**Requirements Document**

**EduArchive**

**Team Data Access Geeks**

# Business Requirements

## Introduction

The objective of this web application is to offer a digital platform for learners to access reference materials on three subjects: art, mathematics, and technology. The platform will feature articles, biographies, and other relevant content, ensuring a comprehensive learning experience for its users. The name of our web application is ***EduArchive***.

## Statement of Problem or Need

### Problem

The client has indicated that there is a need for a centralised, user-friendly platform that provides quality learning resources on art, mathematics, and technology. Learners often find themselves sifting through multiple resources to find the information they require. This takes time and leads to fragmented learning.

### Stakeholder Concerns

* Concerns about the clarity and quality of content.
* Potential complexity in using the platform.
* The aesthetic appeal of the application.
* Privacy concerns regarding personal data.
* Content may be added/modified/deleted by unauthorised users.

### Affected People

* Students across various age groups and academic levels.
* Tutors seeking to provide quality reference materials.
* Administrators overseeing the platform’s learning content and functionality.

### Impact

* Inefficient learning due to scattered resources.
* Academic progress hindered due to lack of quality materials.
* Potential misinformation from unreliable sources.

### Unique Value Proposition

* A centralised platform offering comprehensive learning resources on art, mathematics, and technology.
* Intuitive user interface ensuring application is easy-to-use and content is effectively presented.
* The platform enables content to be easily added, modified, and deleted by authorised users.

## Business Requirements

### List of Stakeholders

* Students using the platform to search and view learning content.
* Tutors using the platform to provide reference materials for students.
* Administrators overseeing the platform’s content and functionality.

### Client

**Bilal Ishfaq** is the client for this project and the key contact for questions and reporting relating to this project.

### Business Constraints

* Web application and project documentation due by Friday 15 September 2023, at 11:59pm.

# Business Solution

## Options Considered

### Platform: Mobile Application

Developing the platform as a mobile application would mean that it would only be accessible via a mobile device and be platform-dependent, i.e., Android, iOS, or cross-platform. This would restrict the potential audience of users able to access the platform. Viewing content would also be generally on devices with smaller screen sizes.

### Platform: Web Application

Developing the platform as a web application would make it accessible via a web browser interface across a wider variety of devices, from desktops to mobiles. This would increase the potential audience of users. Users would also have a choice as to what type of device they wanted to use to access the platform’s content. Indeed, tutors and administrators would most likely be using desktops or laptops to interact with the application.

### Database: Relational (SQL)

Relational databases (RDBMS) that utilise SQL offer a structured and organised way of storing data, making it easier to establish relations between different datasets. This type of database supports complex queries, ensuring flexibility in data retrieval. One of the significant advantages is the ACID properties (Atomicity, Consistency, Isolation, Durability), which supports data integrity and reliability. However, they might require more upfront design and planning due to their rigid schema. Scalability can be a challenge, especially when dealing with large amounts of data or rapid growth. For simpler applications or those that don’t require complex relationships between datasets, an SQL database might be overkill.

### Database: Non-relational (NoSQL)

NoSQL databases offer flexibility and are schema-less, which can be advantageous for rapid development and iteration. They are designed with scalability in mind, making it easier to handle large quantities of data or high traffic loads. There are various types of NoSQL databases, such as document, key-value, column-oriented, and graph, each fitting different needs. However, they might not provide the same level of data integrity and ACID compliance as traditional RDBMS. Some NoSQL databases might not support complex queries or transactions as efficiently as SQL databases. Lastly, adopting NoSQL requires a different mindset and approach, which might present a learning curve for teams predominantly familiar with SQL.

## Recommended Solution

### Solution Statement

Students, tutors, and administrators are seeking a comprehensive and interactive platform to access, modify and manage learning resources. Unlike traditional learning resources that are fragmented and time-consuming to utilise, our platform provides a centralised web-based application dedicated to offering structured and categorised information on art, mathematics, and technology.

Our solution will ensure that users have a seamless experience, whether they are browsing articles by category, searching by keywords, or adding and modifying content. By leveraging the benefits of a web application and the flexibility of MongoDB, our platform will offer a cohesive, scalable, and efficient solution tailored to the diverse needs of its users.

Our recommended solution to develop a web application stems from the need for accessibility and versatility. Web applications are platform-independent, meaning users can access the platform from any device, be it a desktop, laptop, tablet, or smartphone. This ensures that the application is available to a broader audience without the constraints of device-specific applications. Furthermore, web applications offer easier updates and maintenance, ensuring that all users get the same experience without needing to download or update an app manually.

