

|  |  |  |
| --- | --- | --- |
|  |  | |
|  |  |  | | |  | |
|  |  | System Design Document  Digital books library mobile application - BookWorm | | |  | |
|  |  | | | | |  |
|  | **Introduction**  This will be a document that will cover a several topics related to the digital book library mobile application, the objectives, the approach I will take, stakeholder identification etc. **The name of the solution that I will introduce is called BookWorm.**  Author: K.P.I. Shenesh Perera  Date: 21/06/2019  IDM | | | | |  |
|  |  | | |  | | |

**Contents**

[Problem Declaration 4](#_Toc14979276)

[Introducing the digital book library - BookWorm 4](#_Toc14979277)

[4](#_Toc14979278)

[Objectives and requirements of the application 5](#_Toc14979279)

[Objectives 5](#_Toc14979280)

[User Requirements 5](#_Toc14979281)

[System Requirements 5](#_Toc14979282)

[Stakeholders 5](#_Toc14979283)

[Research of similar applications 6](#_Toc14979284)

[Functionalities, structure and tools used for BookWorm 7](#_Toc14979285)

[Functions of BookWorm 7](#_Toc14979286)

[BookWorm – A progressive web application 7](#_Toc14979287)

[Tech Stack (Tools used) 8](#_Toc14979288)

[Project plan & Gantt Chart 9](#_Toc14979289)

[Pseudocodes & Flowcharts 10](#_Toc14979290)

[Searching for a book flowchart 10](#_Toc14979291)

[Ratelimitor flowchart 11](#_Toc14979292)

# Problem Declaration

Essentially, the main problem BookWorm attempts to solve is to allow the modern generation of technological book readers to read books at the comfort of their house rather than having to visit and spend time at an inconvenient place such as a library. The current libraries all have the following problems:

* Library memberships are required to take a book home for further reading. Depending on the region the library is, the membership cost maybe expensive.
* If in some case a book is damaged, a penalty cost has to be paid. This happens quite often as books are made of paper and paper is quite fragile.
* Most libraries do not provide allow their members to keep books with them for extended periods of time, almost all the libraries lend books only for 2 weeks. Exceeding this time period has a penalty, each day that passes the penalty increases.
* Libraries hold quite outdated books with outdated knowledge, some may even contain facts that are wrong.
* The newest books are usually not available, it takes at least a period of a year for a library to receive a copy of a new book, that too if the library is actually serious about the goal of a library.
* Libraries are counted as the “traditional” or “old-fashioned” way of finding or reading books, as such most of the libraries around the island have poor maintenance and support from the community.
* Unless the library is sufficiently large and has a lot of backing, it is very likely that the library will not sell books, which in other words means that it is impossible for you to have a book that you have a liking towards forever with you.

# Introducing the digital book library - BookWorm

# C:\Users\Ryft\AppData\Local\Microsoft\Windows\INetCache\Content.Word\bookworm.png

Figure 1.0, dribble.com, 05/22/2016, Patrick Davis

Bookworm, the solution to all the problems that a traditional library has. The name of this project was birthed by how my mother nicknamed me a “bookworm” when I was young. The project attempts to solve all the problems mentioned in the problem declaration, in a smart, creative and modern manner.

BookWorm is the one stop for all your digital book needs, it will function utilizing the GooglePlay Books API, grabbing the books that everyone loves to read from Google to present it in a much more friendly and better manner!

# Objectives and requirements of the application

The primary objective of BookWorm is to create digital library of books that can be easily read or even listened to as an audio track. BookWorm itself has a light-weight tech stack used for the application and attempts to cover all the objectives and requirements the business related problem dictates.

## Objectives

The identified objectives from the business related problem are:

* Provide a solution that will allow anyone across the globe to read a book online.
* The solution must be a mobile application
* The solution must be capable of finding, reading (both text and audio) or downloading books.

## User Requirements

The assumed/set into place requirements that the solution is expected to have are:

* A library of various types of books
* Ratings and reviews must be available
* The user interface must be appealing
* Books must be discoverable easily
* The user interface must not be difficult to use for anyone of any age group

## System Requirements

The assumed system requirements for the solution are:

* Must have a fast and presentative structure
* Users must not have to login and logout
* There must be no traffic monetization involved (Ad-free)
* Some solution has to be provided so that even readers who read at night can read properly
* The colors used must be appealing to users and preferably bright so that anyone who has sight problems will also be capable of seeing the application

BookWorm attempts to cover all these objectives and successfully reach the requirements of this business related problem. This document will describe the design of the application.

