DIGITAL BOOKS

IIHT

Table of Contents

1. Problem Statement

Build **DigitalBooks** app which takes traditional books a step further, combining text with visual and audio elements to make authors' publications truly multimodal. Authors can write down their thoughts and assemble a collection of original or curated content ranging from photos, drawings, and images to audio and video clips -- in some cases, even animated text.

And while **Digital Book** app can magically use images of autobiographical presentations or fantastical tales which are certainly options. It is also important to think beyond personal narratives to how authors might share the "stories/experiences" of their learning on any topic. And beyond author presentations and publications, plenty of students, teachers, doctors, engineers can jump on board, to create dynamic books and presentations that serve as instructional tools.

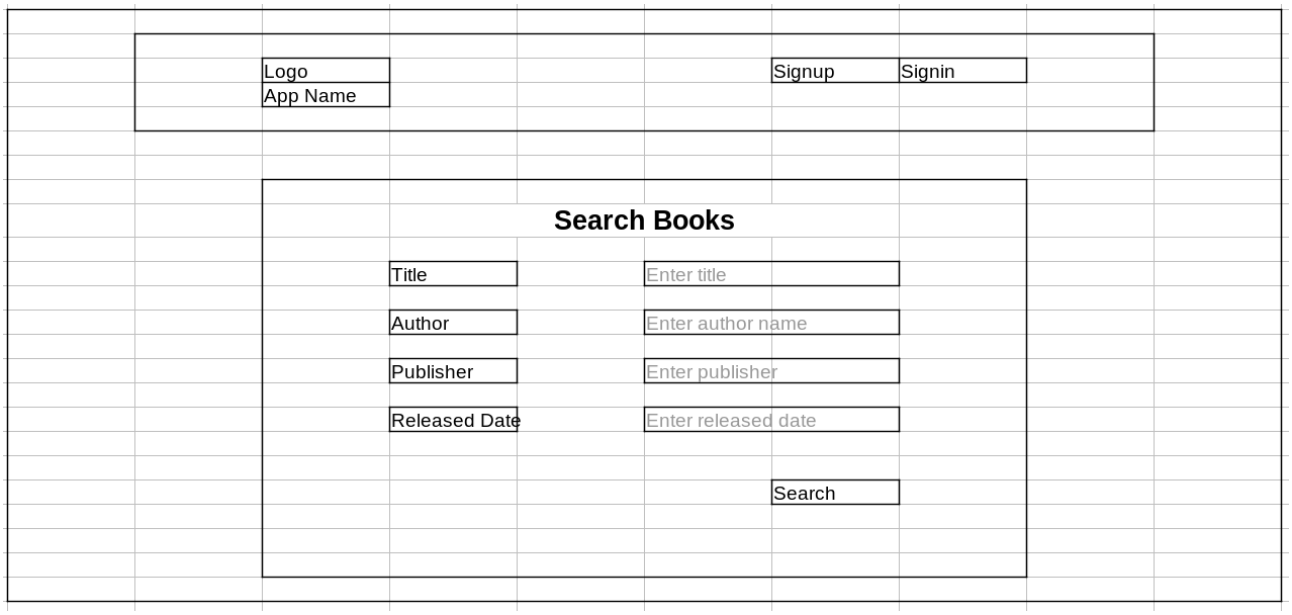
Build digital book app. Below are the different roles, which need to be supported by this software system.