Our recommended solution is to utilise MongoDB (NoSQL) as the database solution for our web application. MongoDB is a leading NoSQL database that offers flexibility and scalability for our platform. Traditional relational databases, while structured, can sometimes be restrictive, especially when dealing with varied data types and structures. MongoDB’s document-oriented structure allows for varied data formats, making it easier to store articles with different attributes without the need for a rigid schema. This flexibility aids rapid development and iteration, accommodating future expansions or modifications to the platform. Additionally, MongoDB’s scalability ensures that as the user base grows, the database can handle increased traffic and data volume without compromising performance.

### Main Features

Three main features of the platform are as follows:

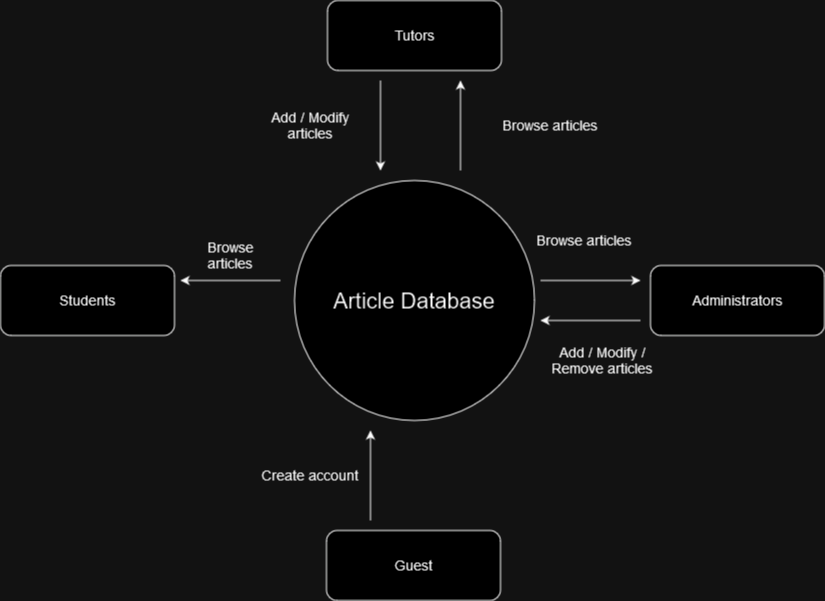
* **Authentication and Authorisation:** Users will be required to log in, ensuring secure access. Interactions with content will be determined by permissions associated with designated roles: student, tutor, and administrator.
* **Content Browsing:** Users can rapidly navigate through categorised resources or use keyword searches to pinpoint specific topics, ensuring efficient access to the platform’s repository of articles.
* **Content Management:** Tutors can add and update content, ensuring the platform’s learning resources remains fresh and relevant. Administrators have the added capability to remove content, maintaining the high quality of the platform’s reference materials.

### Unique value proposition / Justification

Our recommended solution should prove to be an excellent educational platform due to its combination of a universally accessible web application and the robust, flexible backend powered by MongoDB. Unlike other platforms that might be restricted by device compatibility or rigid data structures, our solution offers a cohesive and structured learning environment that can adapt and grow. The use of MongoDB means that as the platform evolves, adding new features or modifying existing ones will be seamless, ensuring longevity and adaptability. This combination of a user-friendly interface and a powerful backend database means that our learning platform will be a future-proof solution for diverse learning needs.

