users**

a) Heavy readers

b) People who are looking for new books

c) Authors

**2) People affected by the system**

a) Book stores

b) Authors

c) Publishers

**3) Managers**

a) Course instructors

b) Christian Huber

c) Tymofii Bereznytskyi

**4) Regulators**

a) Federation of European Publishers

# Requirements

## Functional requirements

1. When the user authenticates, Booklib shall display the welcome screen.
2. Welcome screen shall contain questionary about the mood of the user.
3. If a user answered the mood questionary, Booklib shall consider the current mood of the user for recommendation.
4. If user did not answer the mood questionary, Booklib shall not consider the current mood of the user for recommendation.
5. If the user is already authenticated, Booklib displays home screen.
6. Home screen shall display a navigation bar and recommendations from different categories.
7. Navigation bar shall contain a search bar, home button and profile button.
8. When the user enters a search term and clicks search button, Booklib shall process the request and find the most suitable recommendation.
9. When user scrolls down on the book page, Booklib shall display book recommendation according to the familiarities with the current book and previous experience.
10. If Booklib considered user`s mood, home screen shall contain additional recommendation category on the home screen based on the mood.
11. When user enters the home screen for the first time, recommendations shall be computed.
12. If a book is recommended based on the user`s mood, book preview picture shall contain additional mood label in the left top corner.
13. Recommendation system shall consider user`s reviews for each book
14. Book page shall display the user's review section at the bottom.

## Non-functional requirements

* 1. External interfaces
     1. The Booklib shall be accessible from any browser.
     2. Application shall be deployed on cloud in the container.
  2. Performance
     1. Recommendations’ computation shall take less than 2 seconds.
     2. The Booklib shall not be down more than 1 day per year in sum and not more than 30 minutes per week.
     3. If the system crashes, Booklib shall restart within 5 minutes.
  3. Attributes
     1. Only internal system shall have direct access to the database with user data
     2. Booklib shall validate email address
     3. 80% of recommendation shall be in the user’s field of interests
     4. Each recommendation computing shall give 25 the most suitable books
  4. Design constraints
     1. Database shall have encrypted backup copy
     2. Booklib shall be compatible with Linux platform
     3. AI model shall be implemented in the python language.
     4. The database shall only be change by the maintainer
     5. The user shall have the possibility to give a review about the recommendations.

## AI-related requirements

In general, we will use something like a K nearest Neighborhood algorithm to solve that problem. The recommendation should give back the 25 closest books. We calculate the distance to all our books in the database and give the best 25 back. It will be like KNN but without the classification. We look later deeper into it, there are more solutions to solve that problem, but we want something like Collaborative filtering (CF).

Req3, Req4 - The current mood is an extra feature (categorical), which we consider or not for the recommendation.

Req8 - results which have the closest attributes with search term, the user preferences should also flow into the search. It should return the request, but also recommendations that are correlated to the search and the preferences.

Req13 - The model considers the users reviews as an extra feature which is a quantitative measure that ranges from 1 to 5.

NfReq3 - Because of the loading of all media files, it is possible to allow some latency for recommendation but users should not wait long for the results. When the computation takes longer than 2 Seconds it should be aborted and return the current best recommendations.

NfReq6 - (privacy category) - user`s data should not be exposed to anyone besides the system in order to protect private information.

NfReq8 - There could be a problem when the system aborts, because it can’t consider all the data. Additionally, accuracy should show results which are not so close to the user's preferences in order to explore new types of books.

NfReq12 - python is the most popular language for models which has a lot of available and prepared libraries which help to speed up the development phase.

NfReq13 (Security and Privacy - Integrity) - The only authorized persons can change the database.

NfReq14 - (Safety- reducing mistakes) ensures that the recommendations satisfy the user. So, if he isn’t satisfied with some recommendations or wrong recommendations then he can click on those and specify that.

