# MILESTONE 1

**Keti Hysi**

## 1. Conceptual modeling

### 1.1. Business Model Outline

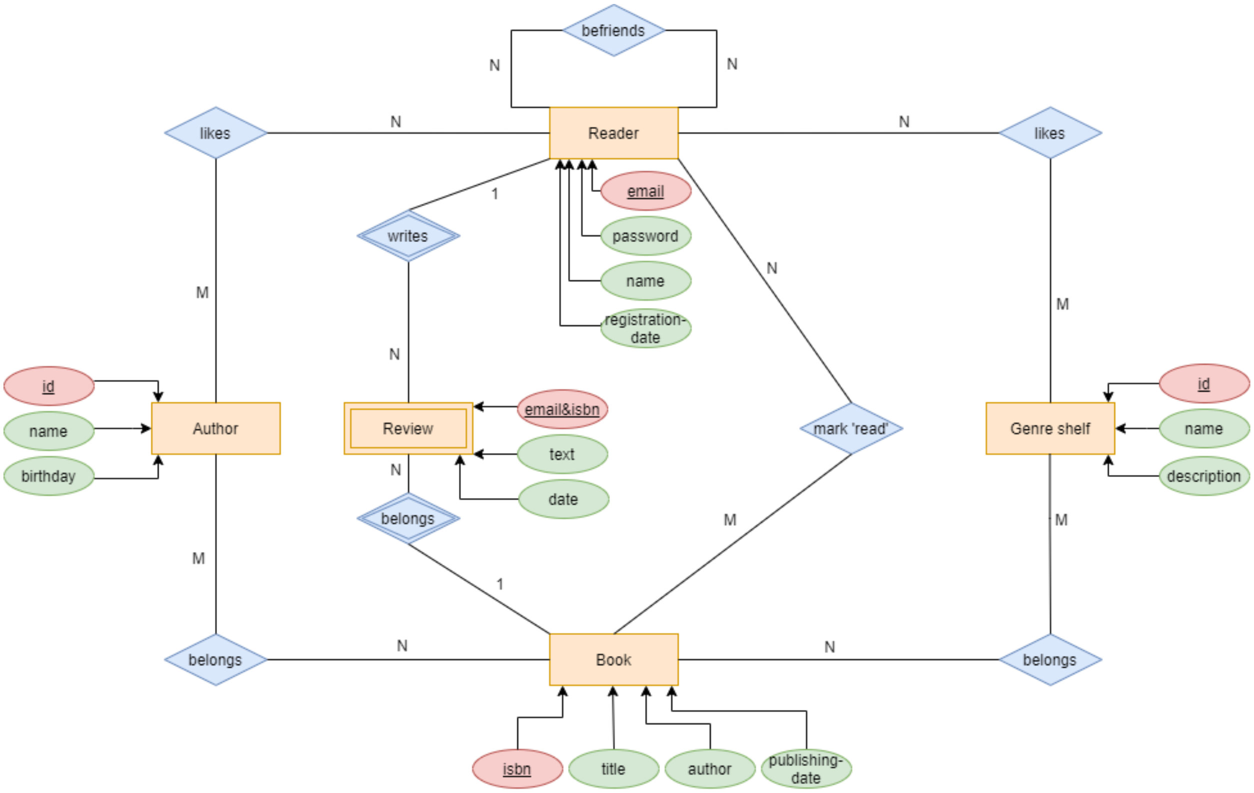
For this project, a website dedicated to book readers is aimed to be built. The general idea of it will be the following: readers can keep track of their read books, like their favorite genres and authors, review books they have already read and befriend fellow bookworms.

There will be only one type of user, the reader. Readers will be identified by their personal emails, and will also have a password, a name and a registration date. For simplicity reasons, there will be no admin user, instead I will be the one managing the content, so I will act as one.

Readers can mark an unlimited number of books as ‘read’. Books will be identified by their unique ISBNs, and the system will also store the title, the publishing date and the author(s) as a foreign key. That means that the authors will have their books, but each of these entities can exist in the system without the other. Authors will be identified by an id, and will also have a name and a birthday. Books will also belong to one or more genre shelves. Shelves will have their ids as identifiers, a name and a short description. On delete of either a shelf or a book, no action is triggered upon the other entity.

The reader can also leave at most one review on a specific book, while the book itself can have many reviews by different users. Reviews will be stored separately and will be identified by the combination of the user’s email and book’s ISBN. In case that any of these two (reader or book) gets deleted, so will the respective review. For reviews, the date of creation and the text will be stored. Reader can also like as many genre shelves and as many authors as they wish. They can also befriend as many other users as they want.

