

1) Introduction

Purpose of Document:

This is a Requirements Specification document for a new Online EBook Maker software. E-books have made it easier for users to access books digitally without having to go to the library or any local book store. They have also revolutionized the way users interact with them by audio, editing, and creating your own from scratch. This software will have an easy experience for its users to create/edit their own Ebook for free. This document describes the scope, objectives and goal of the new Online EBook Maker. In addition to describing non-functional requirements, this document models the functional requirements with use cases, interaction diagrams, and class models.

Project Summary:

- **Project Name:** Online EBook Maker
- **Project Contributors:** Sergei Gabrielov, Nereo Hoill, Md Hossain, Warda Qadeer, Fabian Quijano
- **Project Users:** Emre Yilmaz, any user

Background:

A couple of reasons why we chose this project were that since E-books are more prone to piracy attacks than printed copies, we wanted to have a secured website. We also wanted to create a great experience for the users if they are not proficient with E-books. This Online E-Book Maker Project will allow users to create E-Book for free. This software will have Admin Login and Author Login. This system will allow the authors to make their E-Book without any problem and without any charge. Author will have to register if new member in the system. And existing member can Create new book or resume with old book. One Author cannot keep more than 3 incomplete Books, he has to complete 1 book at least to start a new one. Book will be created on basics of Title, Context, No of Pages in books.

Project Scope:

The scope of this project is to program a website in either C++ or Python to implement login, table of contents, conversion, etc.

We will divide each task into functions/methods and assign them to each team member. We will then meet on a weekly basis to talk about our sections on how and why they will work with the rest of the code.

Modules:

Admin

- Accept request from author.
- Check Author Details
- Checked for finished Books
- Make E-Books
- Can Process request by mailing the E-Books to authors

Author

- Register
- Fill Details
- Login and Choose
- New E-book or Resume Existing
- Enter Context of book
- Can pause E- book after completing context
- And resume it.
- Can have- E-Book in Word Or PDF format.

Plan of work:

Week 1. Begin basic background research and gather information. Meet with author, and determine requirements artifacts. Inspected requirements artifacts.

Weeks 2. Produced analysis artifacts, and inspected analysis artifacts. Showed artifacts to client, who approved them. Produced software project management plan, and inspected software project management plan.

Weeks 3.. Produce design artifacts, inspect design artifacts.

Week 4. Implementation and inspection of each class, unit testing and documentation, integration of each class, integration testing, product testing, and documentation inspection.

Glossary:

E-book: an electronic version of a printed book that can be read on a computer or handheld device designed specifically for this purpose.

Author: a writer of a book, article, or report.

Administrator:a person responsible for running a business, organization, etc.

Reliability:the degree to which the result of a measurement, calculation, or specification can be depended on to be accurate.

Security. A password will be needed to use the product.

usability:the degree to which something is able or fit to be used.

PDF:a file format that provides an electronic image of text or text and graphics that looks like a printed document and can be viewed, printed, and electronically transmitted.

URL(Uniform Resource Locator):a location or address identifying where documents can be found on the internet.

Functional Objectives:

High Priority

- The system shall allow users to access a website interface in order to create and modify ebooks in PDF or doc/docx format.
- The system shall distinguish between registered authors and public viewers. The authors will have access to create, edit, and publish their e-books while viewers will have access to view published e-books.
- The system shall publish e-books by converting the information produced by the author into a PDF or doc/docx file
- The system shall allow authors to register, fill in their account details, login, start a new ebook, resume an existing ebook, enter book content, save and quit, load and resume, receive finished ebook in pdf or docx format
- The system shall allow users to change their passwords at login by sending a URL to the email with which they registered that contains the steps necessary to do so.

Medium Priority

- The system shall include basic text formatting such as font type, font size, bold, italic, underline, indentation, images, and spell checking in the editor interface
- The system shall allow authors to control who can view their e-books. Some may be available to the public, some to specific authors, some to a category of authors.
- The system shall allow authors to follow other authors and request access to view specific e-books.

Low Priority

- The system shall allow optional two-factor authentication at login

Non-Functional Objectives:

Reliability:

- The system shall be completely operational at least 95% of the time.
- Down time after a failure shall not exceed 12 hours.

Usability:

- An administrator should be able to login and view pending requests from authors within 20 seconds.
- An author should be able to access their most recently modified document from the homepage within 5 seconds.