# Use case descriptions

## Main use cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case: System login** | | | |
| **ID** | | UC1 |  |
| **Description** | | Login to the Booklib and specifying pre-conditions further cases |  |
| **Actors** | | All types of readers, Authors, Recommandation System, Mood Selection System |  |
| **Stakeholders:** | | All types of readers, Authors, Managers |  |
| **Pre-Conditions** | | The user has access to Booklib |  |
| **Success end condition:** | | The User is Loged in and see the welcome screen |  |
| **Failure end condition:** | | The authentification fails |  |
|  |  |  |  |
| **Main Success Scenario** | | | **Linked UCs** |
| 1 | User inputs his login data as email and password | |  |
| 2 | System verifies credentials | |  |
| 3 | System shows welcome screen | |  |
| 4 | Mood selection | | [SUC1](file:///C:\Users\Tima\Desktop\UseCases_Traces_K12348067_K12010223%20(4).xlsx#'SUC1'!A1) |
| 5 | System selects recommended books | | [SUC2](file:///C:\Users\Tima\Desktop\UseCases_Traces_K12348067_K12010223%20(4).xlsx#'SUC2'!A1) |
| 6 | User redirected to the home screen with recommended books | |  |
|  |  | |  |
| **Alternative Scenarios** | | |  |
| 1.A1 | User is already authenticated | |  |
| 1.A2 | System shows home screen | |  |
|  |  | |  |
| **Exception Scenario** | |  |  |
| 2.A1 | System can't verify credentials | |  |
| 2.A2 | Authentification fails | |  |
| 2.A3 | Go back to step 1 | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case: Home navigation** | | | |
| **ID** | | UC2 |  |
| **Description** | | Usage of the home screen |  |
| **Actors** | | All types of readers, Authors |  |
| **Stakeholders:** | | All types of readers, Authors, Managers |  |
| **Pre-Conditions** | | The user is on home screen |  |
| **Success end condition:** | | User finds suitable book |  |
| **Failure end condition:** | | User unable to find suitable book |  |
|  |  |  |  |
| **Main Success Scenario** | | | **Linked UCs** |
| 1 | System gets 25 recommanted books and sorts them into different categorys | |  |
| 2 | System shows all the books according to the webpage layout | |  |
| 3 | User scrolls through books | |  |
| 4 | User selects interesting book | | [SUC3](file:///C:\Users\Tima\Desktop\UseCases_Traces_K12348067_K12010223%20(4).xlsx#'SUC3'!A1) |
|  |  | |  |
|  |  | |  |
|  |  | |  |
| **Alternative Scenarios** | | |  |
| 2.A1 | System select recommended books according to the mood | |  |
| 2.A2 | System adds labels for each mood recommended books and group them into subcategory | |  |
| 2.A3 | Go back to step 3 | |  |
| **Exception Scenario** | |  |  |
| 3.A1 | User cannnot find interesting book | |  |
| 3.A2 | User click button "No suitable books" | |  |
| 3.A3 | The system take this information as a previous experiance to use in next recommandations | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case: Books search** | | | |
| **ID** | | UC3 |  |
| **Description** | | User searches for books by substring |  |
| **Actors** | | All types of readers, Authors, Recommandation System, Searching System |  |
| **Stakeholders:** | | All types of readers, Authors |  |
| **Pre-Conditions** | | The user is on Home screen |  |
| **Success end condition:** | | Results of search are shown to user |  |
| **Failure end condition:** | | System unable to find books by substring |  |
|  |  |  |  |
| **Main Success Scenario** | | | **Linked UCs** |
| 1 | User enters query into the search bar | |  |
| 2 | Searching System searches Database acording to query | |  |
