Project 1 Requirements

Group 14
Tiago Carvalho fc51034
Diogo Lopes fc51058
João Roque fc51080
Miguel Saldanha fc51072
João Afonso fc51111

21/04/2021

1 Use Cases

Services	User	Functionalities
Normal	Regular	User Log in/Sign in See Book, Show and Movie Library Set Book/Show/Movie as seen Set Book/Show/Movie as liked Ask for suggestions to read and/or watch
	Admin	Add Book/Show/Movie to Library Remove Book/Show/Movie from Library
Spark	Regular	Count how many views a specific Item has Count how many likes a specific Item has

2 Preliminary Functional and Non-Functional Requirements

SPARK note: Since our likes and views are stored inside each User, Spark makes sense in this context because we need to go through every single one of them to calculate it. A different approach would be storing views and likes in the Item itself, but for this implementation suits well our needs.

2.1 Functional

- User should be able to browse Books, Shows and Movies.
- User should be able to search for a specific item by name, type and/or categories.

- User should be able to mark an Item as seen.
- User should be able to mark an Item as liked.
- User should be able to ask for suggestions.
- Admin should be able to add and remove content from the library.
- **(Spark)** User should be able to see how many likes and views a specific Item has.

2.2 Non-Functional

- Browsing Items shouldn't take more than 1.5 seconds to load.
- Searching shouldn't take more than 2.5 seconds loading the response.
- Marking Item as seen or liked shouldn't take loading time on Users' end.
- Suggestions shouldn't take more than 5 seconds.
- Data stored in cache shouldn't affect the system.
- Adding and Removing Items should persist on the database.
- **(Spark)** Calculating views and like from an Item shouldn't take more than 1 second.

3 Preliminary Architectural Design