Performance:

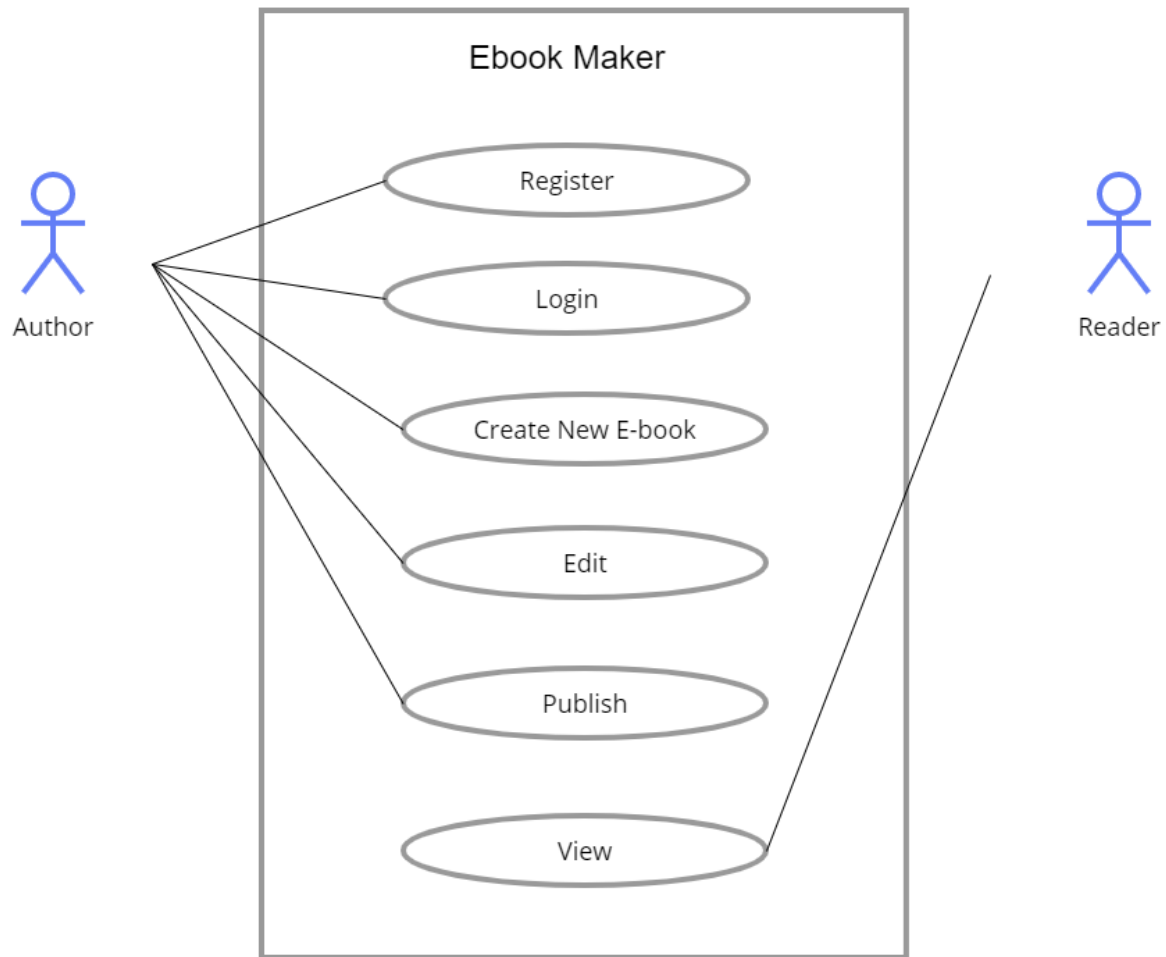
- The system should be able to support 1000 simultaneous users
- The mean time to view a web page over a 1 Mbps modern connection shall not exceed 3 seconds
- The mean time to load, view, or modify an ebook document shall not exceed 8 seconds

Security:

- The system shall use sha-256 authentication for password encryption
- The system shall never store plain-text passwords

Supportability:

- The system shall be designed to allow modification for creating ebooks of different formats in the future
- The system shall allow for new fields to be added in author account details without any significant restructuring or maintenance of the database



Use case name:	User Register
Summary:	In order to be able to access the application, create new books, edit, or publish a new book, a new user must register their account creating a username and password.
Basic Flow:	<ol style="list-style-type: none">1. The use case starts by indicating that the user wants to register for an account.2. The application requests for the new user to create a new username and password.3. User creates username and password4. App checks the database to make sure the username is not taken.5. App requests name, address, as well as an email address.6. User enters information.7. App determines access level.8. App determines what the user wants to try to do.9. App begins a session based on the functions the user wants to run.

