<Re:Book>

Software Architecture Document

Version <2.0>

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| <27/11/2021> | <1.0> |  | <Dinh Dinh, Khoi Quach, Ngan Nguyen, Thu Nguyen, My Le> |
| <11/12/2021> | <2.0> |  | <Dinh Dinh, Khoi Quach, Ngan Nguyen, Thu Nguyen, My Le> |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[**Introduction**](#_heading=h.gjdgxs) **4**

[1.1 Purpose](#_heading=h.81u04fsw3wzz) 4

[1.2 Scope](#_heading=h.1gcgajbruydb) 4

[**Architectural Goals and Constraints**](#_heading=h.30j0zll) **4**

[Security](#_heading=h.pflzp2ptirjj) 4

[**Use-Case Model**](#_heading=h.1fob9te) **4**

[**Logical View**](#_heading=h.2et92p0) **5**

[Account:](#_heading=h.tyjcwt) 7

[**Deployment View:**](#_heading=h.3dy6vkm) **15**

[**Implementation View**](#_heading=h.1t3h5sf) **15**

Software Architecture Document

# Introduction

## 1.1 Purpose

This paper depicts several features of the system using a variety of architectural viewpoints. Its purpose is to document and communicate the system's most important design decisions.

## 1.2 Scope

This Software Architecture Document provides an architectural overview of the website Re:Book. The Re:Book is being developed by Group 5 to support reviewing and publishing books.

## 

# Architectural Goals and Constraints

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

## Security

The system must be secure, because we handle user related information.

Authentication: Login with username and password.

Password security: Passwords will be hashed and salted.

SSL: To prevent man-in-the-middle attacks.

+ **Language**

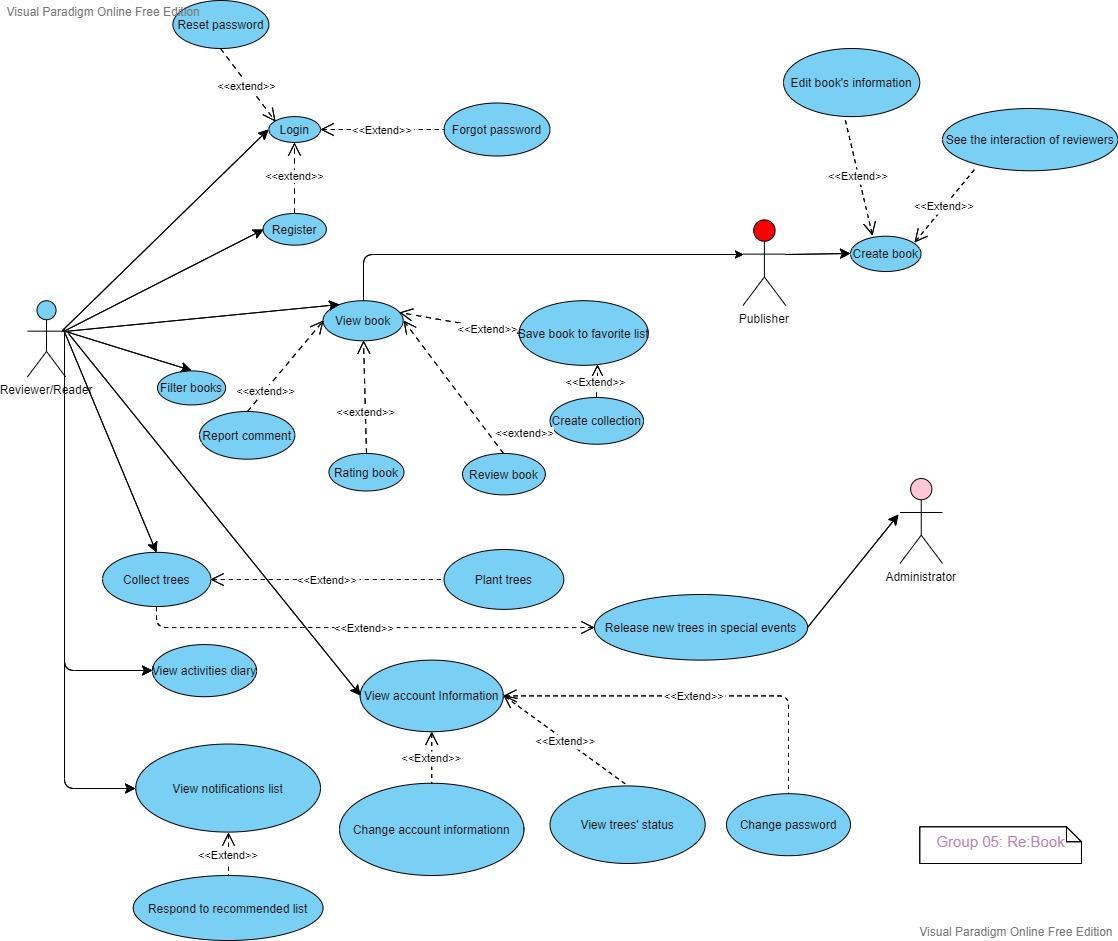
The system is able to deal with several languages.