## 1.2. ER Diagram



# 2. Use Case Design

## 2.1 First main use case - Register as a reader

Objective: Register as a reader

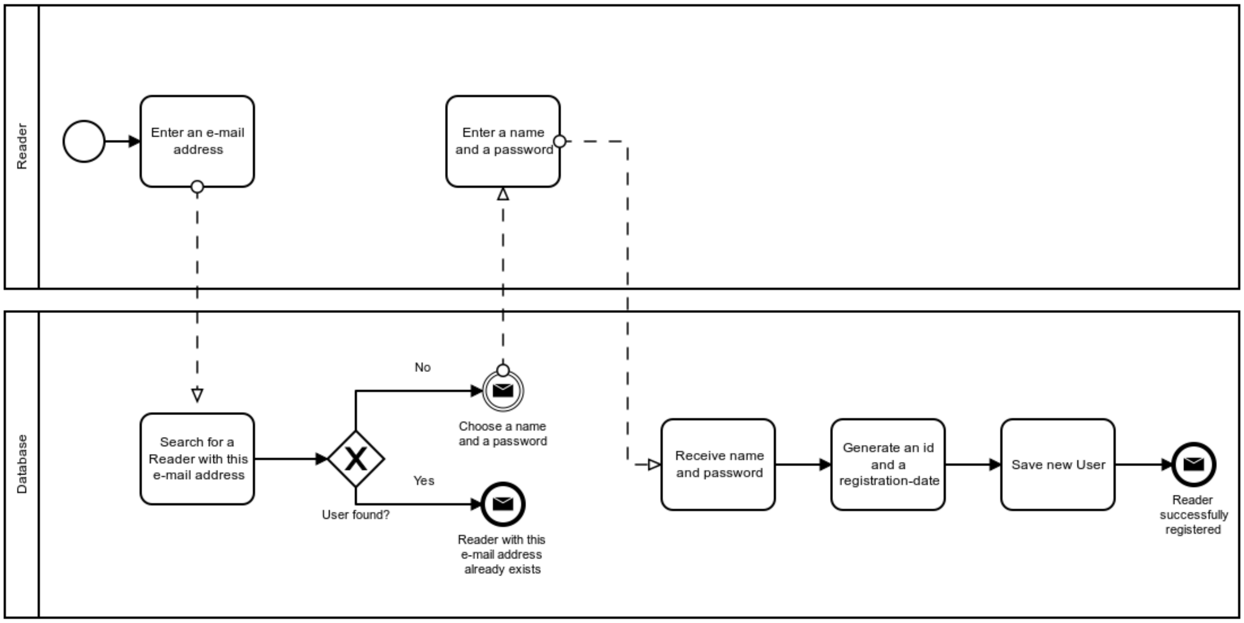
Description: A reader registers on the book review site.

Preconditions: None

Expected Execution: The users enter an e-mail address. They enter a name and a password.

Postcondition(success): The user's data is saved in the database; they are registered successfully.

Postcondition(error): There already exists a user with this e-mail address in the database. An error message is displayed.


Reader
Enter an e-mail address
Enter a name and a password
Database
Search for a Reader with this e-mail address
User found?
No
Yes
Choose a name and a password
Reader with this e-mail address already exists
Receive name and password
Generate an id and a registration-date
Save new User
Reader successfully registered

### 2.2 Second main use case - Leaving reviews

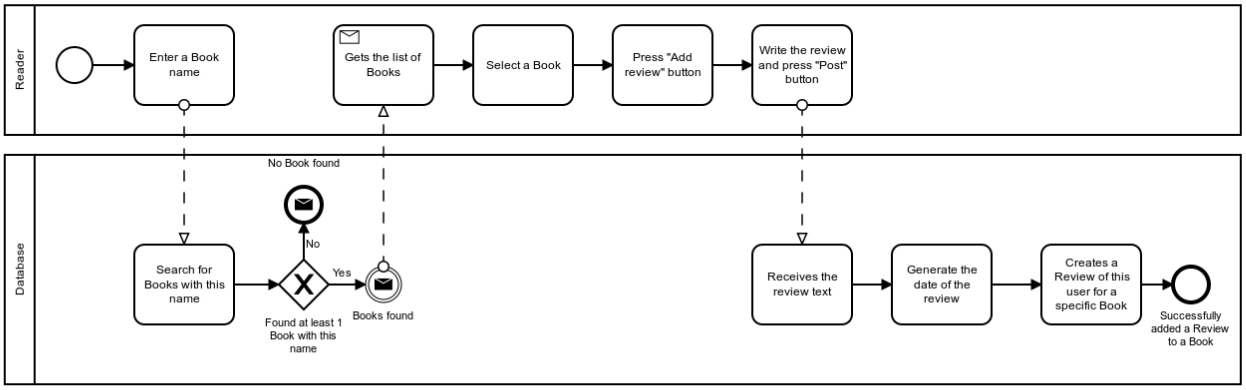
Objective: Reader writes a review.

