

Technical Report - **Product specification**

ReadEase Management

Course: IES - Introdução à Engenharia de Software

Date: Aveiro, 15/11/2023

Students: 107463: Pedro Miguel Ribeiro Rei
108615: Tiago Fonseca Cruz
107323: Vasco Miguel Fernandes Faria
108304: Miguel Soares

Project abstract: Library Management System

Table of contents:

[1 Introduction](#)

[2 Product concept](#)

[Vision statement](#)

[Personas](#)

[Main scenarios](#)

[3 Architecture notebook](#)

[Key requirements and constraints](#)

[Architctural view](#)

[Module interactions](#)

[4 Information perspective](#)

[5 References and resources](#)

1 Introduction

In an increasingly digital world, there arises the need for an efficient tool that simplifies the experience in libraries.

ReadEaseManagement was created with this purpose, within the context of the final project for the Introduction to Software Engineering course.

This document describes the exploration undertaken during the execution of this project, showcasing Personas, Use Cases, Architecture, etc.

Each member of our team has a defined role to efficiently divide the workload.

- Pedro Rei - *Team Manager*
- Tiago Cruz - *Product Owner*
- Miguel Soares - *Architecture*
- Vasco Faria - *DevOps*

2 Product concept

Vision statement

The ReadEaseManagement emerges as an innovative solution, aiming not only to optimize library management but to transform the experience for all involved. By streamlining the reservation of resources such as rooms and books, we provide users with a more agile and personalized interaction.

For librarians, the system goes beyond, providing valuable insights through detailed statistics. Now, library administration extends beyond physical organization, encompassing a deep understanding of user behavior and the overall state of the library.

We believe that ReadEaseManagement is not just a functional tool but a fundamental piece to boost efficiency, enhance user experience, and modernize library management for the challenges of the 21st century.

Personas and Scenarios

1st Persona

Jorge Silva is a 54-year-old librarian, who lives in Estarreja with his wife and two children, aged 20 and 16 respectively.

Jorge is a very happy person and always likes to play with everyone. However, he is very dedicated to his work and enjoys seeing work done, despite people thinking otherwise.

Within the library, he is responsible for different areas, such as contacting users at the end of the reservation period so that they can return books, booking rooms, etc.

Due to the large number of people at the library and the reservation of books and ebooks, Jorge is not always able to manage the library in a good way.

2nd Persona

Ana Margarida is a student at the University of Aveiro, who is far from her family who lives in her homeland, in Beja. Due to the distance, Ana only travels to Beja 1 weekend per month.

She is a very introverted person with some distant friends. This distance makes you feel isolated and sad, but that doesn't stop you from wanting to learn and do more.

As for her academic life, Ana Margarida has always been an excellent student, committed to her purpose and an assiduous reader. She goes to the university library regularly to study and also to borrow books.

Product requirements (User stories)

Jorge

User Story 1: As a librarian, I want to be able to change the availability of books, rooms and ebooks, to keep the library catalog up to date.

User Story 2: As a librarian, I want to manage book loans and returns, including the ability to record the date of loan and return. Additionally, I want to receive notifications when book return deadlines are about to expire, so that I can contact you with users and avoid delays. I also want to be able to view statistics about library usage, such as the number of books borrowed, the most popular books and user borrowing history, to improve library management.

User Story 3: As a librarian, I plan to manage the availability and reservations of reading rooms in the library, including the denying reservation requests and status.

Ana Margarida

User Story 4: As a frequent library person, I want to be able to create a wish list of books I plan to read in the future and receive notifications when those books are available at the library.

User Story 5: Additionally, I want to receive personalized recommendations based on my reading history and previous reviews to discover new books of interest.

User Story 6: As a frequent library user, I want to be able to share my reviews and book recommendations with other users, creating a community of readers on the platform.

3 Architecture notebook

Key requirements and constrains

The system must always be up to date and consistent.

In the system, librarians should be able to use the system on all library computers. They should have access to all books, requests, requested books, rooms, dates, and user information.

In the system, users should be able to view physical books and available rooms in the library for reservation. They should also be able to read ebooks within the app.

The system should be able to generate statistics for better library management.