+ **Persistence**

Data persistence will be addressed using a relational database.

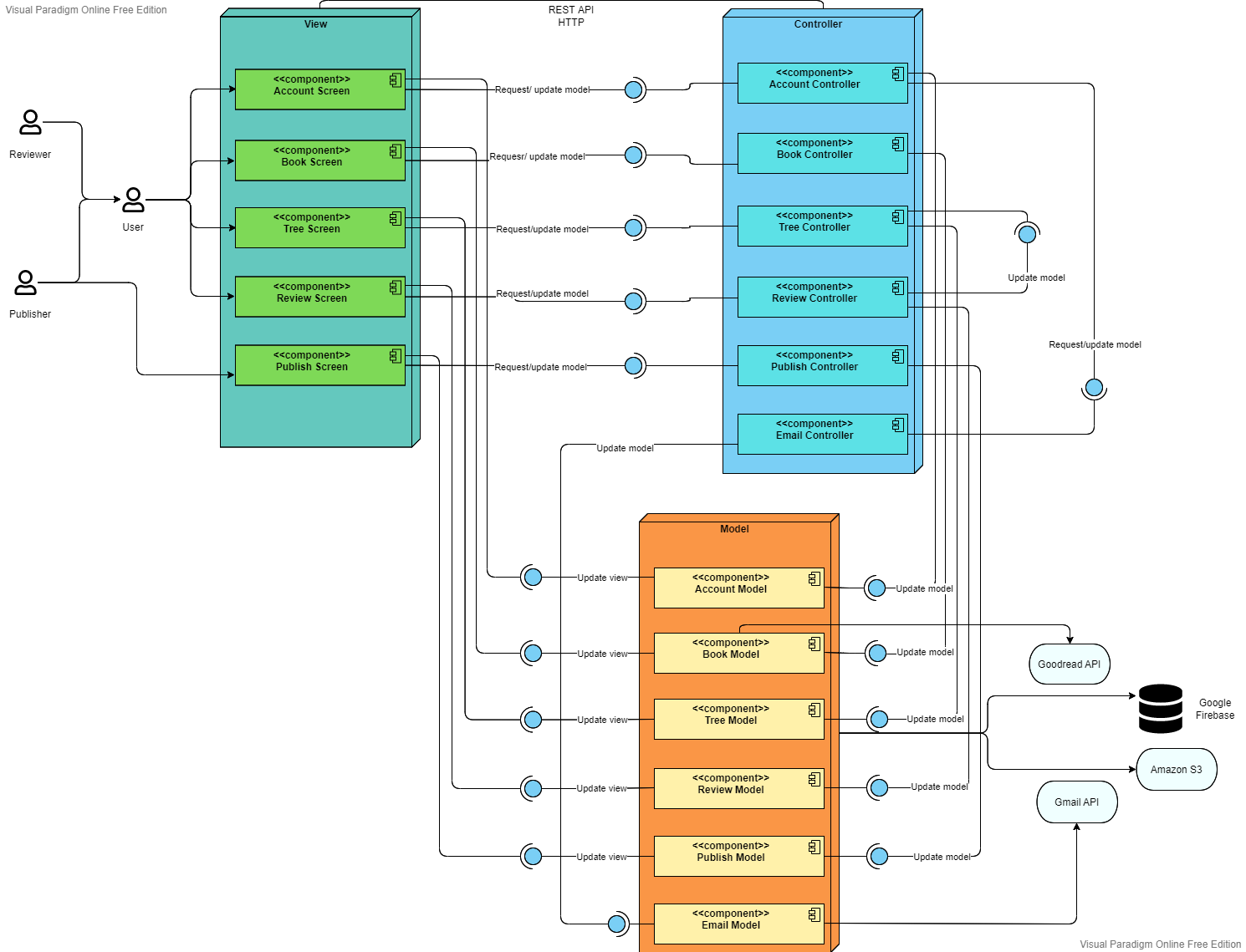
# Use-Case Model

<https://drive.google.com/file/d/10CaMWRM_T98fZHymYlrWNjY1z3_D5xou/view?usp=sharing>