Description:  The user leaves a review on a specific book.

Precondition: At least one reader is registered in the system, and there exists at least one Book. The reader has not previously commented in that book.

Expected Execution: User searches and selects a book to leave a review on, then clicks on “Add review” button, types the review text and clicks “post”.

Postcondition(success): The review is posted in the selected book and stored in the database.


Reader
Enter a Book name
Gets the list of Books
Select a Book
Press "Add review" button
Write the review and press "Post" button
Database
No Book found
No
Search for Books with this name
Yes
Found at least 1 Book with this name
Books found
Receives the review text
Generate the date of the review
Creates a Review of this user for a specific Book
Successfully added a Review to a Book

## 2.3 Third use case - Befriending reader

Objective: Befriending reader

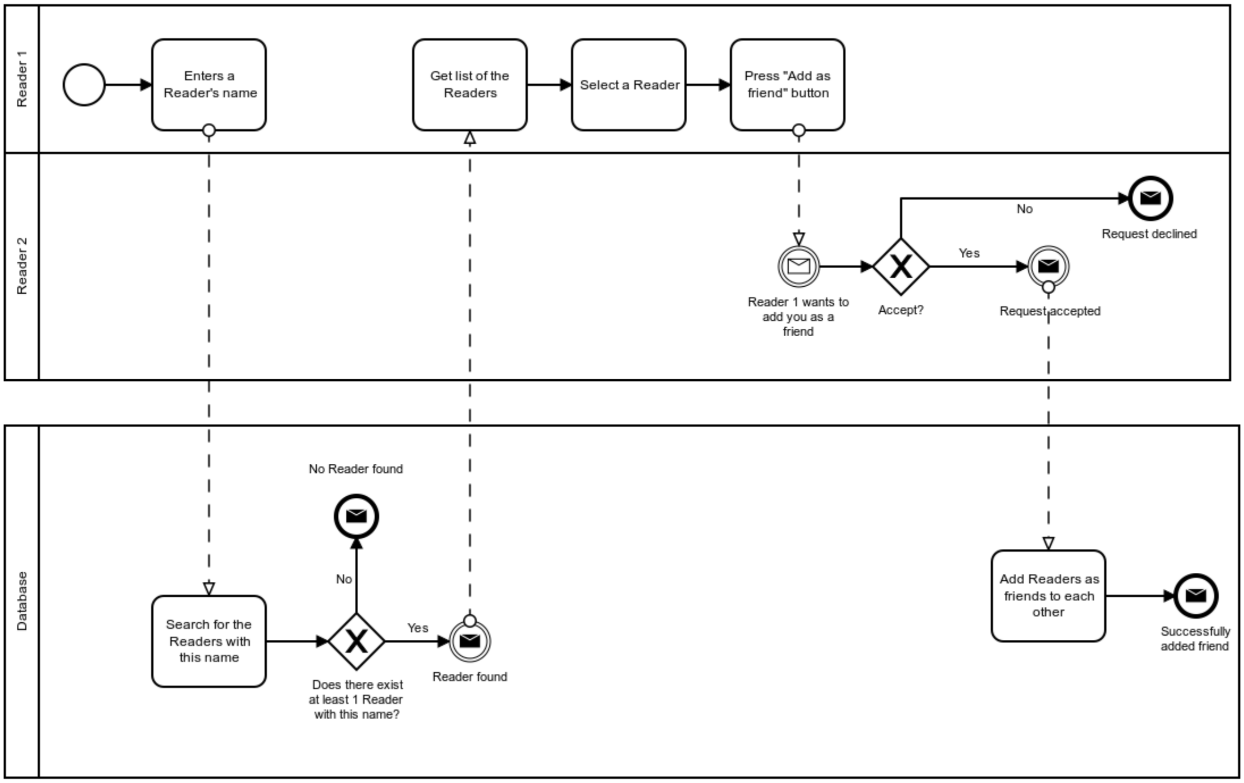
Description: A reader befriends another Reader.