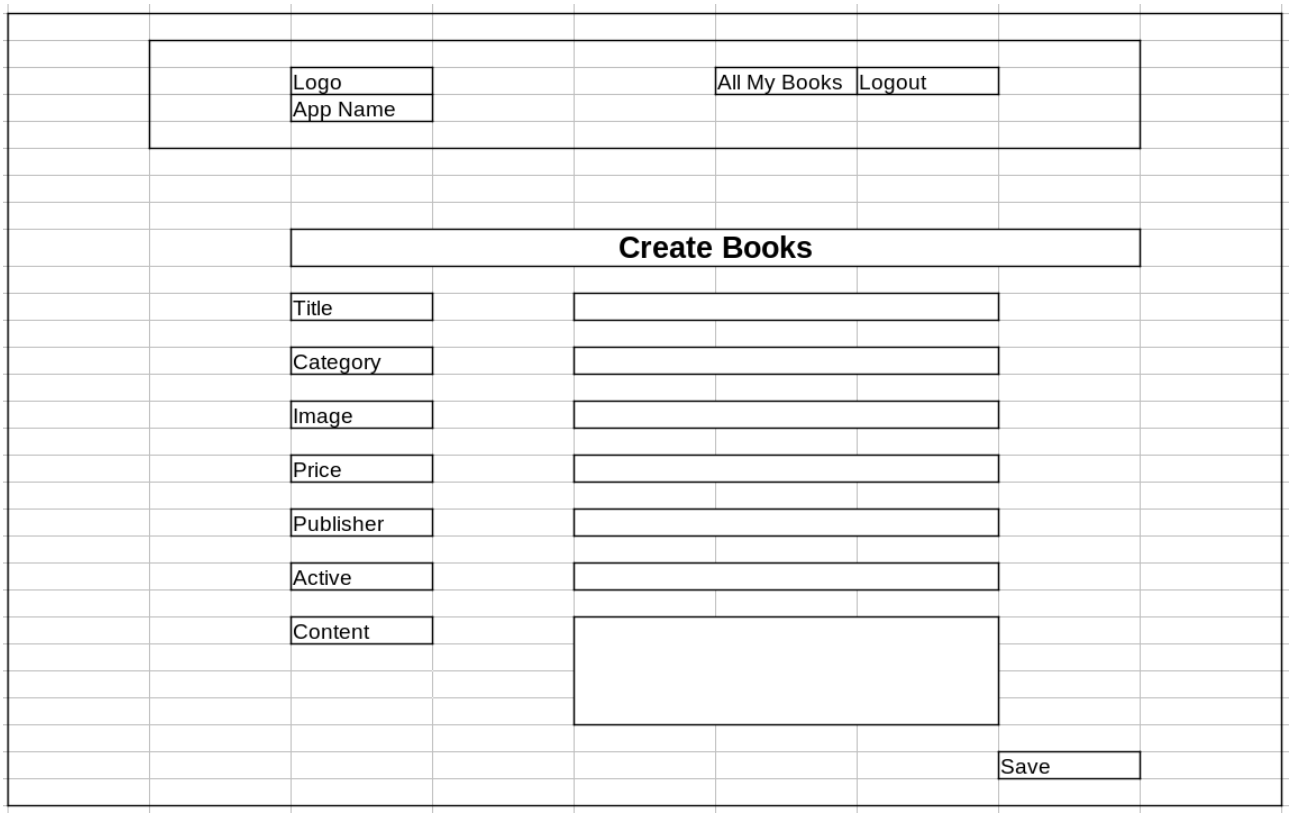
1. Author
2. Reader

The scope includes developing the application using tool chain mentioned below.

2. Wireframes

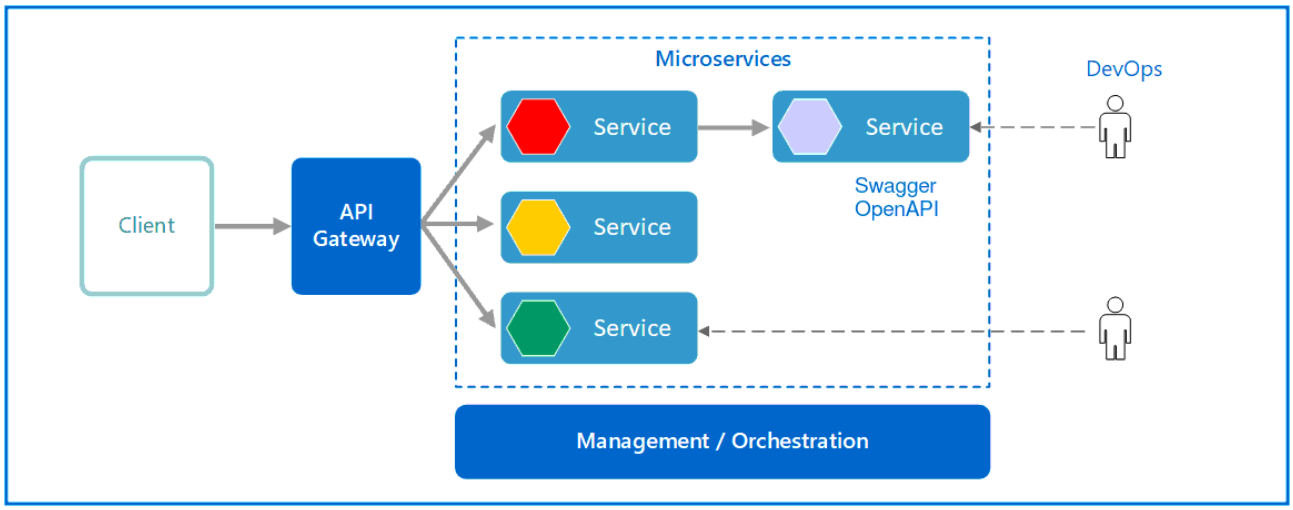
UI needs improvisation and modification as per given use case.