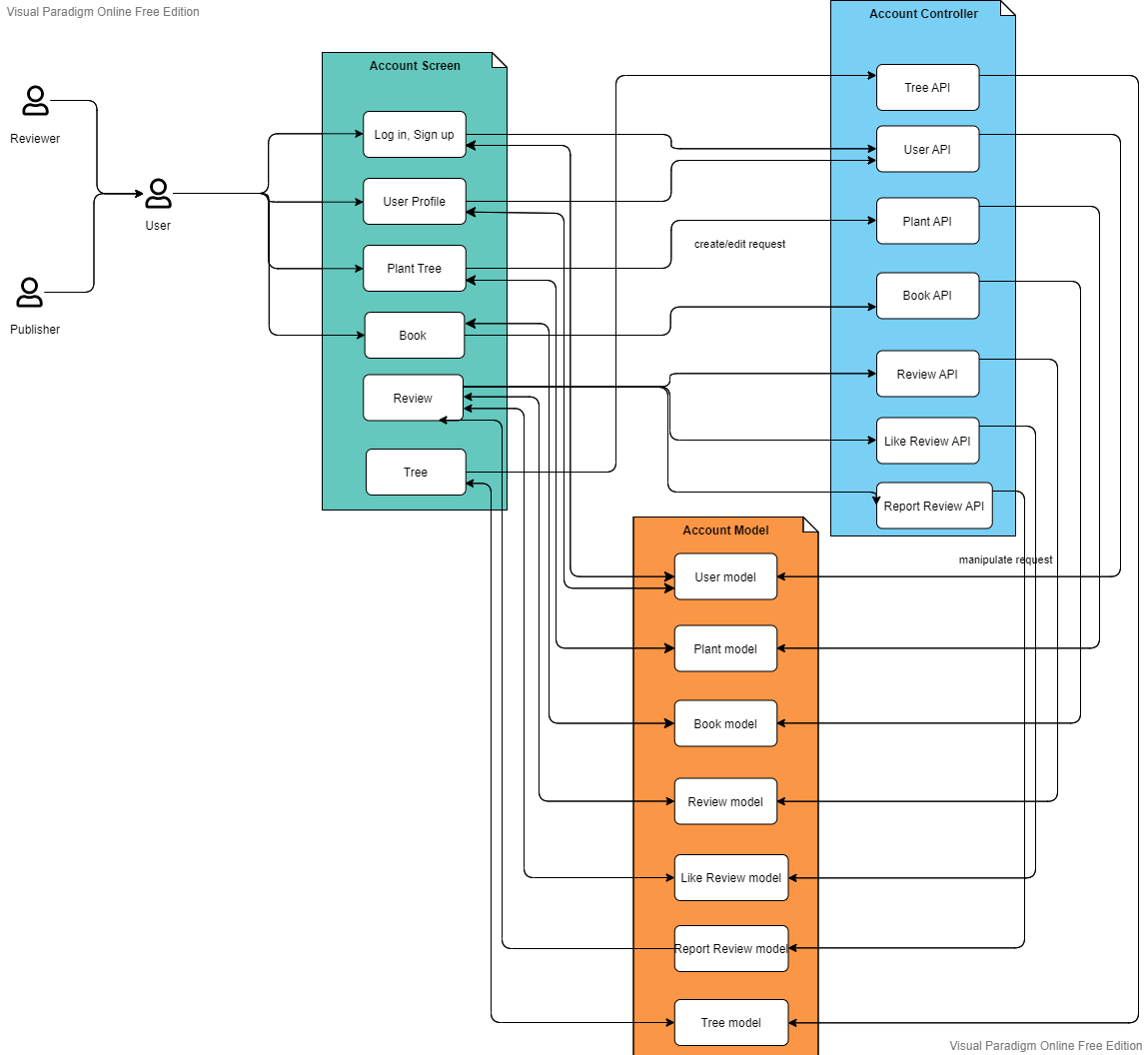


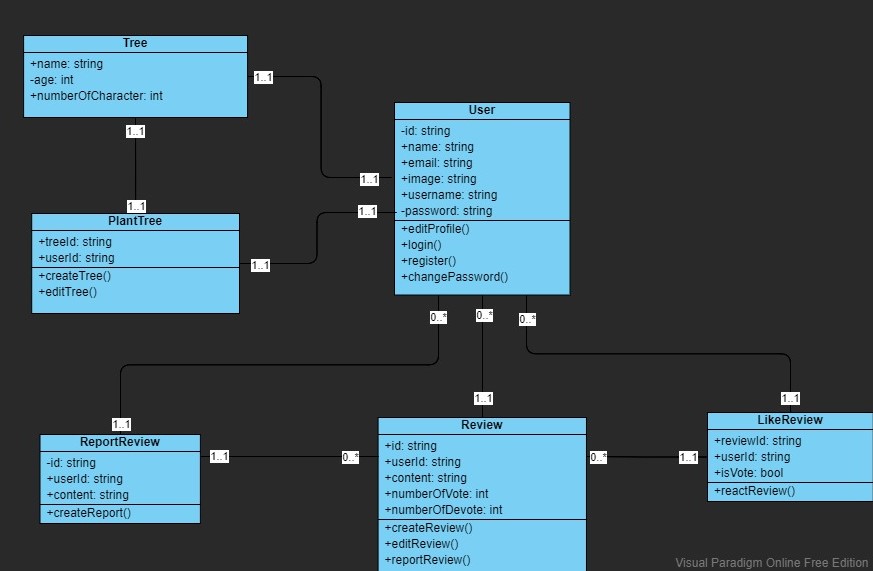
# Logical View

Component diagram for a general view:



## Account:





1. **Authentication**

Users are provided with some basic authentication functionalities, such as register, login, change password, etc,..

1. **User profile**

Users will have their own profile after they register successfully and they can edit their profile by updating information, changing avatar, etc,..

1. **Plant tree**

Users can create a tree when they write a review for the first time, and plant it. Moreover, they can also edit the appearance of their trees.