Alternative Flows:	<ul style="list-style-type: none"> If the username already exists, the app will prompt you to enter a new one, repeat step two.
Extension Points:	<i>Start New E-book, Submit finished Ebook</i>
Preconditions:	None
Postconditions	The user can now access the main functions after registering
Business Rules:	

Use case name:	User Login
summary	Inorder to place order a user must log into the system so that the system can determine his access level.
Basic Flow	<ol style="list-style-type: none"> The use case starts when a user indicates that he wants to login. The system requests the username and password. The user enters his username and password. The system verifies the username and password against all registered users. The system starts a login session and displays a welcome message based on the user's preferences.
Alternativ Flows	<ul style="list-style-type: none"> If username is invalid, the use case goes back to step 2. If the password is invalid the system requests that the user re-enter the password. When the user enters another password the use case continues with step 4 using the original username and new password.
Extension Points:	none
Preconditions:	User is registered
Postconditions	The user can now obtain data and perform functions according to his registered access level.
Business rules	Some data and functions are restricted to certain types of users or users with a particular access level.

Use case name:	Create New E-book
Summary	The user will have the ability to create a new file supporting text and formatting that stores the contents of their e-book.
Basic Flow	<ol style="list-style-type: none"> 1. With an active login session, the user will click to create a new e-book document 2. The application will allocate space in the website database for the e-book metadata 3. The application will create a new file containing the e-book data 4. The application will present the user with a text area in which to type
Alternative Flows:	<ul style="list-style-type: none"> • If a fatal server error is encountered, the system will log the error, inform the user, and return the user back to the home page • If the user decides to quit before creating the new e-book, the process will be canceled and no information will be recorded into the application
Extension Points:	<i>Edit, Publish</i>
Preconditions:	The user is logged in as an author
Postconditions	The user can now access their new e-book in a list of currently in progress e-books within their account.
Business rules	The e-book can only ever be seen or modified by the author that created the e-book

Use case name:	Publish
Summary	The author will have the ability to publish a finished e-book which will convert the document to the desired format and then store the document in a list of published e-books accessible from the website.
Basic Flow	<ol style="list-style-type: none"> 1. The administrator will view a list of e-books that have been submitted for publishing. 2. The administrator will click on the e-book they wish to publish which will open a new page with options 3. The administrator will select options that decide how the e-book will be published including title, description, author, filetype(s) 4. The application will convert the ebook document along with related database information to produce a pdf or docx type file 5. The application will save the published e-book file as copy 6. The application will populate a page containing a list of published e-books with a link that allows a user to access them in their browser.
Alternative Flows:	<ul style="list-style-type: none"> • If a fatal server error is encountered, the application will log the error, inform the user, and return the user to the home page without

	creating any additional files on the server
Extension Points:	
Preconditions	The user is logged in as an author
Postconditions	Readers will now be able to access the published e-book through their browser from a publicly available list
Business rules	Only the author of the e-book will have access to publish

Use case name:	View
Summary	This use case allows readers to view published ebook.
Basic Flow	<ol style="list-style-type: none"> 1. The use case starts when a reader requests to view ebook 2. The system distinguishes between registered authors and public viewers 3. The system grants access to reader 4. Reader is able to view ebook
Alternative Flows:	None
Extension Points:	<i>View</i>
Preconditions	None
Postconditions	The reader will have access to the published ebook
Business rules	Only the author can control who views the ebook. Some may be available to the public, specific authors or to a category of authors.

Use case name:	Edit
Summary	The author will be able to access an existing e-book that has previously been started but not yet published. This will bring them to the e-book editor but populated with their already saved progress
Basic Flow	<ol style="list-style-type: none"> 1. The author will access a list of their in-progress e-books 2. The author will select the e-book they wish to edit from that list 3. The author may then select which version of the e-book they would like to load

	<ol style="list-style-type: none"> 4. The application will load the contents of the e-book and display the editor with the e-book data populated 5. The author will be able to modify the e-book in the editor 6. The application will periodically save work in progress that the author can access later if they need to recover work or revert to an older version 7. Once finished, the author will be able to save and quit which will update the e-book on the server with the new changes
Alternative Flows:	<ul style="list-style-type: none"> • If the author loses access to the editor without saving, they will resume at step 3 the next time they return
Extension Points:	<i>Publish</i>
Preconditions:	The user is logged in as an author
Postconditions	The author will have access to view the e-book they are working on again once they save and quit the editor
Business rules	Only the author of the e-book will have access to edit