

E-book

System Requirements Specification

Version: 0.1

Document Serial Number: 001

Prepared by Nasr Eldin Mohamed

Table of Contents

Table of Contents.....	2
Revision History.....	3
1. Introduction	4
1.1 Purpose.....	4
1.2 Definitions & Acronyms.....	4
1.3 Use Cases and Stories Index	4
2. Project Overview	5
2.1 Business Objectives.....	5
2.2 Scope of Work.....	5
2.3 Product Perspective	5
2.4 Out of Scope	5
3. Actor Catalogue	6
4. Application Non-Functional Requirements.....	6
5. Application Functional Requirements	7
5.1 Use Case Diagram	7
5.2 Use Cases.....	8
5.3 Application Messages.....	10

Revision History Log

Version	Creator	Issue Date	Changes
0.1	Nasr Eldin	11 Jan 2020	Initial draft

1. Introduction

1.1 Purpose

The Purpose of this document is to present Software requirements specifications for E-book service. This document is intended to be used as a reference during the development, implementation and testing of this Test case.

1.2 Definitions & Acronyms

Term\Acronym	Definition
E-book	New test app allows getting a list of electronic books.
MVP	A minimum viable product (MVP) is a development technique in which a new product or website is developed with enough features to satisfy early adopters.

1.3 Use Cases and Stories Index

Sprint	Module	Use Case	Stories
Release 1			
Sprint 1			
		[UC-001] Books List	
		[UC-002] Book Details	

2. Project Overview

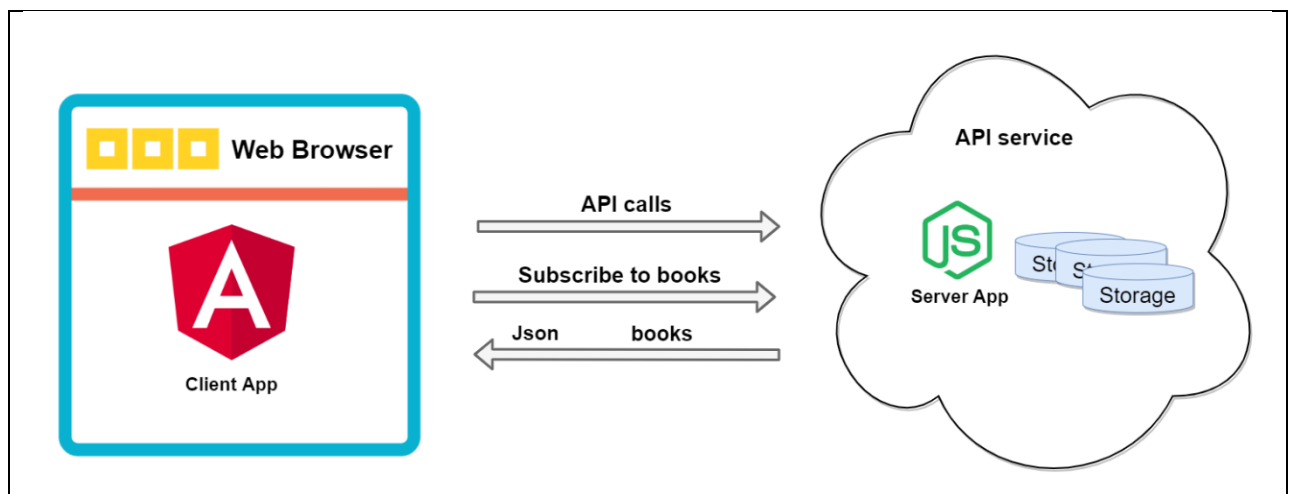
2.1 Business Objectives

We Need to investigate or list the top 10+ favorites book. So, we write an MVP test application to get a list of the book.

2.2 Scope of Work

Display books in the table UI and can search, filter, sort books with load more by scrolling.
View book details in popup modal.

2.3 Product Perspective



2.4 Out of Scope

- Any services not mentioned within this document.