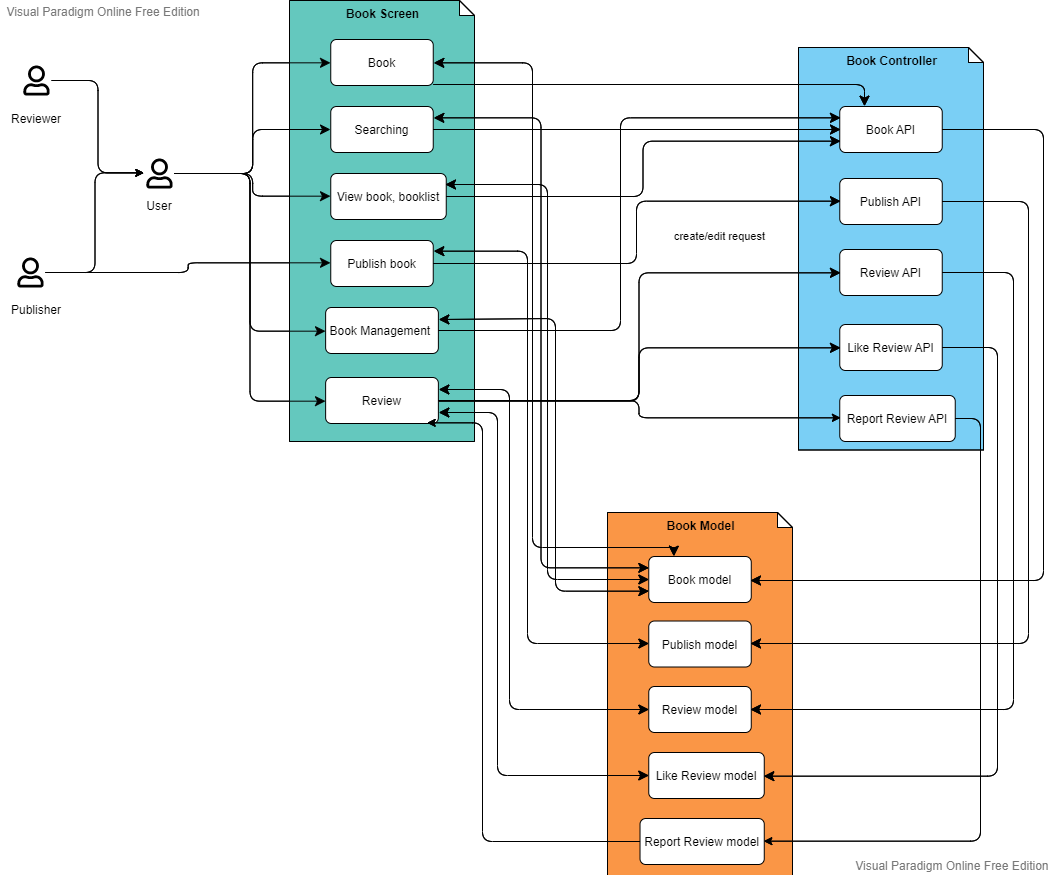
1. **Like review**

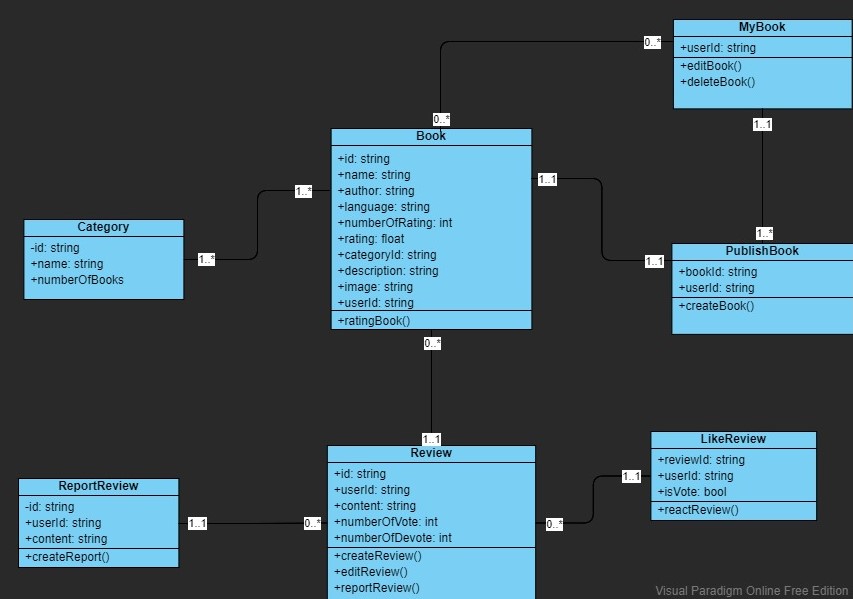
Users can interact with other reviewers by vote/devote their reviews.

1. **Report review**

If users see any reviews violating the community's guidelines, they can make a report.

* 1. **Book:**

****



1. **Book**

Book component allows users to view books, rating books, write reviews for books, create and publish books, etc,...

1. **My book**