| 3 | Recommandation System recomments books which are close to the findings and the useres preferences | | [SUC2](file:///C:\Users\Tima\Desktop\UseCases_Traces_K12348067_K12010223%20(4).xlsx#'SUC2'!A1) |
| 4 | Display books which were found according to the search term | |  |
| 5 | Below the System displays the recommended books | |  |
| 6 | User selects a book he is interessted in | | [SUC3](file:///C:\Users\Tima\Desktop\UseCases_Traces_K12348067_K12010223%20(4).xlsx#'SUC3'!A1) |
|  |  | |  |
| **Alternative Scenarios** | | |  |
| 6.A1 | User enters a new search term | |  |
| 6.A2 | Go to Step 2 | |  |
|  |  |  |  |
|  |  | |  |
| **Exception Scenario** | |  |  |
| 2.A1 | There are no books in database which match the searching term | |  |
| 2.A2 | Notify user that nothing was found according to the search term | |  |
| 2.A3 | Go back to step 5 | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case: System maintaince** | | | |
| **ID** | | UC4 |  |
| **Description** | | Whole system life cycle including delivery and runtime |  |
| **Actors** | | Developers, Maintainer |  |
| **Stakeholders:** | | All types of readers, Authors, Managers |  |
| **Pre-Conditions** | | The system is ready for delivery |  |
| **Success end condition:** | | System is running and available |  |
| **Failure end condition:** | | System crashed |  |
|  |  |  |  |
| **Main Success Scenario** | | | **Linked UCs** |
| 1 | Different modules of application are compiled including Java backend, JS frontend and AI python model | |  |
| 2 | Modules are packed into containers | |  |
| 3 | Containers are deployed under docker on linux server | |  |
| 4 | Application is started | |  |
| 5 | Update Backup once per week | | [SUC5](file:///C:\Users\Tima\Desktop\UseCases_Traces_K12348067_K12010223%20(4).xlsx#'SUC5'!A1) |
|  |  | |  |
|  |  | |  |
| **Alternative Scenarios** | | |  |
| 2.A1 | The system is in testing phase | |  |
| 2.A2 | The system is deployed on the local machine | |  |
| 2.A3 | Go to Step 4 | |  |
| **Exception Scenario** | |  |  |
| 4.A1 | Application crashed | |  |
| 4.A2 | Crash logs are collected | |  |
| 4.A3 | Application is restarted within 5 minutes | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case: Recomendations accuracy calculation** | | | |
| **ID** | | UC5 |  |
| **Description** | | User specifies if the recommendation was accurate |  |
| **Actors** | | All types of readers, Authors, Recommandation System |  |
| **Stakeholders:** | | All types of readers, Authors, Managers |  |
| **Pre-Conditions** | | User clicks 3 dots icon on the label of the book |  |
| **Success end condition:** | | User's feedback is taken into consideration |  |
| **Failure end condition:** | | User did not leave feedback |  |
|  |  |  |  |
| **Main Success Scenario** | | | **Linked UCs** |
| 1 | User enters recommandation feedback page | |  |
| 2 | User chooses the reason why recommandation was made wrong (Checkbox) | |  |
| 3 | System changes weights of the features according to the users choice, so accuracy would be higher then 80% | |  |
| 4 | Redirects user to the home page | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
| **Alternative Scenarios** | | |  |
| 2.A1 | User checked more then 1 checkbox | |  |
| 2.A2 | The system shows additional form for user where he can selecat which reasons are more important | |  |
| 2.A3 | Go back to step 3 | |  |
|  |  | |  |
| **Exception Scenario** | |  |  |
| 2.A1 | User did not specify any reason | |  |
| 2.A2 | System doesn't change any weights | |  |
| 2.A3 | Go back to step 4 | |  |