# Stakeholders

A stakeholder is an individual or set of individuals in an organization that is impacted by the outcome of a particular project. They either have a interest in the project or contribute to the project in some way (financially or otherwise) and can be part of the sponsors or the sponsoring organization of the project. Their influence can be viewed as both positive and negative depending on the project and the type of personality, expectations and requirements they put forth. However, they provide the necessary initialization to a project at times.

There’s lots of people who could be part of any given project all the way from the initial stage to the final stage. When I started this project I had to do a basic stakeholder analysis to identify any potential stakeholders that the business related problem dictates.

However, the business related problem does no such dictation of any stakeholders which became quite apparent on indepth analysis. As such, also given the fact that I worked on this project entirely on my own from design to deployment, the only such stakeholder I could mention indirectly could be Google.

I will describe soon, how I will use the GooglePlay Books API in order to grab books from Google’s database into BookWorm and display it to my potential users. I will also justify why I’ve done such a thing.

However, if there is anyone who will be impacted by the completion of BookWorm, it’s the book lovers out there that reads books day and night.

The inspiration to BookWorm comes from reddit.com’s r/books forum, flooded with book enthusiasts that regularly talk about their favourite books and how they grow quite tired of having to visit the local library to read certain books they can’t afford to buy.

# Research of similar applications

**Amazon Kindle** – Kindle is a software specifically to read a large variety of books, magazines and newspapers. It is one of the most ebook reader applications out there for both smartphones and tablets, mainly however there are Kindle tablets/specialized devices for the functioning of the Kindle software. As it is backed by one of the largest companies out there, the features Amazon Kindle boasts are quite rich. It allows you read, bookmark or highlight any book across any device. One of the most important features of Kindle is that it includes a dictionary that allows anyone to lookup the meaning of a word while reading. Google search and Wikisearch are also 2 other features that Amazon Kindle boasts.

**Aldiko Book Reader –** This is a perfect alternative to most famous paid book readers out there. It was specially designed for PDF and ePUB formats. It has some features that some other book readers don’t have, it doesn’t save books therefore it doesn’t take up a lot of memory. However it is not impossible for you to save the book, you just have to import it to your virtual bookshelf. Aldiko offers a native experience with lots of extensive features.

**NOOK** – Barnes and Noble, offers an ebook reader application for smartphones and tablets called NOOK, it is quite famous with over 2 million books, magazines, child’s books and newspapers with just a click away. It offers a several features that makes reading very easy and comfortable. One of the most distinct features of NOOK is that there is page synchronization, if you read a particular page in a book in your phone, you can goto your PC and expect that page to load up immediately. It is availbe on Android, iOS and on Windows.

**WattPad** – Wattpad is a very simple ebook reader, although not famous for it’s ebook reader functionalities it has quite the friendly and welcoming user interface. Getting an ebook is as easy as typing it in your browser or writing in the ISBN of the book. This app has some specific features that the previous apps do not have, user interface customizability. Wattpad allows you to set your background color or set an image as your background, change text colors or sizes. If you don’t have a lot of memory on your device, then you can download any book on WattPad in parts making it a great contender to the ebook readers.

**Kobo –** The Kobo ebook reader has a very unique feature that makes Kobo, Kobo. It introduces a concept called “Reader Life” that will introduce the ebook reader to the a new world of social reading. You area allowed to share quotes, notes and discuss books online with other ebook enthusiasts who share similar passions. Kobo allows you to read anytime at any place with over 4 million books, magazines and newspapers included. It is available in iOS, Android and Windows 8.

# Functionalities, structure and tools used for BookWorm

## Functions of BookWorm

Bookworm attempts to provide the necessary functionalities to satisfy user and system requirements by providing a feature rich user interface utilizing the Google Play Books Application Programming Interface, books will be taken from the google play books database and then be cached client-side of the user which will allow any user who wishes to browse through books to find books.

Any user that wishes to read the book may simply search and find the book or choose from one of the already recommended set of books, after which a link to the google play books storage api will be taken, so when you click on any particular book, the user will be redirected to a pdf version of that book.

It goes without saying that audiobooks are also capable of being discovered through bookworm, when you want to find a particular audiobook or wish to listen to a particular book in audio, the google play books api provides an interface to listen to the books.