My book contains all the favorite books that users save as well as the books that they create.

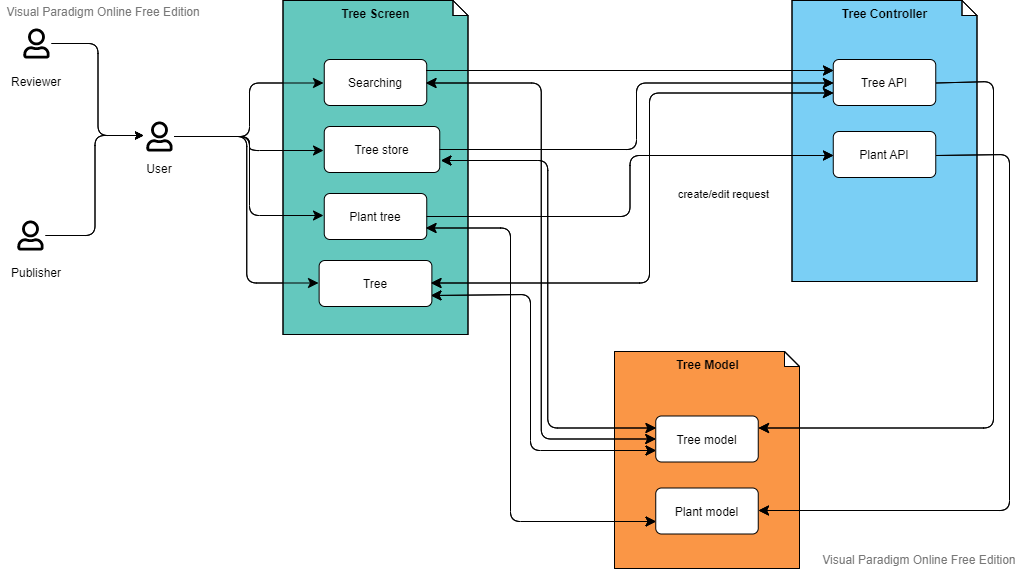
1. **Publish book**

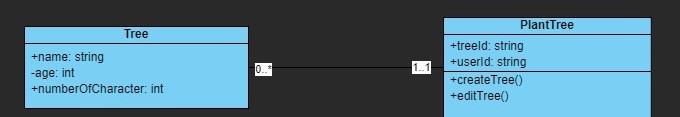
Book management component provides users with the functionality of creating a book if they cannot find a book that they want.

1. **Review**

Book component may contain none or many reviews. Users can write reviews for a book as well as interact with other reviews.