## Supporting use cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Supporting Use case: Mood selection** | | | |
| **ID** | | SUC1 |  |
| **Description** | | User selects mood which will be used later for recommendations |  |
| **Actors** | | All types of readers, Authors, Mood Selection System |  |
| **Stakeholders:** | | All types of readers, Authors, Managers |  |
| **Pre-Conditions** | | The user is authenticated and is on welcome screen |  |
| **Success end condition:** | | The user's mood was selected |  |
| **Failure end condition:** | | Page restart |  |
|  |  |  |  |
| **Main Success Scenario** | | |  |
| 1 | The System shows mood questionary | |  |
| 2 | User selectes the current mood in the questionary | |  |
| 3 | The user can choose how strongly the current mood should be reflected in the recommandation | |  |
| 4 | The user klicks on "Okay" Button | |  |
| 5 | The questionary closes | |  |
| 6 | The System processes the input | |  |
| **Alternative Scenarios** | | |  |
| 2.A1 | User clicked "Skip" on the mood selection form | |  |
| 2.A2 | Mood is not considered in recommendations | |  |
| 2.A2 | The system returns the result | |  |
| **Exception Scenario** | |  |  |
| 6.A1 | An error accured during the data processing | |  |
| 6.A2 | The System asks the user to start over | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |

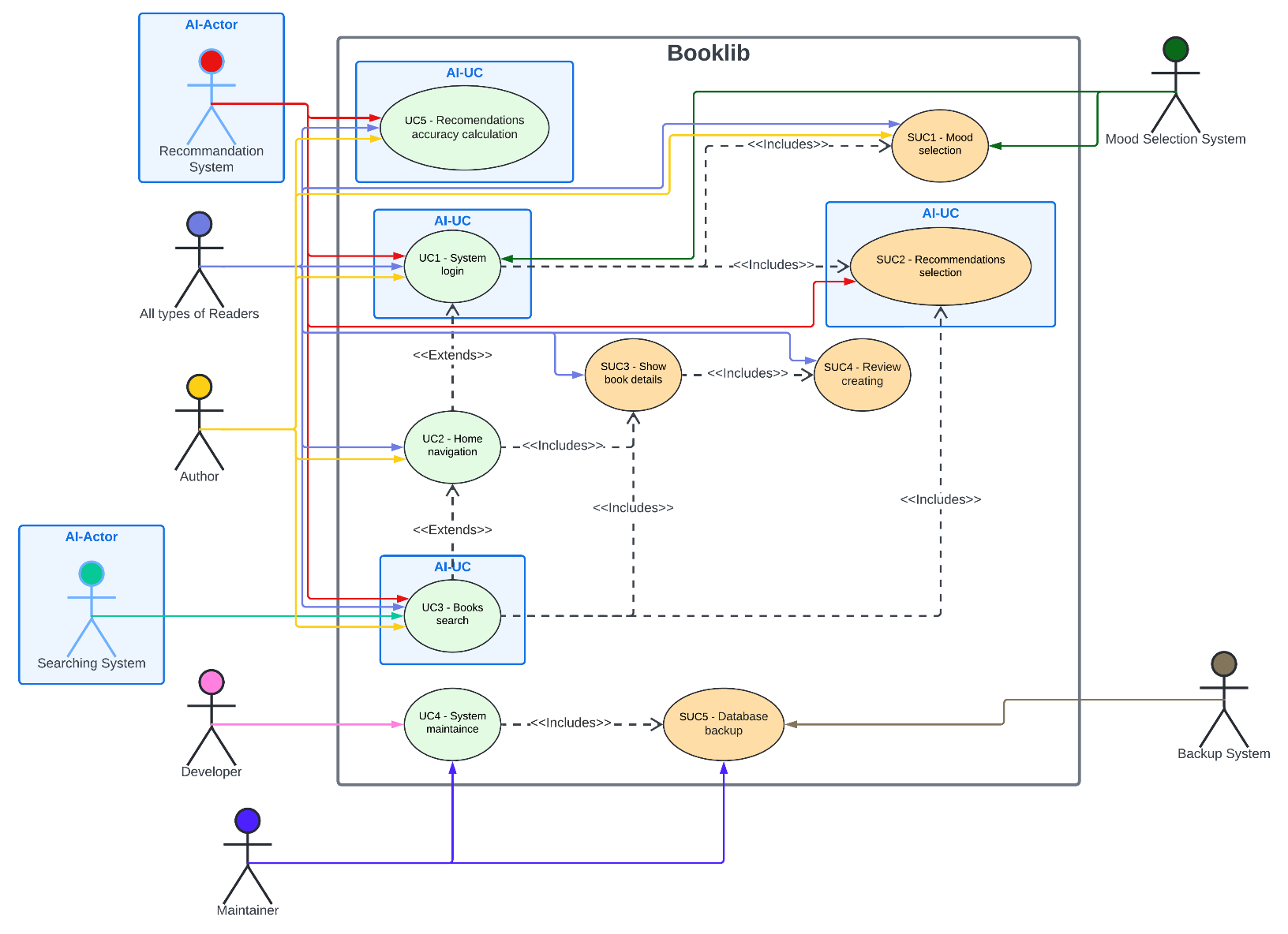
|  |  |  |  |
| --- | --- | --- | --- |
| **Supporting Use case: Recommendations selection** | | | |
| **ID** | | SUC2 |  |
| **Description** | | Selection of recommended books according to the entered data |  |
| **Actors** | | Recommandation System |  |
| **Stakeholders:** | | Authors, Managers |  |
| **Pre-Conditions** | | Mood selection is finished |  |
| **Success end condition:** | | The recommandation system returns the 25 most suitable books |  |
| **Failure end condition:** | | The processing of the recommandation took to long |  |
|  |  |  |  |
| **Main Success Scenario** | | |  |
| 1 | The System gathers data inputs from other systems | |  |
| 2 | The recommandation system applies weights of mood, previous history and trends | |  |
| 3 | The recommandation system calculates the distance to all the books | |  |
| 4 | The recommandation system takes the 25 most suitable ones | |  |
| 5 | The recommandation system returns the result | |  |
| **Alternative Scenarios** | | |  |
| 2.A1 | User skipped mood selection, so the recommendation system does not have current mood weight | |  |
| 2.A2 | The recommandation system does not apply mood weight | |  |
| 2.A2 | Go to step 3 | |  |
| **Exception Scenario** | |  |  |
| 3.A1 | Processing takes longer then 2 seconds | |  |
| 3.A2 | The System returns current best results | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Supporting Use case: Show book details** | | | |
| **ID** | | SUC3 |  |
| **Description** | | Usage of the book page |  |
| **Actors** | | All types of readers, Authors |  |
| **Stakeholders:** | | All types of readers, Authors, managers |  |
| **Pre-Conditions** | | User has selected a book |  |
| **Success end condition:** | | User gets all the book information and reviews |  |
| **Failure end condition:** | | Not all information is provided for selected book |  |
|  |  |  |  |
| **Main Success Scenario** | | |  |