The functionality of bookworm is however limited to the free books offered by Google Play, it is easily justifiable because books behind payment gateways or actually cost money have all rights reserved to the author, so if the author of that book declares that you must pay to read the book, BookWorm nor Google Play nor any company of the world has the right to go against it.

The objectives of the business related problem and the user/system requirements that I personally put to place are all achieved through this one project. However, what makes BookWorm great isn’t all these rich features, but the fact that it is going to be a PWA.

## BookWorm – A progressive web application

In 2015, Alex Russell introduced the term progressive web app online. He recounted a conversation between himself and Frances Berriman, in which they enumerated the attributes of a new class of applications based on the gradual and powerful evolution of modern browsers. Those attributes are, verbatim:

* **Responsive**: to fit any form factor
* **Connectivity independent**: Progressively enhanced with Service Workers (part of ES6 JS ) to let them work offline
* **App-like-interactions**: Adopt a shell and have a content application model to create app-like navigations & interactions
* **Fresh**: Transparently always up-to-date thanks to the Service Worker update process
* **Safe**: Served through TLS/SSL to prevent snooping which means HTTPS is compulsory
* **Discoverable**: Are identifiable as applications utilizing the W3C Manifests and Service Worker registration scope allowing search engines to find them
* **Re-engageable**: Can access the re-engagement UIs of the OS; e.g. Push Notifications
* **Installable**: to the home screen through browser-provided prompts, allowing users to keep apps they find most useful without the hassle of an app store
* **Linkable**: meaning they’re zero-friction, zero-install, and easy to share. The social power of URLs matters.

So basically what a Progressive Web Application is a application that brings the best of the browser world and the mobile app world into one single application, making it possible for the application run and behave like a mobile application in a mobile and a web application in a browser.

This approach cuts down the strenuous process of developing applications with full native tools like Xamarin, Flutter etc. for mobile apps and then have no website or something for a user to use on a PC.

Bookworm is also a Progressive Web Application, it is the first PWA that I will be writing and I believe bringing the world of the mature web technologies into the world of mobile technologies will help prove increase the notion of “Write once, use anywhere”.

## Tech Stack (Tools used)

* HTML – For general semantic markup
* CSS – For styling semantic markup to make it both responsive and mobile-first
* Javascript – To declare application logic, interact with the document object model and to fit the needs of scripting within this application.
* Workbox ServiceWorker – This allows you to work well with progressive web applications, it helps you to cache static assets like text data or image data so that a device does not have to frequently request for them.
* Fetch API – The HTTP cross-origion request API that comes with Javascript for the client side, this will be what’ll be used to interact with the Google Play Books API.
* Google Play Books API – Will provide all necessary meta information of books and will allow a user to discover new books to read. The Fetch API will be used to send forth and receive data from and to the application.
* Visual Studio Code- The Code Editor that I will use for the process of development, it has great tools like inbuilt debugging, a live server that allows you to serve static files, auto code completion, Emmet, IntelliCode, code highlighting and much more.
* Netlify – The static asset serving service, that provides a domain for a git repository that contains any static documents with TLS/SSL.