* 1. **Tree:**

****



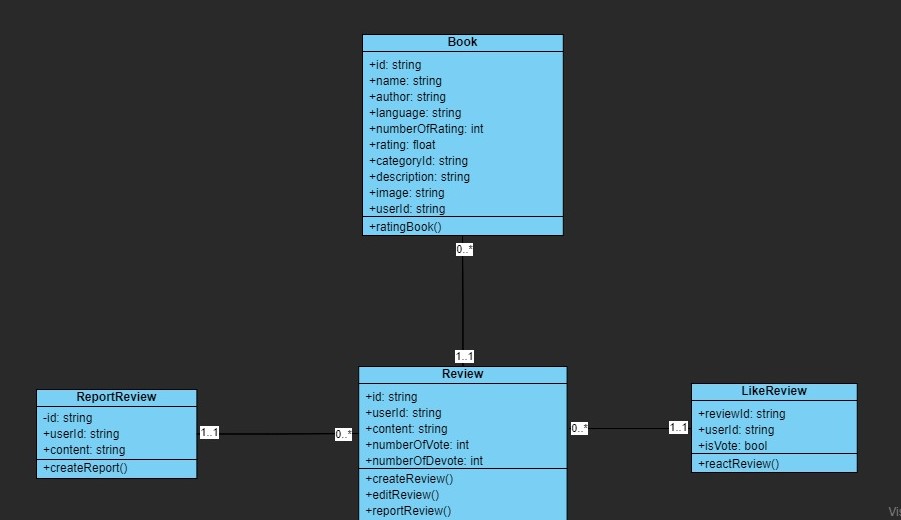
1. **Tree**

Tree management component contains many trees, in which the age of trees is based on the number of characters in the users’ reviews.

1. **Plant tree**

Users can create a tree after they write a first review for a book. The growth of trees is according to the users’ reviews. Furthermore, users can also decorate their tree if they want.

* 1. **Review:**



1. **Review**

Review management component may contain none or many reviews. Users can write reviews for a book as well as interact with other reviews.

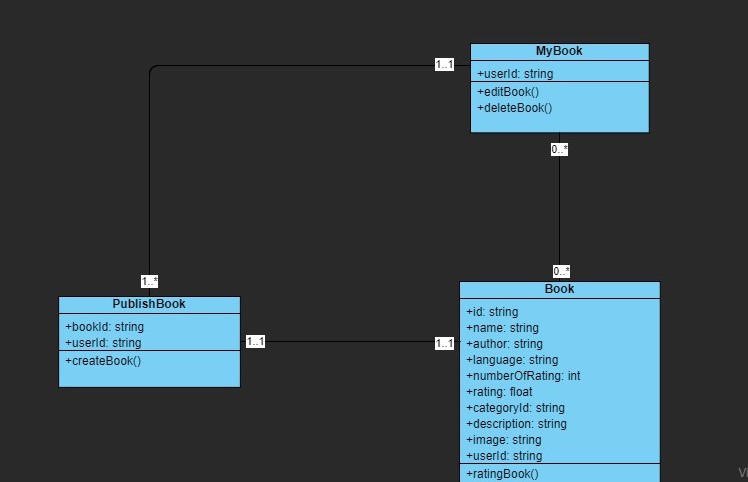
1. **Like review**

Review management component provides users with the functionality of interacting with other reviewers by vote/devote their reviews.

1. **Report review**

If users see any reviews violating the community's guidelines, they can make a report.

* 1. **Publish:**

****

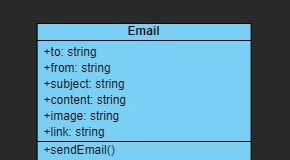
1. **Publish book**

The Publish book component provides users with the functionality of creating a book if they cannot find a book that they want.

1. **My book**

My book contains all the favorite books that users save as well as the books that they create.