Precondition: At least two Readers are registered.

Expected Execution: One Reader enters a name. He/She then selects a Reader from the list and presses the "Add as friend" button. The selected Reader receives a message and can accept or decline this friend-request.

Postcondition(success): Both Reader's lists of liked friends get updated, as each is added in the other's list.

Postcondition(error): In case no Reader with the entered name was found, the Reader is notified via an alert message.


Reader 1
Enters a Reader's name
Get list of the Readers
Select a Reader
Press "Add as friend" button
Reader 2
No
Request declined
Yes
Reader 1 wants to add you as a friend
Accept?
Request accepted
Database
No Reader found
No
Yes
Search for the Readers with this name
Does there exist at least 1 Reader with this name?
Reader found
Add Readers as friends to each other
Successfully added friend

### 2.4 Fourth use case - Reader likes an author

Objective: Liking an Author

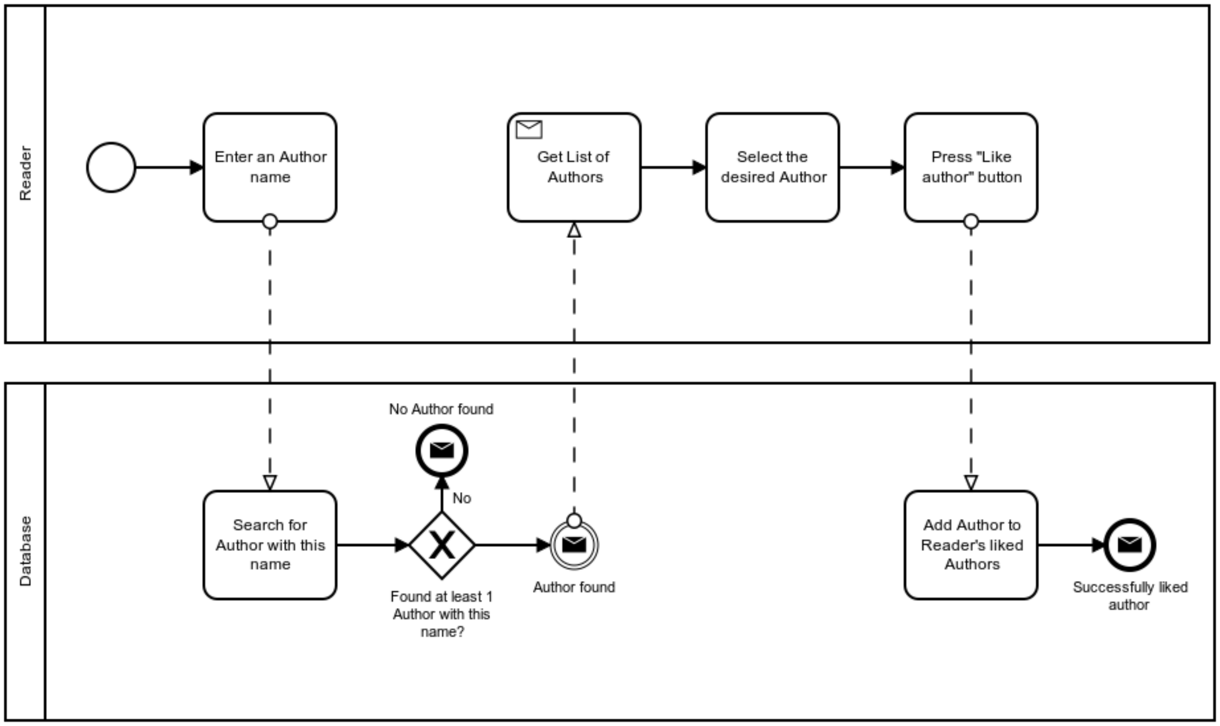
Description: A reader likes an Author of choice.

Precondition: At least one Reader is registered, there exists at least one Book.

Expected Execution: The Reader enters an Author name, then selects an Author from the list of Authors and presses the “Like author” button.

Postcondition(success): The Reader’s list of liked Authors gets updated; the selected Author is added.

Postcondition(error): In case no Author with the entered name was found, the Reader is notified via an alert message.