| 1 | System checks that type of the user who entered system is "User" | |  |
| 2 | System retrievs book object | |  |
| 3 | The book page displays a picture of the book and some information about it | |  |
| 4 | The user scrolls down the book abstract is shown | |  |
| 5 | The user scrolls further down, the review section is shown | |  |
| 6 | After the review section recommanted books are shown | |  |
| 7 | The usere returns to where he came from (search page, home screen) | |  |
| **Alternative Scenarios** | | |  |
| 3.A1 | User clicks on review section | |  |
| 3.A2 | User is redirected to the review screen | | [SUC4](file:///C:\Users\Tima\Desktop\UseCases_Traces_K12348067_K12010223%20(4).xlsx#'SUC4'!A1) |
|  |  | |  |
| 1.A1 | User type is "Author" | |  |
| 1.A2 | The system retrievs additional metrics for Author from database and goes to step 3 | |  |
|  |  | |  |
| **Exception Scenario** | |  |  |
| 2.A1 | System retrievs book object with empty important | |  |
| 2.A2 | The System needs to show user that information is missing | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Supporting Use case: review creating** | | | |
| **ID** | | SUC4 |  |
| **Description** | | User creates a review to the currently oppened book |  |
| **Actors** | | All types of readers |  |
| **Stakeholders:** | | All types of readers, Authors |  |
| **Pre-Conditions** | | User is on review page |  |
| **Success end condition:** | | New review is added |  |
| **Failure end condition:** | | Users feedback is not saved |  |
|  |  |  |  |
| **Main Success Scenario** | | |  |
| 1 | The usere selects 1-5 Stars | |  |
| 2 | The user enters a review | |  |
| 3 | The user klicks on the "Send Review"- Button | |  |
| 4 | The system includes the review (book rating is changed, review is posted) | |  |
| 5 | The usere is redirected to the previous page | |  |
| **Alternative Scenarios** | | |  |
| 2.A1 | User did not write any text | |  |
| 2.A2 | Book raiting is changed according to the user's provided raiting | |  |
| 2.A3 | User review is not shown | |  |
| 2.A4 | Go to step 5 | |  |
| **Exception Scenario** | |  |  |
| 3.A1 | User did not send the review and closed the page | |  |
| 3.A2 | Review is nor saved, nor considered | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Supporting Use case: Database backup** | | | |
| **ID** | | SUC5 |  |
| **Description** | | Backup of system database |  |
| **Actors** | | Backup System, maintainer |  |
| **Stakeholders:** | | Maintainers, managers, developers |  |
| **Pre-Conditions** | | Deployed application is running |  |
| **Success end condition:** | | New backup is created |  |
| **Failure end condition:** | | Impossible to save backup |  |
|  |  |  |  |
| **Main Success Scenario** | | |  |
| 1 | System automatically initiates backup procedure | |  |
| 2 | Backup system copies all changed data | |  |
| 3 | Backup file is encrypted | |  |
| 4 | Backup file is place in the backups starage | |  |
|  |  | |  |
| **Alternative Scenarios** | | |  |
| 1.A1 | Maintainer initiates backup procedure | |  |
| 1.A2 | Maintainer specifies which data should be backed up | |  |
| 1.A3 | Go to step 3 | |  |
|  |  | |  |
| **Exception Scenario** | |  |  |
| 4.A1 | Backup file to big for the backups storage | |  |
| 4.A2 | Backup file is placed into temporary storage | |  |
| 4.A3 | Mentainers are notified | |  |