Architetural view

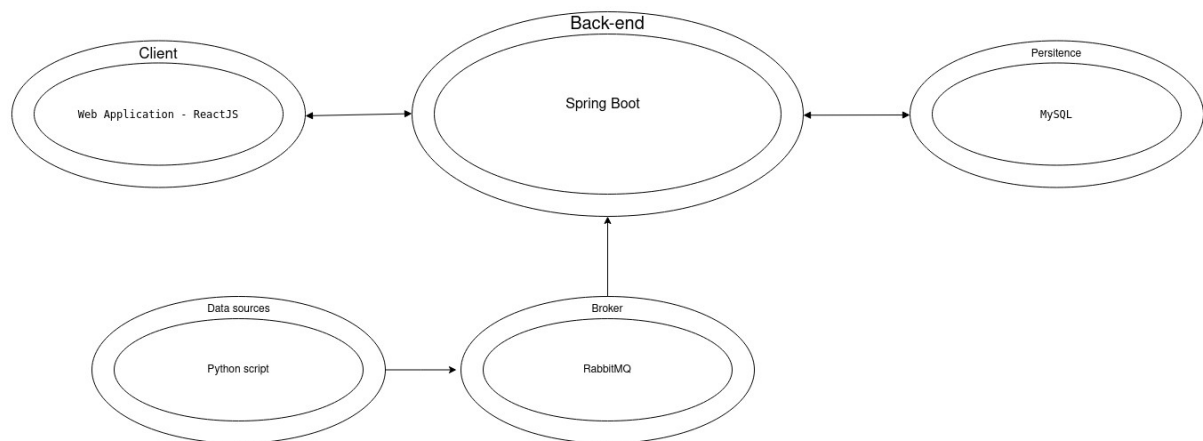
Data Sources: Using a Python script, data will be generated regarding the available reading resources, book loans and reservations, and reading room reservations in the library.

Broker: It will receive information from the data generation process and send it to the backend, where it will be processed and stored (RabbitMQ).

Backend: It will have access to the database and communicate with both the client and the broker. This is where the main processing will take place (SpringBoot - Rest API, Broker, and Database).

Client: A web application developed based on the React JS template, later adapted to both visualization and communication needs with the backend.

Persistence: The system's database (MySQL) is of the relational type, and it is crucial to securely store all the data generated by the system.



Module interactions

1. User Interaction Module:

This module handles user interactions through the React JS web application.

Interaction:

- User logs in and interacts with the system by making reservations for reading resources (books or rooms).
- The system sends reservation requests to the Reservation Management Module.
- User informs the system when picking up or returning resources.

2. Reservation Management Module:

Manages the reservation of reading resources and rooms.

Interaction:

- Receives reservation requests from the User Interaction Module.
- Communicates with the Data Sources to check resource availability.
- Updates the database with reservation information.
- Monitors resource status and due dates.
- Sends notifications to users based on resource status (e.g., close to return date).

3. Data Sources Module:

Generates data regarding available reading resources, book reviews, book loans, and reservations.

Interaction:

- Generates data through a Python script.

- Sends generated data to the Broker.

4. Broker Module:

Acts as an intermediary between the Data Sources and the Backend.

Interaction:

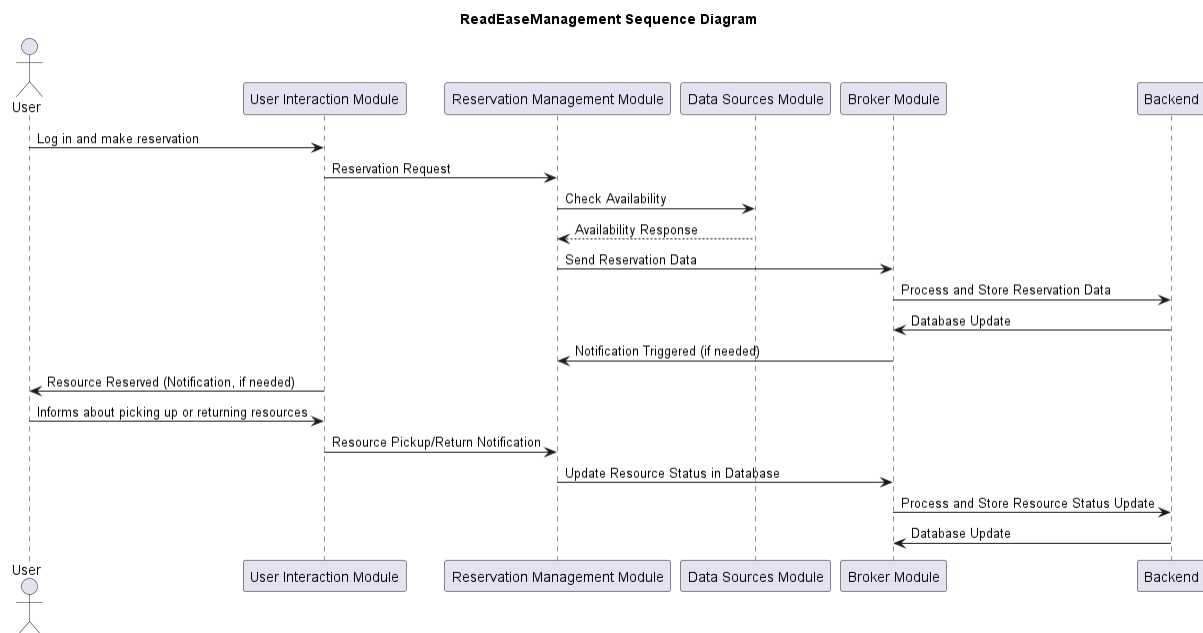
- Receives data from the Data Sources Module.
- Sends data to the Backend for processing and storage through RabbitMQ.
- Manages the communication flow between different modules.