3. Application architecture

Compute and Integration/Presentation/Networking and Content Delivery



Possible rest clients

We will use below **clients** for our microservice applications

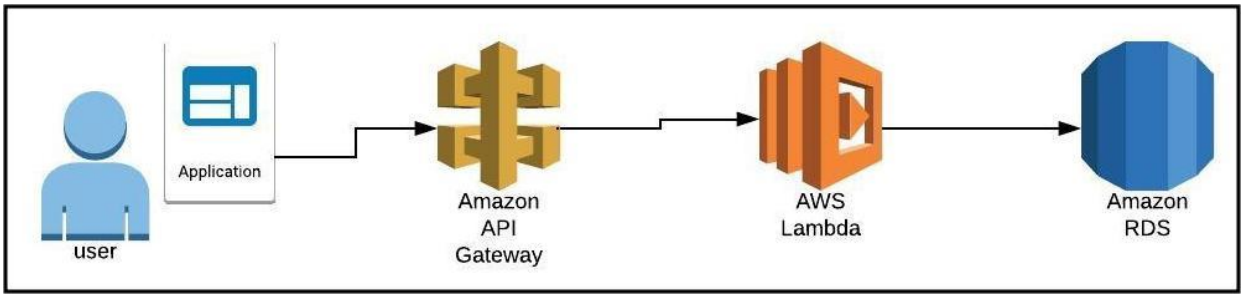
- ✓ Postman
- ✓ Postwoman (hoppscotch.io)
- ✓ Swagger
- ✓ Rest Client
- ✓ Angular app
- ✓ Java application

Any client from below list can consume our microservice (we will not use them):

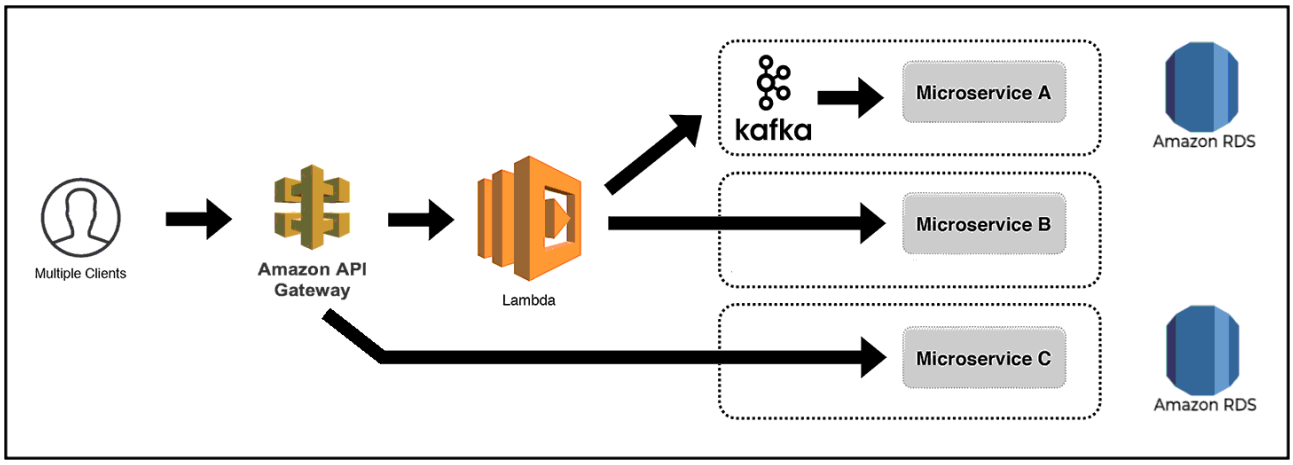
- ✓ React app
- ✓ Jmeter
- ✓ Android app
- ✓ Ios app
- ✓ .Net application
- ✓ python application
- ✓ many more…

4. Cloud Architecture

a. Api gateway → Lambda → RDS



a. Api gateway → Lambda → Kafka → EC2 → RDS
b. Api gateway → Lambda → EC2 → RDS
c. Api gateway → Lambda → EC2
d. Api gateway → EC2 → RDS



5. Tool Chain

Competency	Skill	Skill Detail
Engineering Mindset	Networking and Content Delivery	
	DevOps	
	Secure Coding	Veracode
	Code Quality	Sonar
Programming Languages	Application Language	Java
Products & Frameworks	Presentation	Angular
		Material/Bootstrap, rxjs, ngrx/store
		Javascript/Typescript
	Networking and Content Delivery	Zuul (SpringCloud)
	Security and Identity	OpenIAM
	Compute & Integration	Spring Boot
		Kafka
	Database & Storage	MySQL
	Governance & Tooling	Git
		Junit
		Mockito
Engineering Quality Platform	Cloud Tools	AWS EC2
		RDS-MySQL/Aurora
		AWS Lambda
		AWS API Gateway
		AWS ELB(Elastic Load Balancer)
		AWS CloudWatch