# Solution Requirements

## Context Diagram

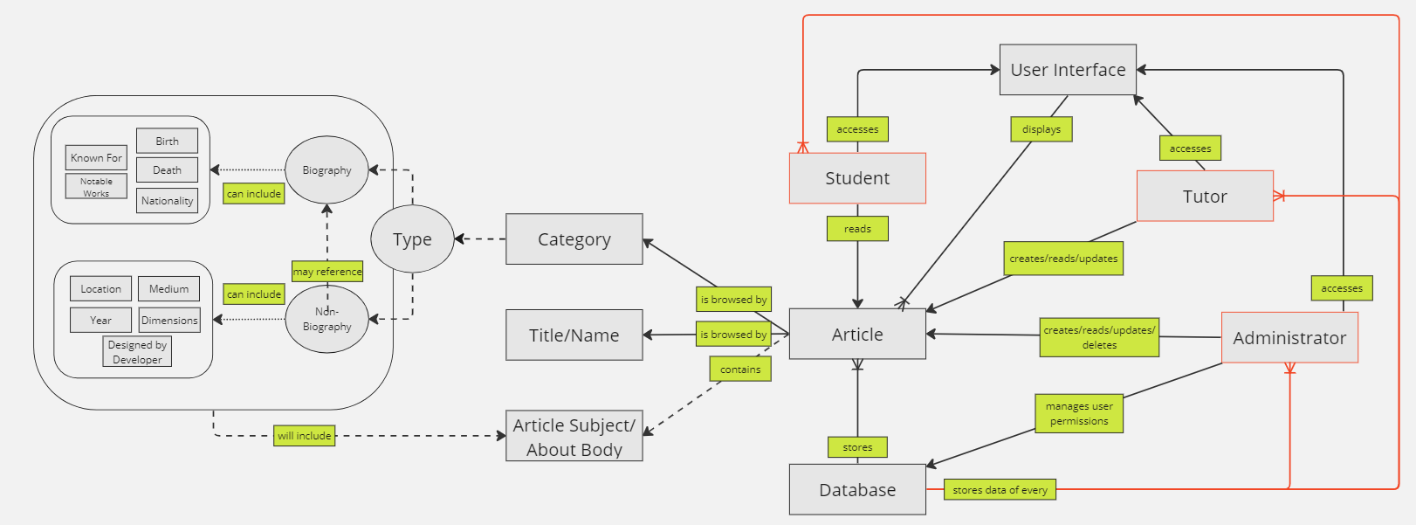


## User Roles

* **Guest:** Unregistered user. Cannot browse content without first registering as a user and then logging into the platform.
* **Student:** Registered user who has logged into the application. Can browse articles by category or keyword in the title.
* **Tutor:** Registered user who has logged into the application. Can browse articles by category or keyword in the title, plus add or modify articles.
* **Administrator:** Registered user who has logged into the application. Can browse articles by category or keyword in the title, plus add, modify, or delete articles.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Role** | **Browse articles by category or keyword in title** | **Add articles** | **Modify articles** | **Delete articles** |
| **Guest** | û | û | û | û |
| **Student** | ü | û | û | û |
| **Tutor** | ü | ü | ü | û |
| **Administrator** | ü | ü | ü | ü |

## Business Domain Model



**User Interface:** The web application is the landing point for users to interact with our database of educational articles, which will be displayed on the website. Depending on the user type that has logged in, the functionality displayed and offered will differ.

**Users:**

* **Student:** A user group that can read and search for/filter articles on the website.
* **Tutor:** A user group that can read, create, and update articles, and search for/filter articles on the website.
* **Administrator:** A user group that can read, create, update, and delete articles, and search for/filter articles on the website.

**Database:** MongoDB will be used as the backend to store all the articles and user data.

**Article:** This will be the displayed on the web application and can be accessed by all user groups so long as they are logged in to the application. An article can be filtered by and searched for by:

* **Category:** Users will be able to find articles associated with the category they have specified, comprised of educational subjects e.g., mathematics, art. Furthermore, these will include the **type** of article that each category may contain, and what kind of data they may include.
* **Keywords in title:** Users can also find articles associated with specific key words that would be contained in an article's title/name.
* **Article Subject:** refers to the topic of the article, and the body of content.

## User Stories

* As a student, I want to login to the application.
* As a student, I want to browse through all Art articles.
* As a student, I want to browse through all Technology articles.
* As a student, I want to browse through all Mathematics articles.
* As a student, I want to be able to search by keywords in the title.
* As a tutor, I want to add and modify articles in the database.
* As an administrator, I want to remove articles from the database.
* As a guest, I want to see an option to login.
* As a guest, I want to see an option to sign up.

## Non-Functional Requirements

### Security

* User Authentication: The application should provide secure user authentication mechanisms, such as password hashing and encryption, to protect user accounts from unauthorized access.
* Data Protection: User data, including personal information should be securely stored and encrypted to prevent unauthorized disclosure or tampering.
* Role-Based Access Control: The system should implement role-based access control to ensure that users have appropriate access rights based on their roles (e.g., admin, instructor, learner).
* Protection Against Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF): The application should implement measures to prevent common web vulnerabilities like XSS and CSRF attacks.
* Regular Security Updates: The underlying frameworks, libraries, and dependencies should be kept up to date with the latest security patches and updates.

### Availability

* System Uptime: The web application should strive for high availability, minimizing downtime and ensuring the system is accessible to users for most of the time.

### Usability

* The web application should aim to be easily usable by a wide range of users and provide a pleasant user experience.
  1. The process to create an account should be easy to follow.
  2. The process to log in to an account should be easy to follow.
  3. The process to create an article should be easy to follow.
  4. The process to edit an article should be easy to follow.
  5. The process to delete an article should be easy to follow.
  6. The process to browse articles by category should be easy to follow.
  7. The process to browse articles by key word should be easy to follow.
  8. The text should be a legible size across all bodies of text.
  9. The colours used in the web application will have a suitable amount of contrast.

### Performance

* The web application must load within 2 seconds, striving for less than 0.5 seconds.
* The search results must load within 1 second