

Technical Report - **Product specification**

# BoardGamesHub

Course: IES - Introdução à Engenharia de Software

Date: Aveiro, 27/10/2023

Students: 105937: Bárbara Nóbrega Galiza  
107969: João Miguel Dias Andrade  
109018: Tomás António de Oliveira Victal  
109986: Pedro Daniel Fidalgo de Pinho

Project abstract: BoardGamesHub is a web application for comparing board games prices across multiple web stores, providing their current price and showcasing their price history.

## Table of contents:

### [1 Introduction](#)

### [2 Product concept](#)

[Vision statement](#)

[Personas](#)

[Main scenarios](#)

### [3 Architecture notebook](#)

[Key requirements and constraints](#)

[Architectural view](#)

[Module interactions](#)

### [4 Information perspective](#)

### [5 References and resources](#)

## **1 Introduction**

This project was made in the scope of the class “Introduction to Software Engineering”, with the intention of learning and applying software engineering best practices. In this case, the proposed method was the agile process, alongside with the use of user stories and scenarios, and the theme was given by the students. Our product choice was a web application for board games price comparison, the BoardGamesHub.

This report describes the product concept, the architecture and the information model in which the system is based on.

## **2 Product concept**

### **Vision statement**

The BoardGamesHub application will be used by board games players to find the lowest price of a certain game across the offers of many vendors. This idea was based on already existent websites for price comparison, like “Kuantokusta” and “Trivago”, with the difference being the scope of the product.

The choice of the scope “Board games” was made based on the fact of it being a not very well explored subarea within the price comparison area, and it being a business domain known by one of the team members.

In that sense, alongside with the price comparison, our website will showcase detailed information about the games and a price history to assist the user on making an even better choice on what game, where and when to buy it.

## Personas and Scenarios

John is our common user, he is 47 years old, male, and lives in California with his wife.

He's a big board game collector and always wants to keep track of when the newest board games are coming out.

Due to this hobby being very expensive, he requires a way to help him keep growing his collection with a somewhat tight budget.



Matilda is our admin, she is 36 years old, female, and is currently engaged.

She's a very busy person due to running multiple businesses, so her time is short. She wants things done fast and efficiently.

When she hears that there are problems in Board Games Hub, she wants them fixed fast. Be they wrong products in the catalog, outdated information or an inappropriate review.

### Scenario 1:

John has been looking forward to buying "Voidfall" but has been waiting for a sale to come around. Suddenly an e-mail notification saying that this game was on sale.

### Scenario 2:

Matilda heard a complaint from a company that they do not wish to have their game displayed on our frontend. She is quick to act.

## Product requirements (User stories)

John (User):

Epic: Finding the best price for a specific game.

- As John, I want to see all the prices for a game across multiple stores, so that I can buy the game for the lowest price there is.
- As John, I want to consult the price history of a board game, so that I can identify the best time to buy it.

Epic: Searching and comparing various games.

- As John, I want to have access to a list of all board games, along with filters to help me find games within certain categories.
- As John, I want to have recommendations based on the games I play/look at so that I can find similar games with a better price.
- As John, I want to edit my account preferences (price range, board game types, language, etc) so that they match my wants when searching for a game.
- As John, I want to search for board game designers, publishers, companies and artists, to help me find similar games with a better price.

Epic: Wishlist management.

- As John, I want to be able to create a wishlist, so that I can keep track of all the board games I wish to get.
- As John, I want to be able to update my wishlist, in order to add new games that interest me and remove games I'm no longer interested in.
- As John, I want to receive a notification informing me that one of my games in my wishlist is currently on sale, so that I can buy it for a lower price.

Matilda (Admin):

Epic: Website management.