6. Business Requirements

As an application developer, develop microservices with below guidelines:

User Story #	User Story Name	User Story
US_01	Reader Mode	1. Reader can search for books based on category, author, price 2. Each search result need to display book logo, title, author, publisher, price, published date, category. 3. From search results, reader should be able to select a specific book and go ahead to complete purchase by providing **Name and Email ID** 4. On successful purchase, unique **payment_id** need to be generated. **Payment should be taken as a guest user only.** 5. Reader should be able to download invoice. 6. Reader can read only purchased books. 7. Reader should be able to • view history of all previous payments • view invoice using payment_id • ask for the refund within 24 hrs of payment
US_02	Author Mode	1. Author shall be able to create account. 2. Author shall be able to login/logout. 3. There can be pre-defined username/password for Author. 4. Author shall be able to add/edit book. For example a. **logo**: image b. **title**: Spiderman is back c. **category**: comic d. **price**: 24 e. **author**: current user name} f. **publisher**: Moon publisher g. **published date**: 22/04/2022 h. **chapters/content** i. **active:** true 5. Author shall be able to block and unblock an book. When book is blocked a. Books will not be shown in Search results. b. Readers (who have purchased book) should get notification about the unavailability of book.

7. Proposed Rest Endpoints

If you think rest endpoints need improvisation and modification as per given use case, you can make necessary changes.

GET	/api/v1/digitalbooks/books/search?category&author&price&publisher	reader can search books
POST	/api/v1/digitalbooks/books/buy payload: {bookId, reader {name, email}}	reader can buy book
GET	/api/v1/digitalbooks/readers/{emailId}/books	reader can find all purchased books
GET	/api/v1/digitalbooks/readers/{emailId}/books/{bookId}	reader can read book
POST	/api/v1/digitalbooks/readers/{emailId}/books?pid	reader can find purchased book by paymentId
POST	/api/v1/digitalbooks/readers/{emailId}/books/{bookId}/refund	reader can ask for refund within 24 hrs of payment
POST	/api/v1/digitalbooks/author/signup	author can create account
POST	/api/v1/digitalbooks/author/login	author can login
POST	/api/v1/digitalbooks/author/{authorId}/books	author can create book
PUT	/api/v1/digitalbooks/author/{authorId}/books/{bookId}	author can edit/block/unblock hist book

8. Key Rubrics (Expected Deliverables)

A. As an application developer:

a. Develop the application as a microservice architecture.
b. Ensure package Structure for project is like **com.digitalbooks.*** with proper naming conventions for package and beans.
c. Use **application.properties** or **application.yaml** file to maintain all spring boot config.
d. Implemented the package structure - Controller, Interface, Service, DAO, Testing, Validation, Security etc
e. Implementation as follows:
    I. Use Domain Driven Design to implement distributed architecture
    II. Follow Single Data Store per microservice practice
    III. Document REST endpoints with OpenAPI/ Swagger
    IV. Add CQRS pattern for Event Sourcing using Kafka for saving and retrieving book details, using Kafka (producer & consumer) topics
    V. Expose all rest Endpoints using a common API Gateway

B. Debugging & Troubleshooting

f. Generate bug report & error logs - Report must be linked with final deliverables which should also suggest the resolution for the encountered bugs and errors.

C. Code Quality/Optimizations

g. Associates should write clean code that is readable
h. Associate should use the Code Analyzer (PMD/SonarQube) to ensure code quality and standard code style.

9. Platform

- ✓ Use EC2 to deploy application on cloud.
- ✓ Use AWS RDS service as a database for the Application.
- ✓ Use AWS Lambda and DB to build a backend process for handling requests for Flight booking App.
- ✓ Use Amazon API Gateway to expose the Lambda functions built in the previous step to be accessible on public internet.
- ✓ Use AWS ELB(Network Load Balancer) to configure the load balancing of the instances

Note : Minimum 2APIs (UI+Backend) to be hosted in cloud

10. Methodology: Agile

- ✓ As an application developer, use project management tool along to update progress as you start implementing solution.
- ✓ As an application developer specify the estimation and planning as a part of Agile process.
- ✓ As an application developer, the scope of discussion with mentor is limited to:
  a. Q/A
  b. New Ideas, New feature implementations and estimation.
  c. Any development related challenges
  d. Skill Gaps
  e. Any other pointers key to UI/UX and Middleware Development