# Use case diagram

****

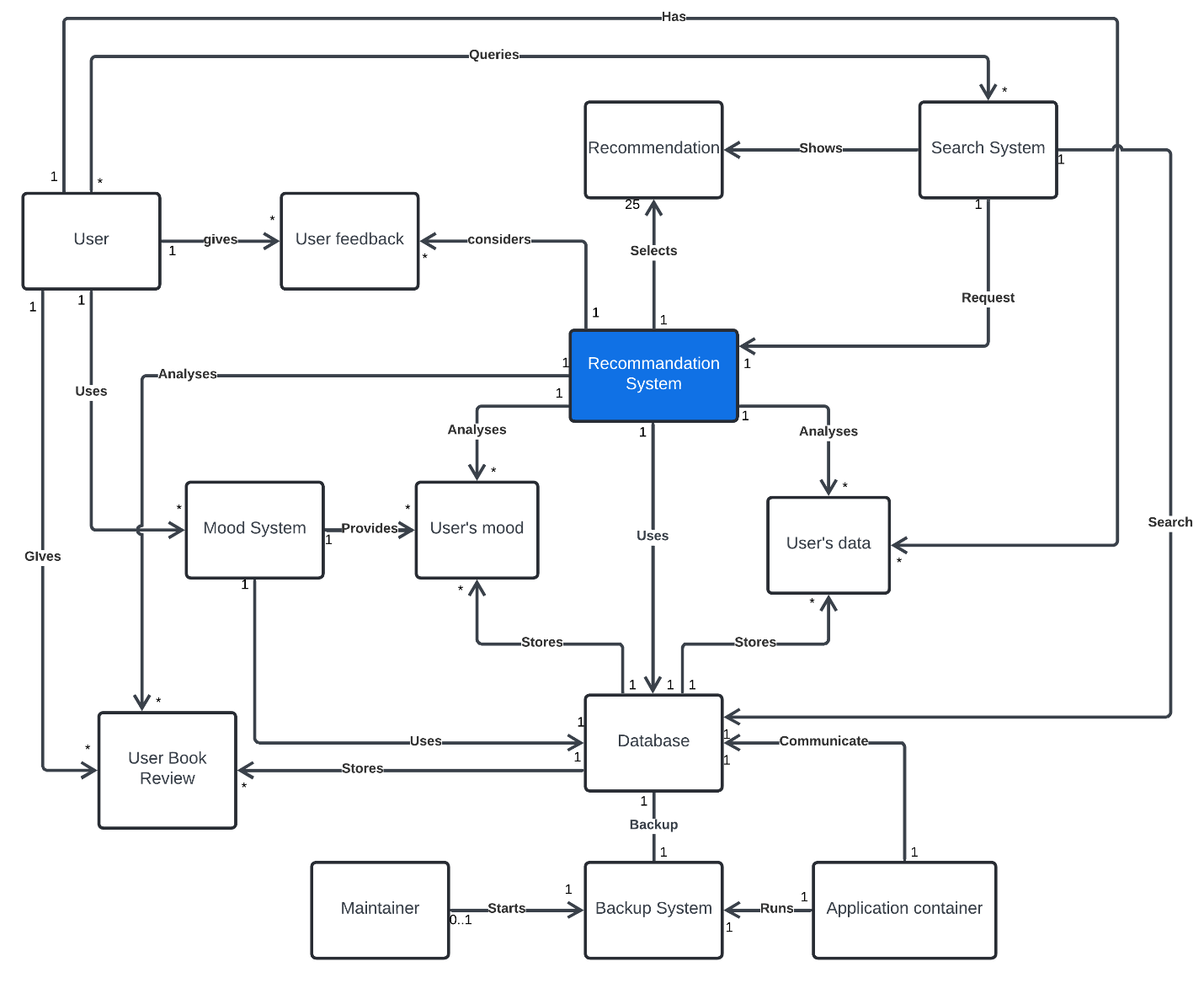
Implemented use cases:

* UC1 – System login
* UC2 – Home navigation
* UC3 – Book search
* SUC1 – Mood selection
* SUC2 – Recommendations selection