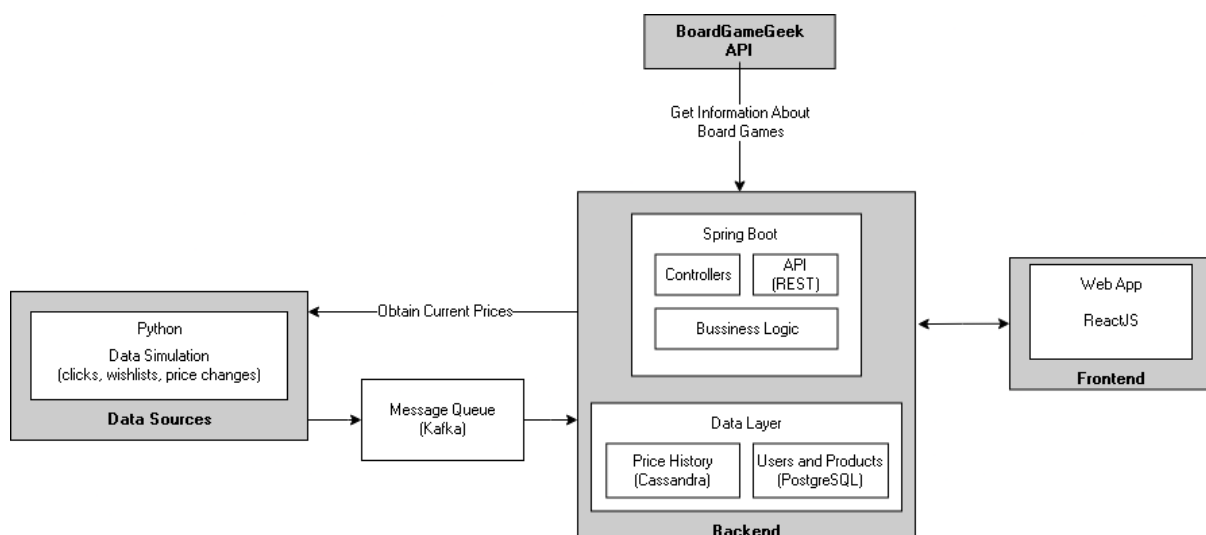
- As Matilda, I want to be capable of adding a new game to be tracked in that site, in order to give more game options to the clients.
- As Matilda, I want to be capable of removing a game upon vendor request, so that the site follows the business rules.
- As Matilda, I want to be capable of updating some game information that it's not automatically fetched, to keep the website up to date.
- As Matilda, I want to create events and send them to my subscribers, to attract new sales.

### 3 Architecture notebook

#### Key requirements and constrains

- The data sent from the data sources must not be lost in any situation. In case of a failure this data must be saved.
- All the prices changes must be stored so that the users are able to view the price history for a product.
- An admin must be able to, when adding a new board game, fill the information about the game automatically, through the information present on BoardGamesGeek API.
- There needs to be a efficient database retrieval and a fast way to analyze the data related to price and price changes.
- The data related to Users and Products must be stored in a consistent manner.

#### Architeturall view



In the Frontend, we will use ReactJS since it allows use to separate the frontend from the backend. React can also easily consume the RESTful APIs created in Spring Boot.

For the Backend, we will use Spring Boot since it allows us to easily implement the controllers, the API, the buisness logic and to easily get the website running. In the Data Layer we will have two databases, one for storing the price history of the products and the other for storing the rest of the information (users, products, etc). We decided to use Cassandra for our prices database because, since it is a column-based database, it allows for easier analysis of large amounts of data. For the other database, we settled on PostgreSQL since it is one of the most popular relational databases.

Our data sources will be emulating the fluctuations in prices and emulating user interactions (clicks, wishlists, etc). This will allow us to, for example, give recomendations to users based on what is most popular or most similar to what the user likes. To pass this

information, we will use Kafka. We chose Kafka because it has data persistence, making it so information doesn't get lost if the service goes down.

In our backend, we also communicate with the BoardGamesGeek API to get information on the different board games. This will allow us to get information for newly added board games.

## Module interactions

The Frontend will be communicating with the controllers in the backend. Information like price changes will be sent to the frontend and, for example, be shown as notifications in case the user has the game in the wishlist.

In the backend, we will get the information about the board games using the BGG (BoardGamesGeek) API. With this API, we are able to obtain information like ratings, ranking, difficulty, categories, etc. This information will also be used to give the users recommendations based on categories the user likes and for search filtering.

Our data sources will retrieve the current prices from the backend through the API and generate new prices to simulate price changes. It will also simulate user actions. The new data will then be sent back to the backend using Kafka as the message queue.

## 4 Information perspective

<which concepts will be managed in this domain? How are they related?>

<use a logical model (UML classes) to explain the concepts of the domain and their attributes>

## 5 References and resources

<document the key components (e.g.: libraries, web services) or key references (e.g.: blog post) used that were really helpful and certainly would help other students pursuing a similar work>