Reader
Enter an Author name
Get List of Authors
Select the desired Author
Press "Like author" button
Database
No Author found
No
Search for Author with this name
Found at least 1 Author with this name?
Author found
Add Author to Reader's liked Authors
Successfully liked author

### 2.5 Fifth use case - Reader marks a book as “read”

Objective: Marking a Book as “read”

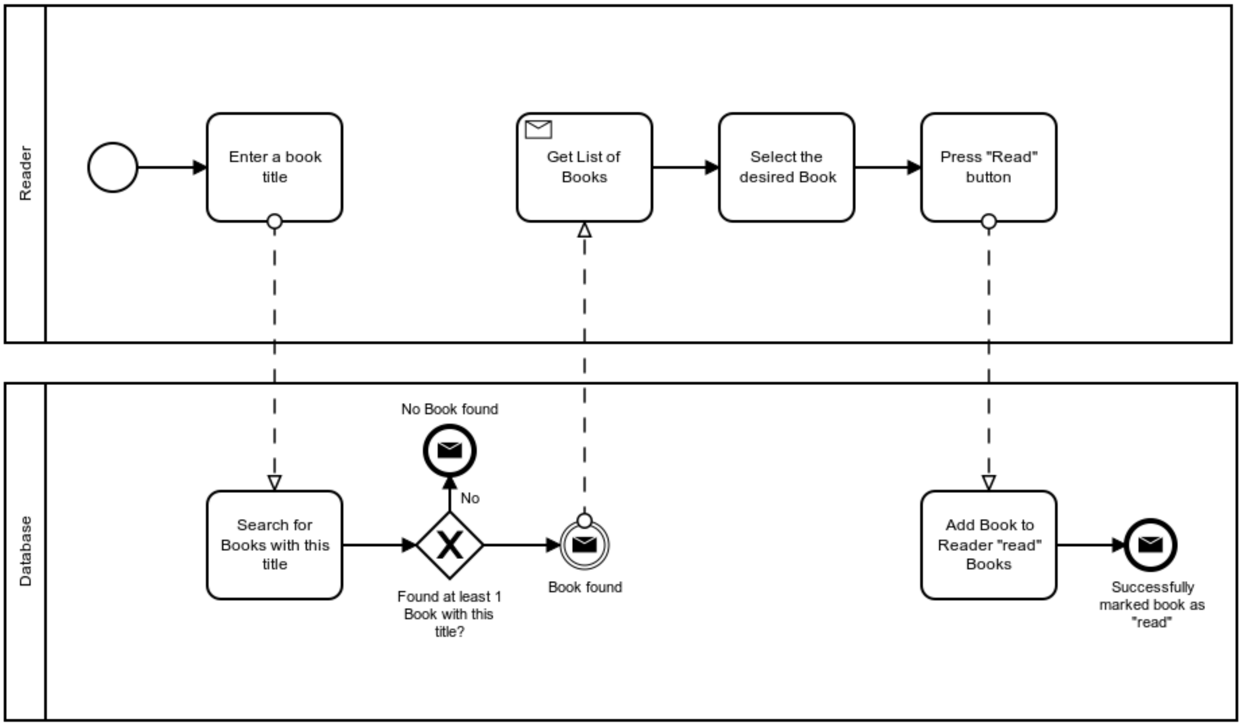
Description: A reader marks a Book as “read”.

Precondition: At least one Reader is registered, there exists at least one Book.

Expected Execution: The Reader enters a Book title, then selects a Book from the list of Book and presses the “Read” button.

Postcondition(success): The Reader’s list of read Books gets updated; the selected Book is added.

Postcondition(error): In case no Book with the entered title was found, the Reader is notified via an alert message.


Reader
Enter a book title
Get List of Books
Select the desired Book
Press "Read" button
Database
No Book found
No
Search for Books with this title
Found at least 1 Book with this title?
Book found
Add Book to Reader "read" Books
Successfully marked book as "read"

# 3. Reporting

## 3.1 First report

**Which are the most read books of a genre?**

Top 5 most read books of a genre: A report about top 5 most read books that belong to a specific genre sorted by the number of readers per book.

- Use entries: Book, Reader, Genre Shelf
- Filter by: Genre
- Sort by: Number of readers per book

## 3.2 Second report

**Which are the most reviewed books of an author?**

Top 5 most reviewed books of an author: A report about top 5 most reviewed book of a specific author sorted by the number of reviews per book.

- Use entities: Author, Book, Review
- Filter by: Author
- Sort by: Number of reviews per book