3. Actor Catalogue

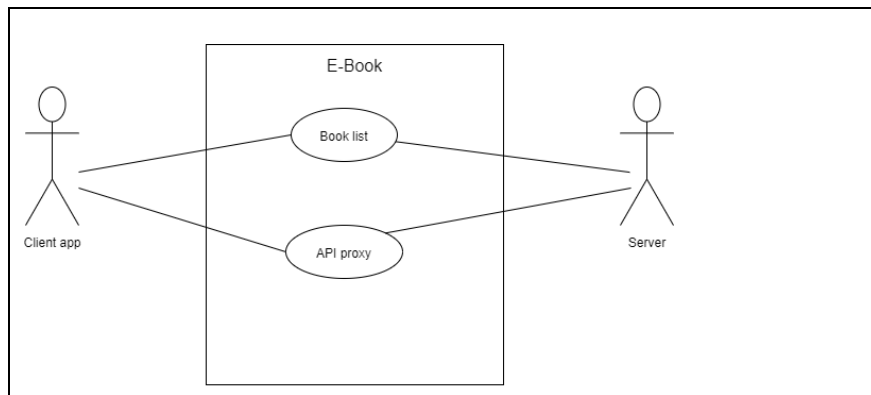
Id	Actor Name	Brief Description	Characteristics
1	Client app	Client app connect to eth network throw web3 to call smart contract and subscribe to events	System
2	Server	Server endpoint host for API service	System

4. Application Non-Functional Requirements

ID	Dependency
1.	App should offer bullet-proof security mechanisms for access.
2.	App should have great user-experience.
3.	Asynchronous manner should be used for long-time processes.
4.	Great user-experience <ul style="list-style-type: none">• Ease of access• Simple & short steps to view books• Excellent performance• Totally secured and bullet-proof against security threats
5.	Performance: <ul style="list-style-type: none">- Load app settings and proxy config within initialize app.-
6.	Usability: <ul style="list-style-type: none">- App should support by Chrome.- App should support auto responsive