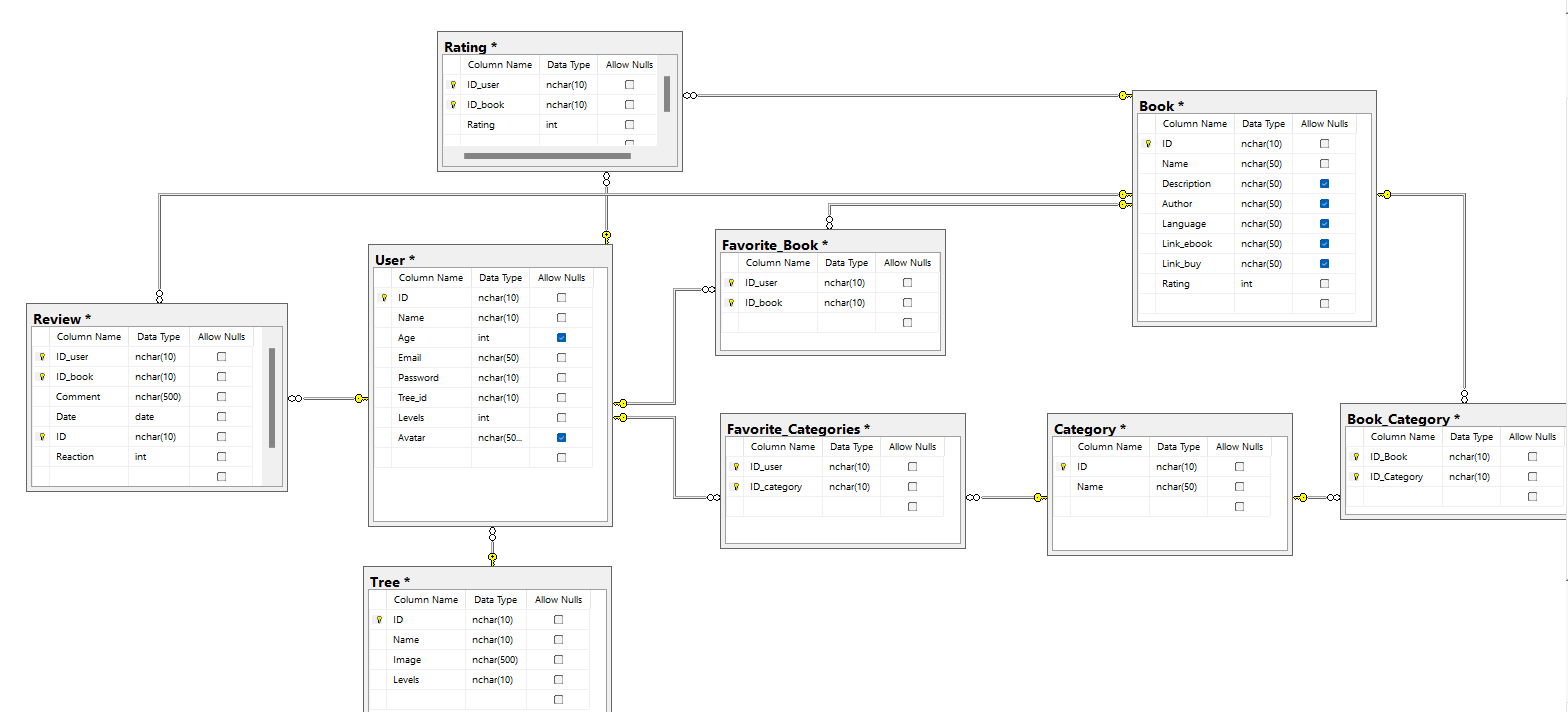
* 1. **Email:**

****

The email component allows the web server to interact with the users by suggesting them recommended books by email. The users can only receive email if they provide the server their email address.

* 1. **Google Firebase:**

[**Database Diagram**](https://drive.google.com/file/d/1RqbldzPxJKMjLi9lnKvJ_i7P2hSmf26g/view?usp=sharing)

****

* Description: Google Firebase technology supports web applications, allowing secure access to the database directly from client-side code
* Data is saved in JSON format and synced in real time across all connected clients.

1. **Tree**

Tree entity contains the information about a tree, which are id, name, image, and levels/age. Id is the primary key.

1. **Book**

Book entity provides information about a book, including id, name, description, author, language, link to buy book or link e-book, and rating. Id is the primary key.

1. **Category**

Category entity includes information about book genre, which are id and name. Id is the primary key.

1. **User**

User entity provides information about an user, including: id, name, age, email, password, id of user’s tree, age of user’s tree, and avatar. Each user has their own tree so the foreign key of the user entity is tree\_id, which is linked to the id of the tree entity. Primary key of the user entity is user id.

1. **Book Category**

Book category entity is used to identify the category of a book, so that it contains book id and category id and both of them are also primary keys. In which, book id is a foreign key linked to the id of the book entity and category id is a foreign key linked to the id of the category entity.

1. **Favorite Book**

Favorite book entity is used to store the information about the favorite books of a user. It contains book id and user id, and they are primary keys. Moreover, they are also the foreign keys linked to the corresponding primary keys of user entity and book entity.