# Traceability matrix

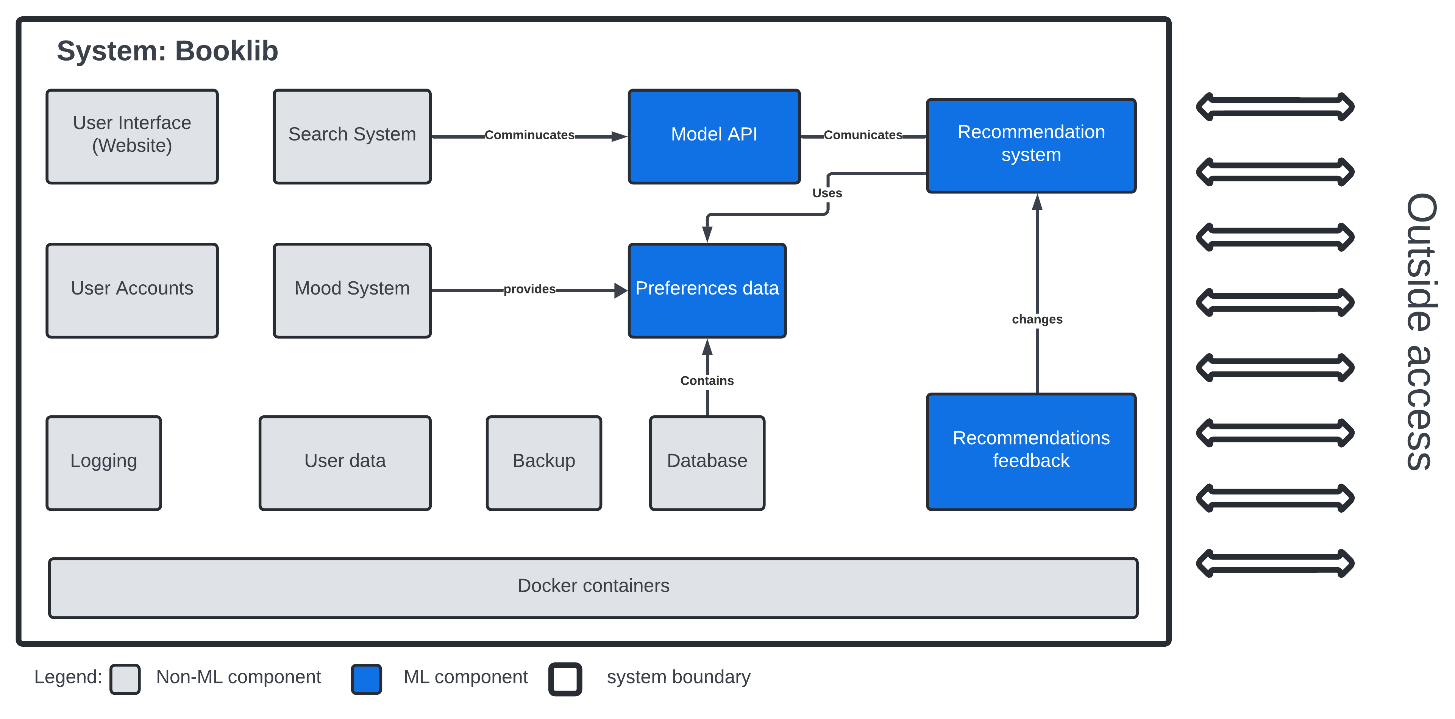
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use cases** | **UC1** | **UC2** | **UC3** | **UC4** | **UC5** | **SUC1** | **SUC2** | **SUC3** | **SUC4** | **SUC5** |
| **Requirements** |  |  |  |  |  |  |  |  |  |  |
| **Req1** | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| **Req2** | TRUE | FALSE | FALSE | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | FALSE |
| **Req3** | TRUE | TRUE | TRUE | FALSE | FALSE | TRUE | TRUE | TRUE | FALSE | FALSE |
| **Req4** | TRUE | TRUE | TRUE | FALSE | FALSE | FALSE | TRUE | TRUE | FALSE | FALSE |
| **Req5** | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| **Req6** | FALSE | TRUE | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| **Req7** | FALSE | TRUE | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| **Req8** | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE |
| **Req9** | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | TRUE | FALSE | FALSE |
| **Req10** | FALSE | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| **Req11** | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| **Req12** | FALSE | TRUE | TRUE | FALSE | FALSE | FALSE | FALSE | TRUE | FALSE | FALSE |
| **Req13** | TRUE | TRUE | TRUE | FALSE | FALSE | FALSE | TRUE | TRUE | FALSE | FALSE |
| **Req14** | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | TRUE | FALSE | FALSE |
| **NfReq1** | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | TRUE | FALSE |
| **NfReq2** | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | TRUE | FALSE |
| **NfReq3** | TRUE | TRUE | TRUE | FALSE | FALSE | TRUE | TRUE | TRUE | FALSE | FALSE |
| **NfReq4** | FALSE | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| **NfReq5** | FALSE | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| **NfReq6** | FALSE | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | TRUE |
| **NfReq7** | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| **NfReq8** | FALSE | FALSE | FALSE | FALSE | TRUE | FALSE | TRUE | FALSE | FALSE | FALSE |
| **NfReq9** | TRUE | TRUE | TRUE | FALSE | FALSE | FALSE | TRUE | TRUE | FALSE | FALSE |
| **NfReq10** | FALSE | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | TRUE |
| **NfReq11** | FALSE | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| **NfReq12** | FALSE | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| **NfReq13** | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | TRUE |
| **NfReq14** | FALSE | FALSE | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE |

# Domain model



# System architecture

## Diagram



## Description

Architecture Components:

* User Interface (Website)
  + Provides the whole user experience. The entire

communication and interaction with the user are processed

via this component.

