Development of an Electronic Book Software

1st Juan David Salcedo Sanchez
district University Francisco Jose de Caldas
Bogota, Colombia

2nd Carlos Santiago Gongora Ramirez
district University Francisco Jose de Caldas
Bogota, Colombia

Abstract—This document describes the process on the development of an Electronic Book Software (EBook). This includes diagrams, design patterns, and more useful tools to make this software friendly, optimized and well made

Index Terms—design patterns, software, development

I. INTRODUCTION

Since 1960's, the software development has been rapidly increase, the technologies, the services and also the demand quantity, making the jobs and even careers related to this field desired by young people and adults. The Software development is part of the majority of things of daily use, the apps on your phone, the GPS on your car, the registration page of your job, your bank account management, literally everything we use today is software. Software isn't only the writing of code or logic understanding, there's more topics to understand and apply to develop a project in a right way.

On the other hand, writing and publishing literature are activities that have been practiced since the beginning of what we know today as "paper." Also reading, being classified as a good habit, is included in the daily activities of a wide variety of people. But one of the main problems of this activity is the lack of support for the unknown writer, here is where the Software capabilities and book publishing come together to solve this big problem.

The development of a Software where you as a reader are looking to have a wide variety of books to choose from and can find new books to read, where you as a writer are looking for a place to post your books and get some recognition for your effort. This Software provides a lot of advantages, it has the capability to take books to everywhere, to continue writing your books and post them from anywhere to everyone in the world.

This document describes the technical development of an Electronic Book platform, from the base abstraction, the analysis of requirements, design and development. It's focused on the explanation of all of the technical aspects involved in the platform creation, and also stages like the initial conceptualization, the definition of the architecture, implementation of characteristics and functionalities and finally, tests and system debugging.

II. METHODS AND MATERIALS

- **Development Environment:** The operating system where all the project is performed is *Windows*, also the *IDE* used on the development is *Visual Studio Code* which has a large variety of useful extensions that helps with code documentation, auto-format code and simplify the manage and understanding of the version control tool *Git*, which helps in collaboration between developers and writing code.
- **Programming Languages and Technologies:** *Bootstrap* is the best choice to simplify the Graphical User Interface development, also *Fast API* to build interfaces in a fast and easy way. To develop the front-end part *HTML* and *CSS* provide the best customization options for a great interface development. And, for the back-end development *Python* provides an easier way to manage data.
- **Data Storage:** The database management system *PostgreSQL* it's an SQL database that works together with *SQL Alchemy* that is used to manipulate the saved data.
- **Development Methodology:** The agile methodology is great for developing quality software without focusing so much on documentation and more on the code, so in this case *Kanban* is useful for its *Kanban board*, showing tabs where are the prioritized backlog (to-do list), work-in-progress, validate and complete tasks.
- **Diagrams and models:** The diagrams that help understand the software architecture.

III. EXPERIMENTS AND RESULTS

There are metrics to measure the performance of the platform, those can be response time, speed, scalability, capacity, and application stability. To guarantee the good performance the majority of tests are unit test for each new functionality. Also, to improve the development, the use of design patterns is obligatory to reuse code and also to increase the scalability of the Software.

REFERENCES

[1] Pressman, Roger S. Software engineering: a practitioner's approach. Palgrave macmillan, 2005.
[2] Debrauwer, Laurent, and Fien Van der Heyde. UML 2.5: iniciación, ejemplos y ejercicios corregidos. Ediciones ENI, 2016.
[3] Fowler, Martin, and Kendall Scott. UML gota a gota: actualizado para cubrir la version 1. Vol. 2. Pearson Educación, 1999.