|  |  |
| --- | --- |
| Leah Rampono | 20231562 |
| Guruprasaath Gopalakrishnan | 20231284 |
| Abinash Vibin | 20231246 |

system design

Solution Design

Table of Contents

[Introduction 2](#_Toc1905777179)

[Architecture 2](#_Toc1009591835)

[Justification 3](#_Toc1748033284)

[Database Design 3](#_Toc2089814284)

[ERD: 4](#_Toc223806623)

[Queries 4](#_Toc432440799)

[User Access Design 4](#_Toc724793004)

[Authentication 5](#_Toc906175407)

[User Experience Design 5](#_Toc1018720796)

[UX Goals 5](#_Toc900509334)

[Wireframe 5](#_Toc736655340)

# Introduction

This system design document outlines the architecture, data model, user access control, and the user experience design for the student reference resource application. the system is being developed to provide a structured, role-based access to learning resources in Art, Mathematics, and Technology.

The platform will serve as a platform where:

* **Students** can browse and search resources
* **Tutors** can add and edit the resources
* **Administrators** can add, edit, and remove resources

# Architecture

|  |  |
| --- | --- |
| **Layer** | **Technology** |
| **Frontend** | React Native |
| **Backend** | Node.js |
| **Database** | MongoDB |
| **Authentication** | Firebase Authentication |
| **Version Control** | GitHub |

Table 1: Technology stack

## Justification

**React Native**

* Makes it easy to build applications for both Android and iOS using one codebase.
* Saves time and is efficient

**Node.js**

* Is fast and great for building APIs

**MongoDB**

* Is reliable and good for working with structured data

**Firebase Authentication**

* Easy to add login features and It is secure and easy to set up
* Great for hanging user access

**GitHub**

* Helps to manage code and track changes
* Great for team collaboration

# Database Design

## Queries

Below is a table showing the features ad descriptions of queries within the database:

|  |  |  |
| --- | --- | --- |
| **User Role** | **Feature** | **Description** |
| **Student** | Browse articles by subject | Filter articles by subject. |
| **Student** | Search articles by keyword | Students can search articles by searching keywords on the title of the article. |
| **Tutor / Admin** | Add / edit article | Tutors and admin are the only roles that can add and edit articles |
| **Admin** | Remove an article | Admin is the only role that can remove articles form the database |

Table 2: Query features and descriptions

# User Access Design

Below is a table showing the various user roles as well as the access those user roles have.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Role** | **View Articles** | **Search by keyword** | **Add Articles** | **Edit Articles** | **Delete Articles** |
| **Administrator** | YES | YES | YES | YES | YES |
| **Tutor** | YES | YES | YES | YES | NO |
| **Student** | YES | YES | NO | NO | NO |

Table 3: User access matrix

## Authentication

Authentication of users will be handled using Firebase Authentication. Within the application users will have to sign in on the login page and from there the app will have to check what type of account / user is signed in. if the user is a tutor or admin, they will have access to functions that the student accounts do not have access to.

# User Experience Design

Below is a table of the views that will be in the application:

|  |  |  |
| --- | --- | --- |
| **View** | **Description** | **Access** |
| Login | Login / register buttons | All |
| Home page | Browse all articles and search or filter | All |
| View article | Page to view all the article information | All |
| Add and edit page | Page for adding or editing articles | Tutor /admin |
| Remove view | Pop up to confirm removal or article | Admin |

Table 4: Page views for the application

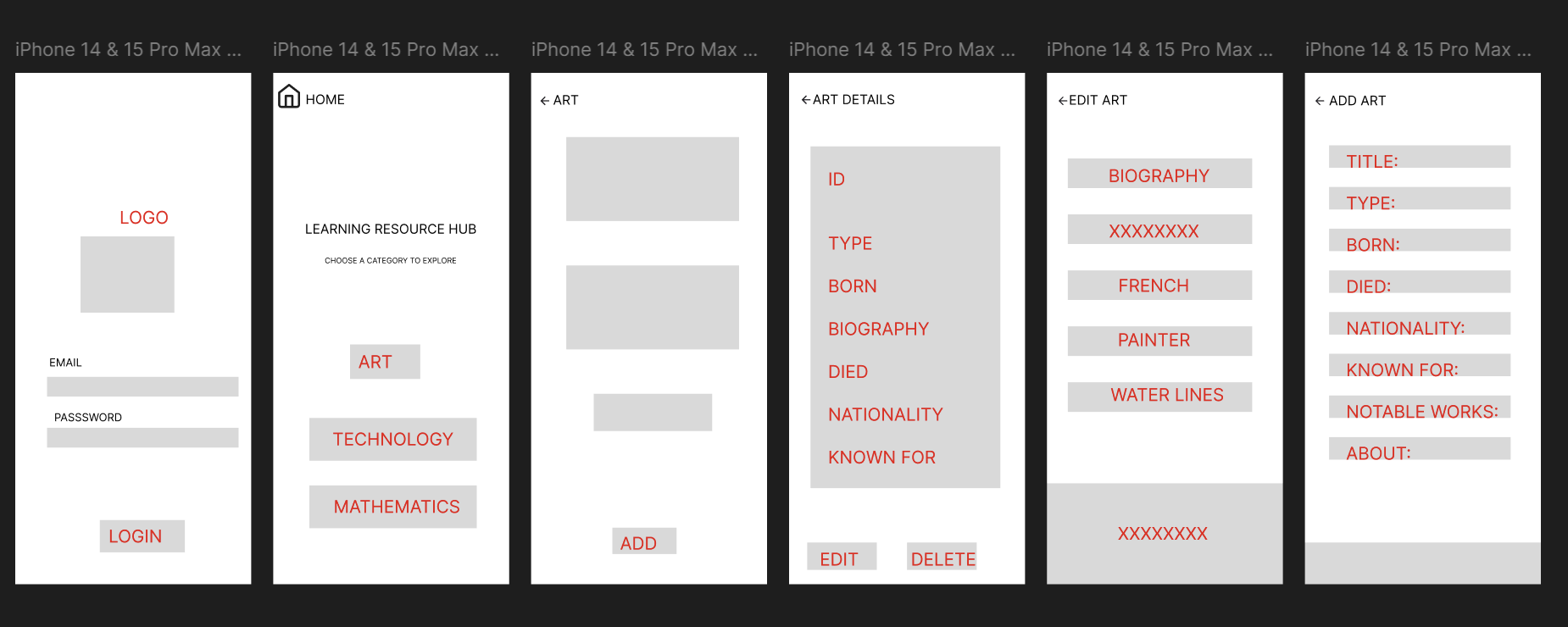
## UX Goals:

The goals for the application are:

* Clean and minimal layout
* Minimal clicks to access content

## Wireframe:

Screenshot:



Link: <https://www.figma.com/design/2vWXmVM696jJyO0Up9B0MV/Untitled?node-id=0-1&p=f&t=wkH2lSncuGs0CvSL-0>