* + Requirements: Req: 1, 2, 5, 6, 7, 9, 10, 12, 14; NfReq: 1
* Search System
  + This system enables the user to search for books by substring. It also considers books which are related to the user’s preferences.
  + Requirements: Req8
* User Accounts
  + This component handles all the things that has to do with user accounts. Account creation, E-Mail verification, User Authentication
  + Requirements: Req: 1, 5; NfReq7
* Mood System
  + This component handles the mood selection. It stores the entered data in the reference data component, where it is used in the recommendation system.
  + Requirements: Req: 2, 3, 4
* • Logging
  + Module which manages user’s session, responsible for creating new sessions and restricting access for the invalid logins.
  + Requirements: Req: 1,5
* • User data
  + Subset of data which is used for storing non-users' created data. It includes general information about all books, accounts and logging information.
  + Requirements: Req1
* • Backup
  + In order not to lose the information, system needs to do additional copies. These backups need to be made regularly and be encrypted in order to prevent illegal data access to sensitive information.
  + Requirements: NfReq10
* Model API
  + Additional layer for communication of webpage backend with AI model written in python
  + Requirements: NfReq12
* Preferences data
  + Subset of data which is used by AI model for selecting best recommendations. Data is collected from users and stored in database.
  + Requirements: Req: 9, 13
* Database
  + Contains all application information including data needed for AI model. All sensitive data is accessed by system and changes can introduce only maintainers.
  + Requirements: NfReq: 6, 13
* Docker containers
  + The whole system run under 4 containers in Linux: model, backend, frontend, database. Their status is monitored maintainers are notified if something is wrong. In case of crush, container is restarting.
  + Requirements: NfReq: 1, 2, 5, 11
* Recommendation system
  + This is the main AI component of our system. It computes the most suitable books for the user according to their preferences and current mood. In the case of a book search it recommends the books which are near according to the substring and the user preferences. It returns the best 25 books.
  + Requirements: Req3, 4, 6, 8, 10, 13; NfReq: 3, 8, 9, 12, 14
* Recommendations feedback
  + Users should be able to provide feedback regarding how accurate recommendations were. System will consider it later for improving the accuracy. As well general metrics should be generated as accuracy of the system.
  + Requirements: NfReq8

# Design questions

1. Can we measure the accuracy of our recommendations system?
   * When the user is not satisfied with one recommendation then he can specify that explicitly via the Recommendations feedback. We can count how often he does that to the overall recommendations.
2. When is a recommendation wrong?
   * When the user explicitly specifies it as wrong. He must also enter a reason why the recommendation was wrong and according to that information the recommendations are changed.
3. How can we estimate if user is satisfied with the system?
   * We can estimate it via a questionnaire, but it is more likely that only the users’ answers which are not satisfied (Non-Response Bias). We can also estimate it with the average duration of use per month.
4. How can we keep our database up to date? So that it contains all the important books.
   * When our system is more well known then the authors will enter the books themselves, as they benefit from it. But in the beginning the system maintainer or site owner will be responsible for this.
5. Is there a risk for recommendation of unproper books for younger readers?
   * During account registration process, user will have to provide an age. Additionally, each book has reading age. This information has to be taken into the account during the recommendations’ computation.
6. When do we need to take into account users’ feedback regarding recommendations accuracy?
   * Firstly, users’ feedback should be anonymous in order not to expose any personal information. Then feedback should be verified with the system and should be compared with other feedbacks. After this, weights of parameters can be changed.
7. Will the website be usable at mobile devices?
   * The front-end should be written in the modern framework which make it possible to do general interface for both mobile and stationary devices.
8. How can we correctly include the mood into the recommendations? So that it is not too much or less for different users.
   * The easiest way to do it is to let the user decide how much it should be included in the recommendations. The more advanced version would be, that we analyse the users’ behaviour how he likes our mood recommendations and decide according to that information.
9. How do we handle books with bigger rating in the recommendation?
   * We can consider rating as a weight and update it once per some time according to the changed book rating.
10. Would it be possible to scale the system for more professional domains (e.g. recommendation of scientific articles)?
    * System can be scaled into more specific domain with adding additional information tags for each searching term (book or article in this case). Then users with bigger interest in specific topic can get more accurate recommendations.
11. What do we do, when we have to much data to consider for the recommendation so that it takes too long?
    * When it takes too long to consider all the data, then we need to randomly subsample the data and choose only as much data as possible.
12. How should we deal with increasing number of requests?
    * First of all, system metrics should be observed. When we are at the hardware limitations, we need allocate more resources, which leads to more hardware. So, system infrastructure should be ready for hardware scaling.