# Project plan & Gantt Chart

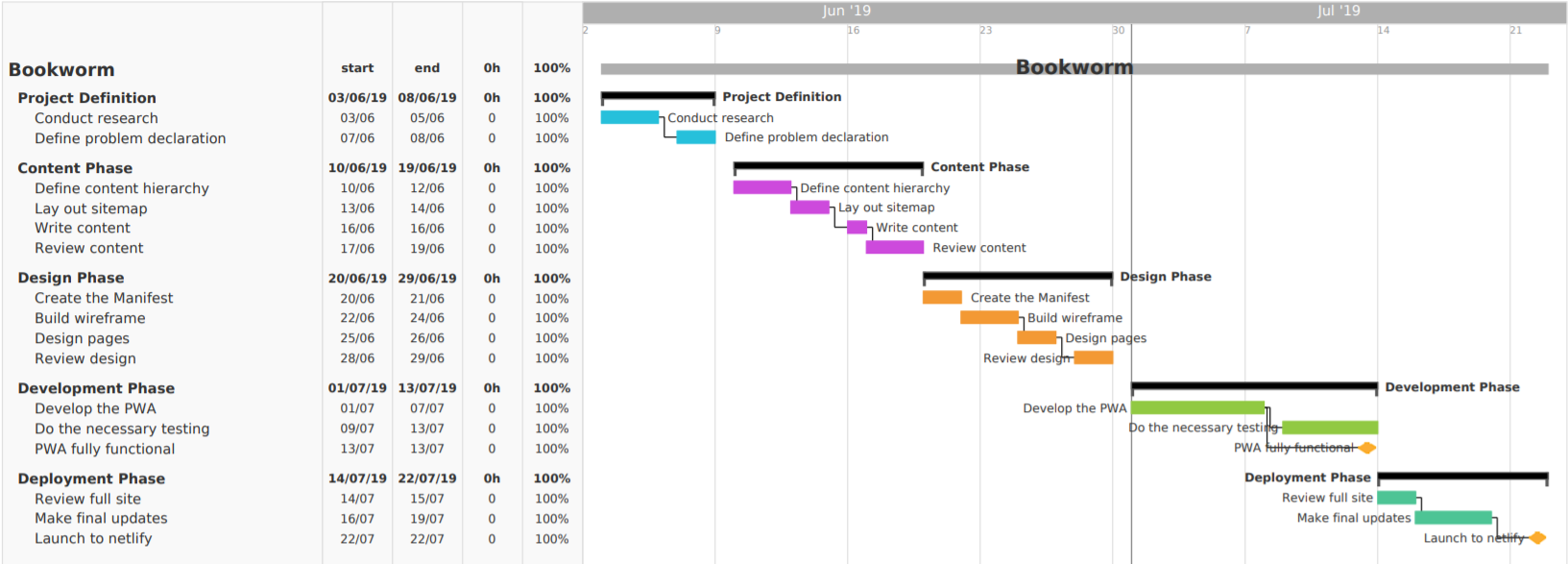


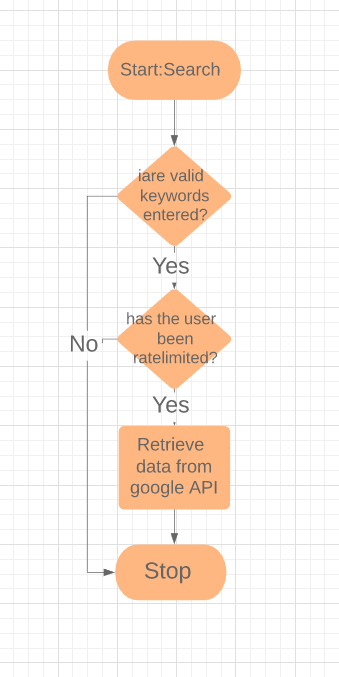
Figure 1.1, 03/06/2019, Shenesh Perera

This ambitious project began on the 3rd of June 2019, and ended successfully on the 22nd of July 2019. It is currently hosted at <https://bookwormz.netlify.com/>, for the public to see and interact with it. There were tremendous difficulties to face as this was my first PWA, although I’m not new to web development it was my first time working with service workers, the fetch API and what not. The documentation will declare how I faced these problems.

# Pseudocodes & Flowcharts

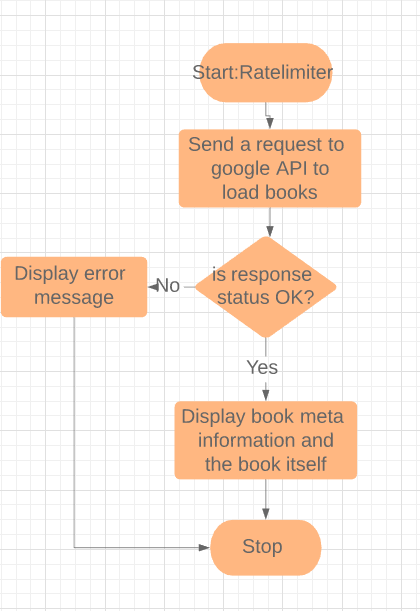
## Searching for a book flowchart

The following flowchart illustrates what the application does in order to take the keywords input into the search field and return meta information if the ratelimitor has not denied access to the search function. If it passes the ratelimiting check, then the user receives book information.



## Ratelimitor flowchart

The ratelimiter is a part of the application that makes sure that the user doesn’t make too many requests which would result in the application’s HTTP requests to the Google Play API to be rejected. It does this by checking if the response status message is OK and the code is 200, when a given HTTP request is sent. If the status code is not 200, then an error message is displayed if not, the HTTP request is successful and the meta information of the book will be displayed.



Ratelimint Pseudocode