5. Backend Module:

Handles the main processing of data, communicates with the database, and interacts with the Client.

Interaction:

- Receives data from the Broker through RabbitMQ.
- Processes data and stores it securely in the MySQL database.
- Communicates with the Client to send and receive information.
- Provides valuable insights to the librarians through statistics.



Advanced App Design Issues:

1. Integration with Internet-based External Services:

- Integrate external services for book reviews, online catalogs, and other relevant information.
- Use APIs to fetch data from external sources and enhance the richness of information within the system.(Google Books API)

2. Data Synchronization Strategy:

- Ensure that changes in reservations, book availability, or user information are reflected instantly across all modules.

3. Distributed Workflows:

- Design workflows that can be distributed across multiple servers or cloud instances.
- Enable load balancing to distribute processing tasks efficiently.

4. Push Notifications Mechanism:

- Implement push notifications to alert users about reservation due dates and available books.
- Utilize services like Firebase Cloud Messaging (FCM) for efficient push notification delivery.

5. Distribution of Updates to Distributed Devices:

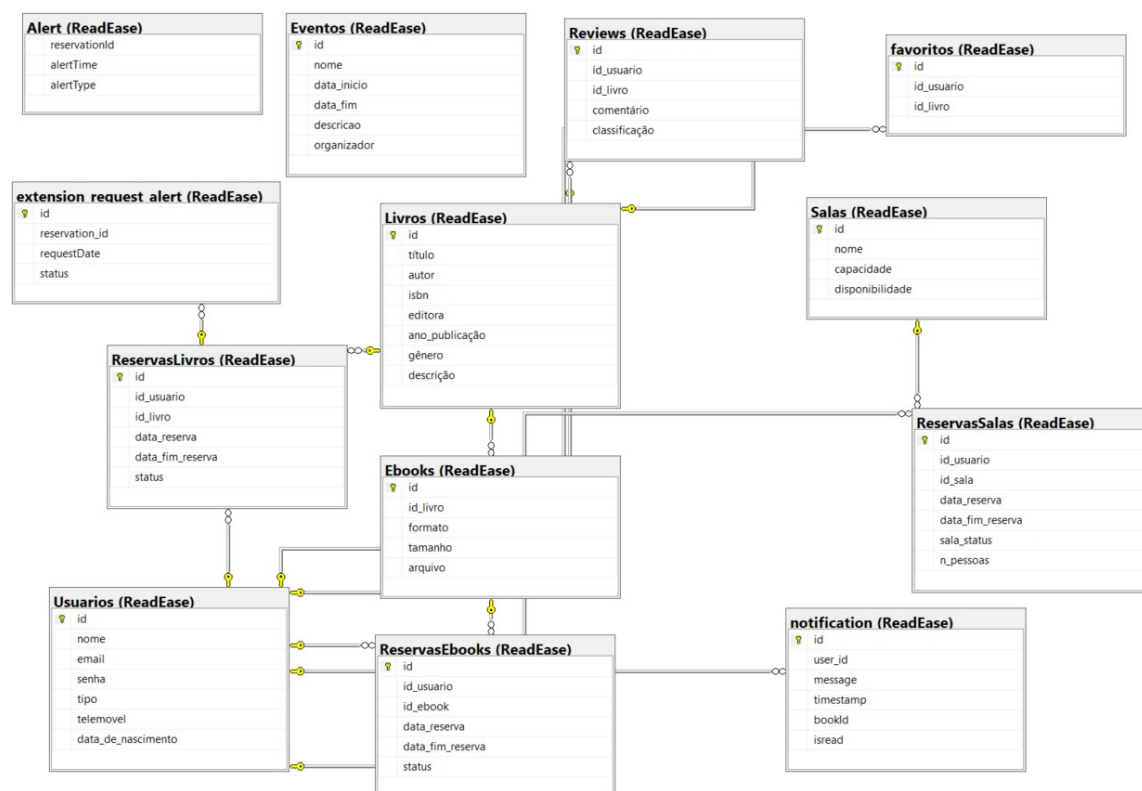
- Use a versioning system to manage updates and distribute them seamlessly to all instances of the web application.
- Implement an update mechanism that ensures minimal downtime during the update process.

In summary, the ReadEaseManagement system follows a modular architecture with clear interactions between modules. The system integrates with external services, employs real-time data synchronization, handles distributed workflows, implements push notifications, and ensures efficient distribution of updates to provide a seamless and modern library management experience.

4 Information perspective

Some entities are related to each other, which makes the option of a relational database more suitable. Therefore, in a joint decision of the group, we chose to use MySQL as the database.

Here is the database structure:



5 References and resources

<https://otonomo.io/blog-redis-kafka-or-rabbitmq-which-microservices-message-broker-to-choose/>

<https://swagger.io/>