5. Application Functional Requirements

5.1 Use Case Diagram

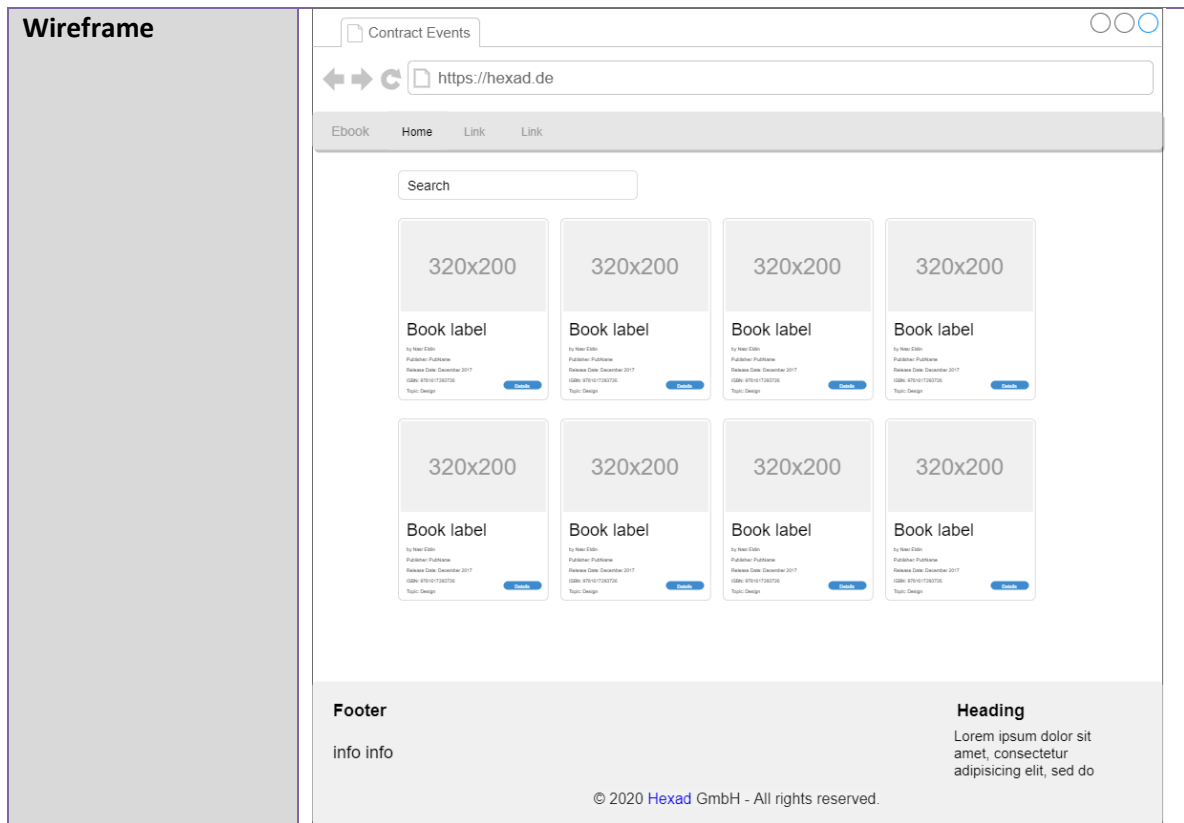


ID	UCD-01	Name	ENS Contract Events
Client app interact with API and get books list as json.			
Remarks:			

5.2 Use Cases

[UC-001] Books List

Use Case ID	UC-001		
Use Case Title	Books List		
Description	This use case to explain how the app interacts with service to allowed get books data and display to the user as a list of favorite books.		
Primary Actor	Client app		
Secondary Actors	User		
Assumptions	- Service return books		
Triggering Event	Load books list in initialize app		
Pre-Condition(s)	<ul style="list-style-type: none"> - App call backend service and return with books in json - Provide endpoint and provider API name 		
Post-Condition(s)	<ul style="list-style-type: none"> - View of services details - map return values properties to book model 		
Included Use Cases	N/A		
Extended Use Cases	N/A		
Related Business Use Case (BRS/BC)	N/A		
Related Non-Functional Requirements	- Application Non-Functional Requirements		
Logging Requirements	N/A		
Priority	High		
Basic Flow	#	Step	Other Flows
	1	User open App and click on Home list [books list] (app load books in initialize app automatic)	
	2	App display list of books	
Exception Flows			
Activity Diagram	<div style="text-align: center;"> <p>Book Entity</p> <pre> <<Interface>> Book + _id: int + isbn: String + title: String + author: String + description: String + publisher: String + rating: String + topic: String + cover: String </pre> </div>		



[UC-002] Book Details

Use Case ID	UC-002
Use Case Title	Book Details
Description	This use case to explain how to view/display book details.
Primary Actor	Client app
Secondary Actors	User
Assumptions	- display book details
Triggering Event	Load books list in initialize app
Pre-Condition(s)	- App call backend service and return with book details in json - Provide endpoint and provider API name
Post-Condition(s)	- View of services details - map return values properties to book model
Included Use Cases	N/A
Extended Use Cases	N/A
Related Business Use Case (BRS/BC)	N/A
Related Non-Functional Requirements	- Application Non-Functional Requirements
Logging Requirements	N/A
Priority	High

Basic Flow	#	Step	Other Flows
	1	User open App and click on Home list [books list] (app load books in initialize app automatic)	
	2	App display list of books	
Exception Flows			
Activity Diagram	<div style="text-align: center;"> <p>Book Entity</p> <pre> <<Interface>> Book + _id: int + isbn: String + title: String + author: String + description: String + publisher: String + rating: String + topic: String + cover: String </pre> </div>		
Wireframe			

5.3 Application Messages

MSG ID	Error Type	English
MSG001	Error	Could not load file 'json file name'
MSG002	Error	error occurred
MSG003	Success	successfully connected.
MSG004	Alert	No data found.