1. **Favorite Category**

Similarly, favorite category entity is used to store the information about the favorite categories of a user. It contains category id and user id, and they are primary keys. Moreover, they are also the foreign keys linked to the corresponding primary keys of user entity and category entity.

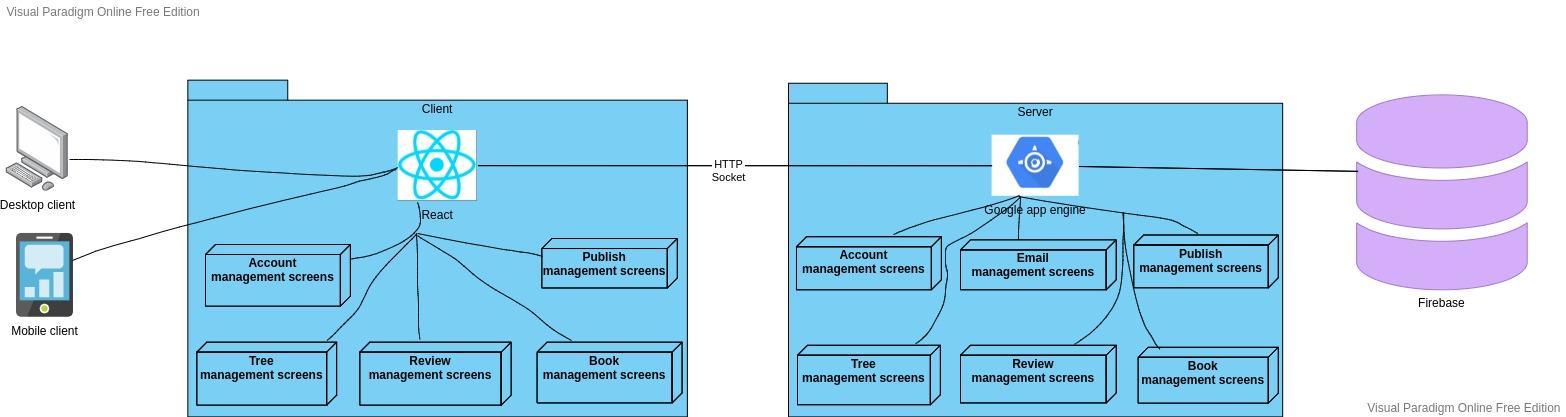
1. **Rating**

Rating entity provides the information about who user rates for a book, so it contains user id and book id, as well as the rating values. Both user id and book id are primary keys and they are also the foreign keys linked to the primary keys of user entity and book entity.

1. **Review**

Review entity provides the information about who user reviews for a book, so it contains user id, book id, review id, content and reaction. In which, user id, book id and review id are primary keys and user id and book id are also the foreign keys linked to the primary keys of user entity and book entity.

# Deployment View:



# Implementation View

The Rebook folder is used for online interaction, such as creating user interfaces. The model settings and certain other server operations are stored in the server folder.

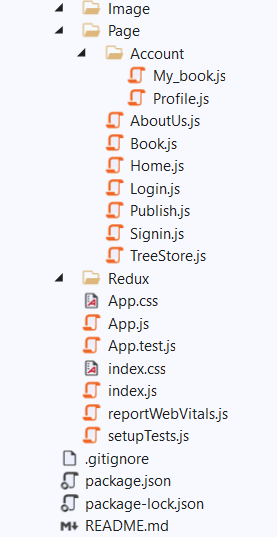
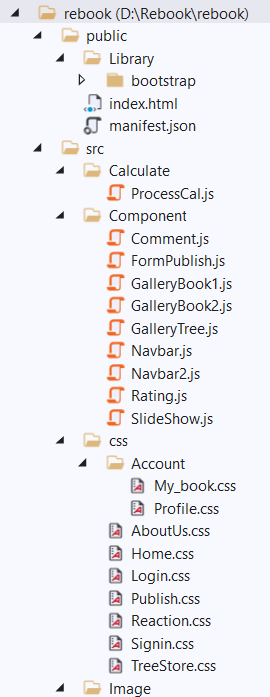
The Rebook folder contains public folder and src folder, with:

* public: storing libraries and index.html
* src contains these folders:
  + component: storing the common components shown throughout the website
  + page: storing many main page folders
  + redux: storing the states and api call, to make changes to the website view if needed
  + css: storing the style (e.g., fonts, colors, spacing) of the page
  + image: storing image

Server folder contains:

* common: contains configuration of models and their attributes.
* server contains 3 folders: authentication, book, and tree. They store some functions of the server.

Demonstration of Rebook folder:



Demonstration of Server folder:

