edYOU

CEN3031

Project Documentation

Introduction to Software Engineering

Project Manager: Mariana Torres Torres

Scrum Master: Yair Temkin

Software Developer: Antonio Villarreal

Software Developer: Diane Chae

4/20/22 - University of Florida

Table of Contents

Product Vision Statement 3

Project Overview 3

Software Quality 5

Risk Management 6

Requirements Documentation 7

System Documentation 11

Developer Documentation 13

Product Vision Statement

“FOR students WHO feel unmotivated to learn in a virtual environment, edYOU is an educational mobile app THAT centralizes learning resources based on topics that interest users. UNLIKE other learning apps, where you have to search for only one type of content (videos, articles, courses) manually, OUR PRODUCT will automatically show students a “feed” of all types of content that will help them learn in the field(s) they are interested in.”

Project Overview

Challenge Statement:

Due to the pandemic and the shift to online learning, the education system continues to face multiple challenges in terms of the quality and delivery of education. Tech-based solutions can bridge the gap in learning and streamline the education system. The education system includes students at all levels, parents, schools, administration and the effects of quality education can have long-term implications on how things evolve in the world.

Overall Objectives:

- **Gain experience with the software engineering Agile methodology.**
    - Learn the agile methodology
    - Project planning, backlog, forecasting, and retrospectives.
    - Learn continuous integration, collaboration, version control and software testing.
- **Work in a team oriented environment.**
    - Task distribution, communication and coordination.
    - Learn to break down tasks and define responsibilities.
    - Learn to use supporting tools for team collaboration.
- **Create solutions systematically.**
    - Learn to create design models.
    - Develop implementation that is consistent with design.

Our solution addresses the following challenges:

#1 Shared or online learning experiences
#2 Increase student engagement

With our product, we mostly focus on addressing challenge statement #1 (Shared or online learning experiences) because it focuses on combating the lack of interest in online learning

experiences. This product mimics common social media platforms' behavior but instead provides a continuous stream of learning and educational resources in short bursts. In addition, we address challenge statement #2 because our product reemphasizes education which can also increase students' engagement in the online classroom and provide educational resources on specific topics.

Product Name: edYOU

Product Description:

An educational social media mobile app that streamlines information (videos, articles, news, etc.) about specific topics that the user is interested in. It formats the information in shorter bursts of consumption like a typical social media platform but emphasizes educational content.

Target Audience: FOR students WHO feel unmotivated to learn in a virtual environment

Project Management Tools: Github, Circle CI, Asana

Programming Languages/Frameworks: Flutter (Dart), NodeJs, MongoDB, GraphQL, Google APIs, and iPhone/Android Emulators

Software Quality

Reliability:

The product's goal was to be simple in concept but powerful in functionality. This concept forced the team to focus on the quality behind the product to make the user features count the most which enabled us to produce a reliable product.

Availability:

We spent time choosing the technology behind the product to ensure that there would be minimal user problems, allowing it to be consistently available to the users.

Resilience:

We put emphasis on creating a product that utilizes clean code and reliable programming. This enables the product to endure most technical problems because the software team can easily find the source of the problem. In addition, the quality of the code in both the frontend and backend was held to a higher standard to prevent any unforeseeable problems that could arise. The team also spent time researching the appropriate tech stacks that could fulfill the purpose of the application while making it resilient.

Maintainability:

The code is thoroughly organized by each frontend page. This enables us to easily add new content to a specific page or incorporate a completely new page with ease. Another main component that enables maintainability is our interest options. We are able to stay up to date with popular topics that the users enjoy by adding new interests to the application which enables it to remain relevant.

Responsiveness:

The application is responsive to the user because of its unique customizations. The user can always go to their profile page to edit their interests and refresh to immediately receive new, personalized content.

Usability:

The functionality of our product focuses on executing the basic goals of the application which is to act as a tool for the user to learn about their educational interests. This means that the user is able to create an account, set their interests, and access the homepage with their personalized content without any complex hurdles.

Security:

Our product has built-in security on both sides of the application. From a user perspective, the user is required to make an account with a username, password, and email to protect their account. We use input validation to ensure that the user is entering the accurate information to further ensure security. On the backend, we encrypted the user's information, such as their passwords, to take the extra steps to protect user information.

Risk Management

Risk	Probability	Effect	Strategy
Underestimated development time	High	Serious	Use more strategic estimation and build in time for delays.
Staff illness	Moderate	Serious	Understand each other's tasks and be able to reorganize team distribution.
Database performance	High	Tolerable	Investigate another possible higher-performance database.

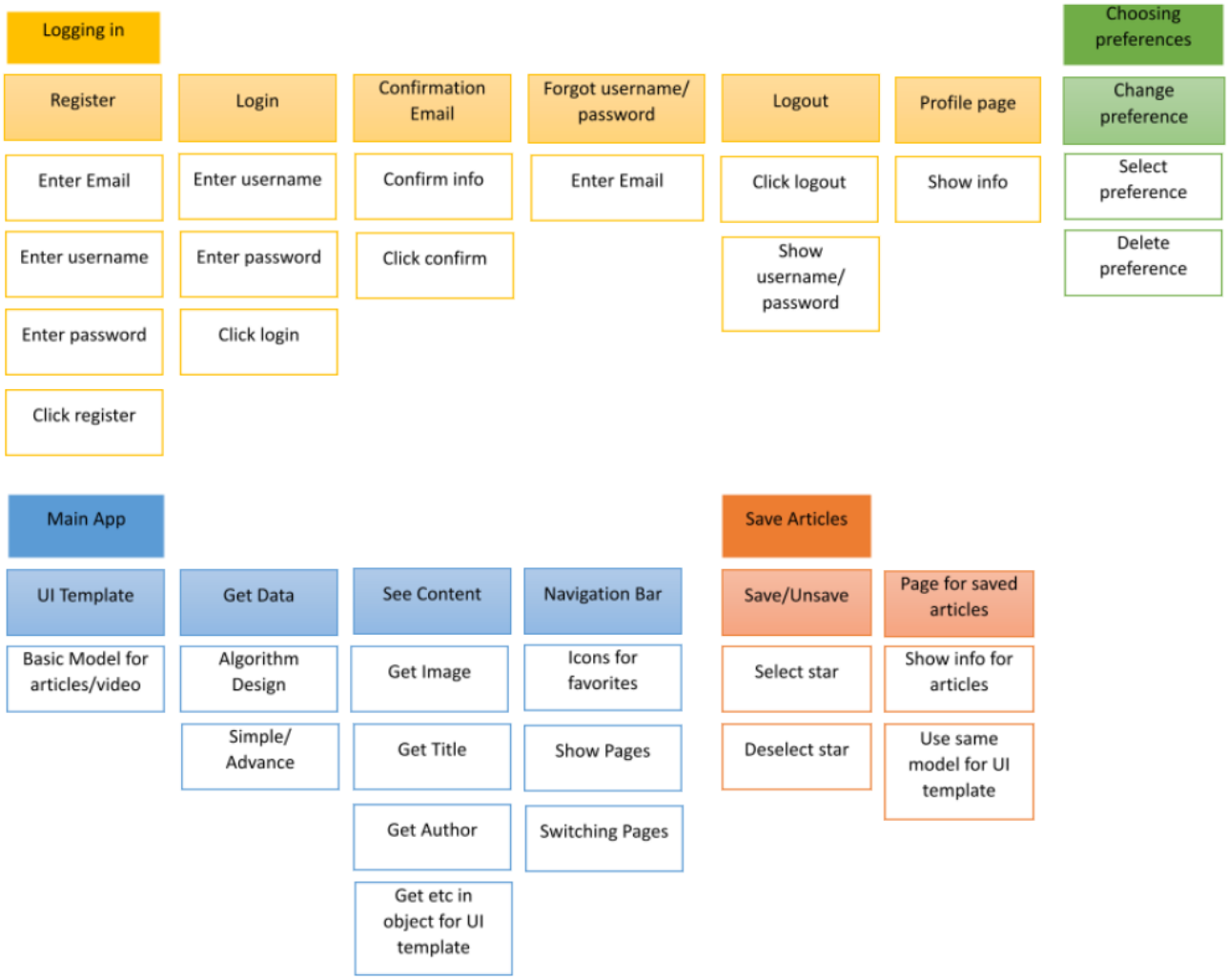
Requirements Documentation

Overview:

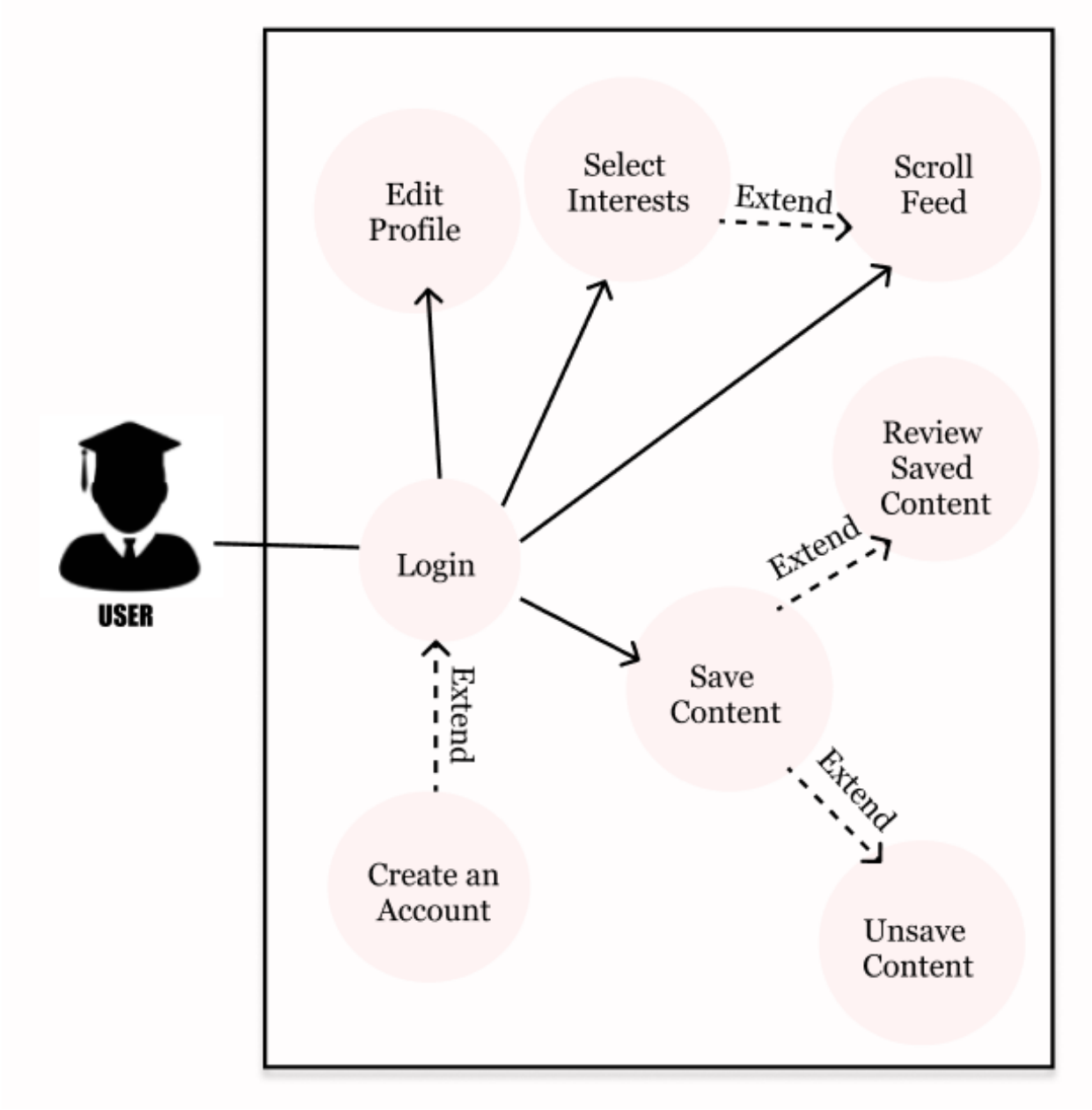
This mobile app lets you choose different educational topics, which gives you articles and videos related to that topic in return.

Features - Story Map:

Users can register an account with the app while developers confirm the user registration via email. After, users can log in and see the profile page, reset password if it's forgotten, and log out. Also, users can select and change preferences, also save and unsave contents that can be seen on the saved page. Below is our story map for the features and the tasks of each feature, and use case model that demonstrates the flow of actions of the user.

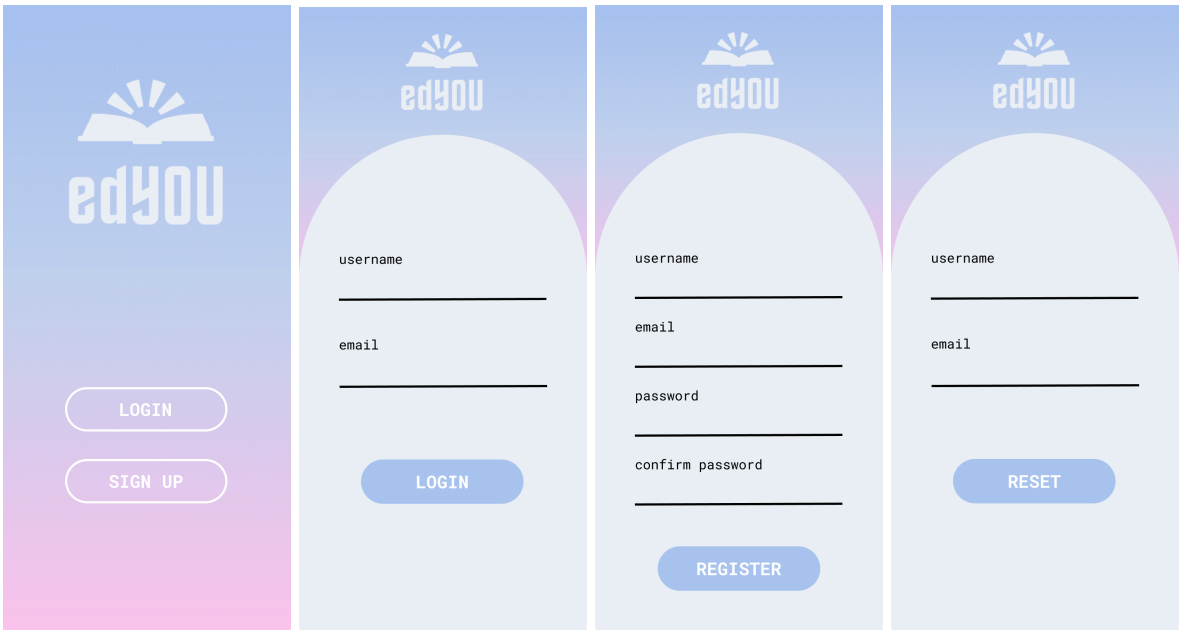


Use Case Model:

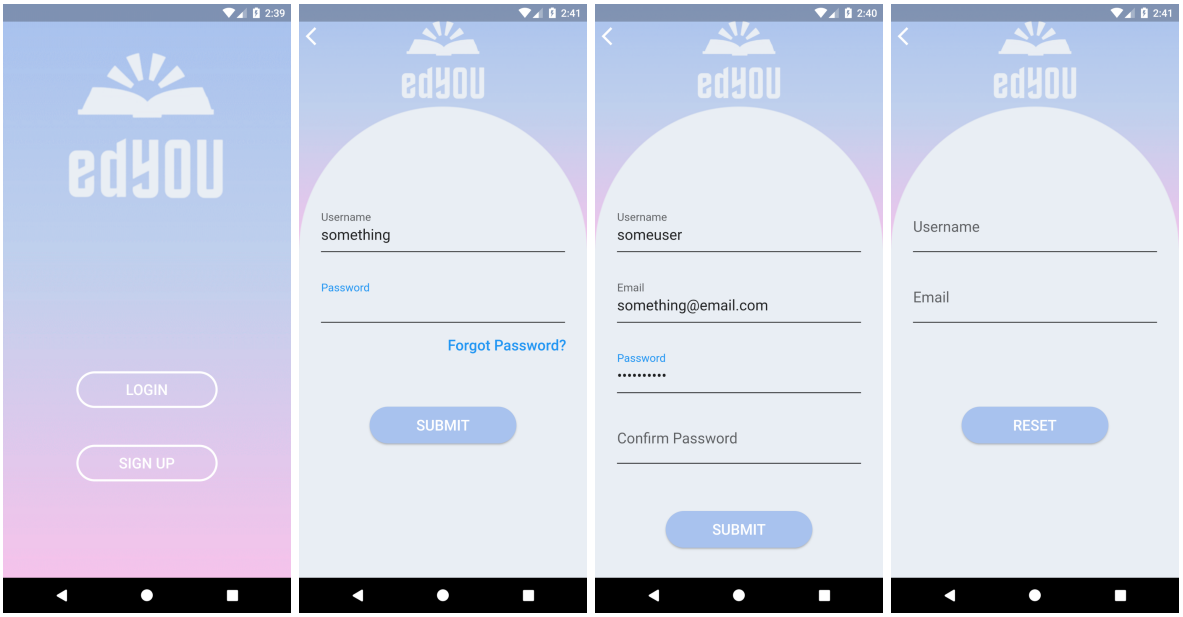


When the user opens the app for the first time (or if they are logged out), they will be greeted with the Welcome Page, where they can choose to either sign up or login. Sign up and Login are very similar in UI, the difference being the functionality (when you click register, it takes you back to the welcome page, when you click login, it takes you to the app itself). The forgot password page sends an email to reset the password.

Wireframes - Welcome Page, Login Page, Forgot Password Page, and Registration Page

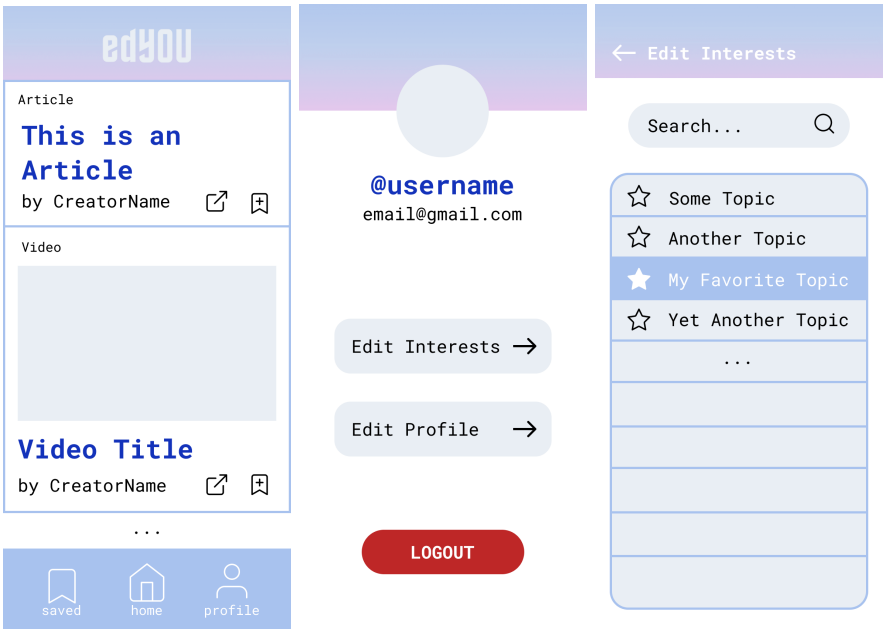


Screenshots - Welcome Page, Login Page, Forgot Password Page, and Registration Page

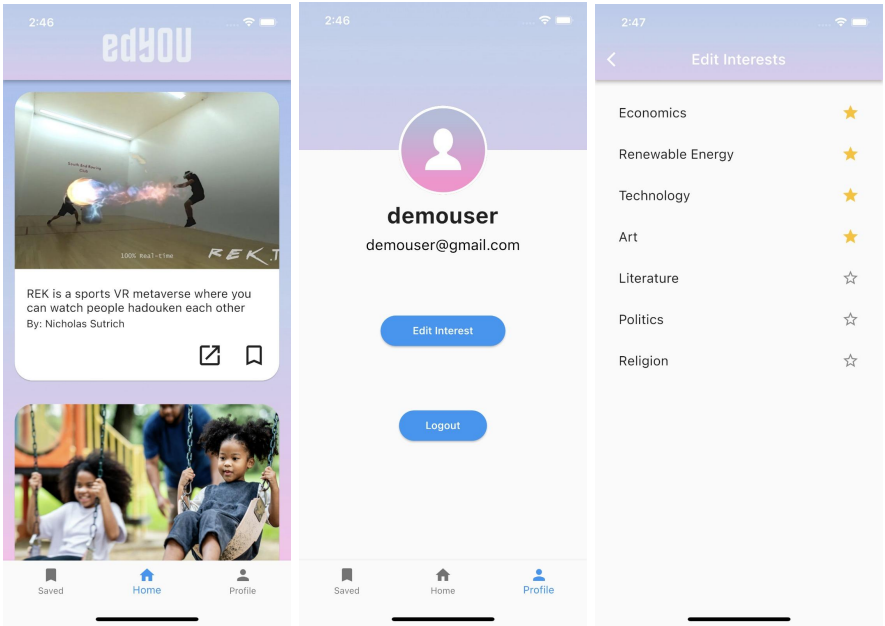


When you login, you go to the homepage where the content is. You can open external links and save content. The Saved page is like the Home page except it has only saved content. The profile shows the user's information and two buttons. The edit preferences button leads you to a list where you can star and unstar different topics. The logout button takes you back to the welcome page.

Wireframes - Home Page, Profile Page, and Edit Interests Page



Screenshots - Home Page, Profile Page, and Edit Interests Page



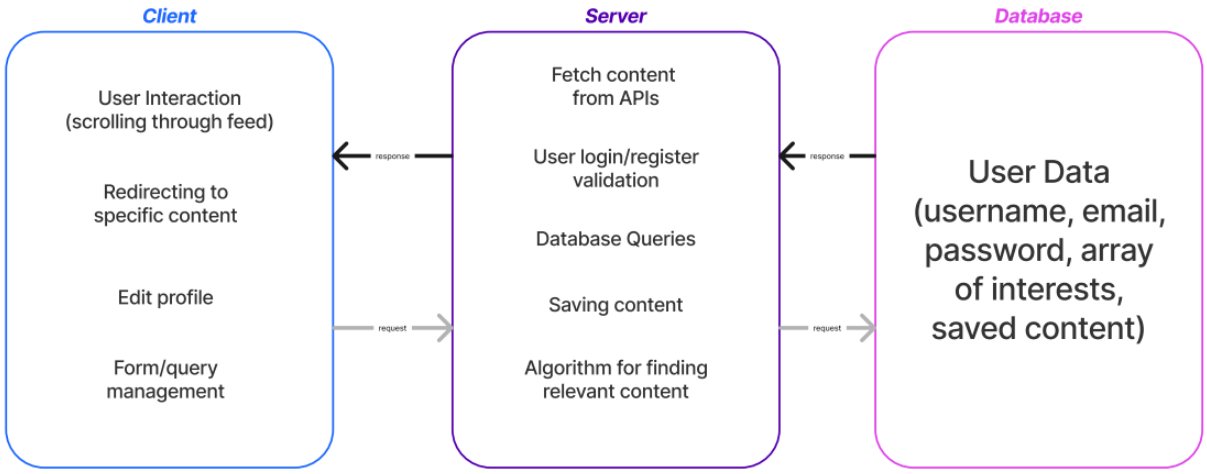
System Documentation

Overview:

The system is built on the client-server architecture and the system context model shows our built environment for our app, edYOU.

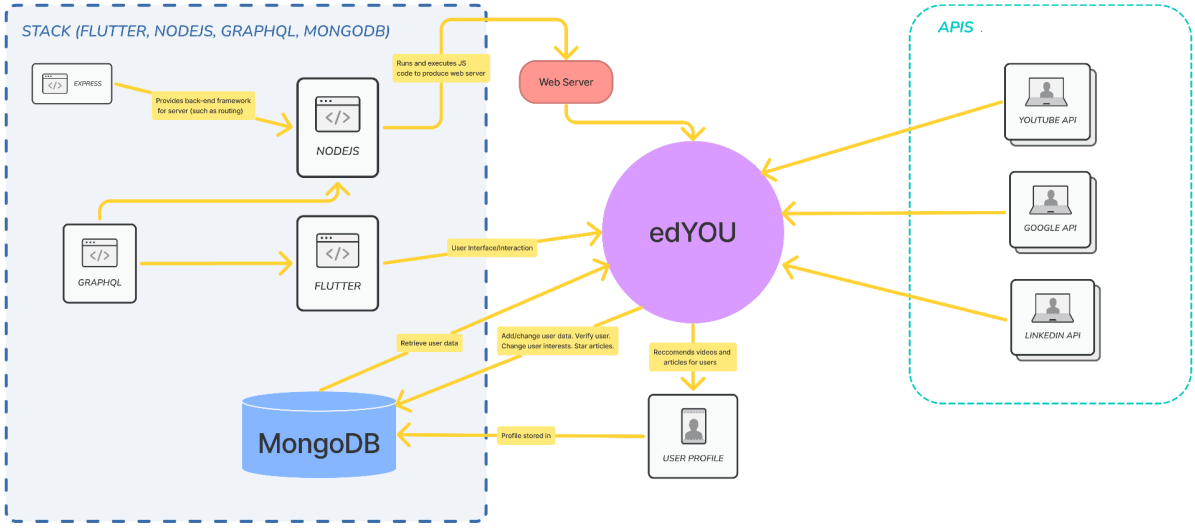
System Architecture:

The architecture we chose is Client-Server architecture since our product enables the user to request content relevant to a specific educational topic. The server utilizes APIs and an algorithm to pull the content for the user, in the order of relevance, and returns it to the client. These actions fit directly to the advantages of a client-server architecture pattern, which focuses on providing an environment for clients to make requests from the server.



System Context Model:

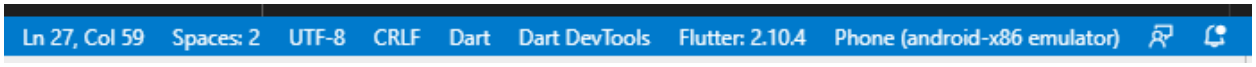
The system uses NodeJS as a runtime environment for back-end servers while GraphQL is the language to create mutations and queries, and Flutter is our framework for front-end to build the user interface of our app. MongoDB is a database for storing user data and contents. We will also use APIs such as YouTube API, and Google API for the contents on homepage.



Developer Documentation

How to run the edYou app:

1. Have git, npm, nodejs, and Flutter installed in your computer. See https://docs.flutter.dev/get-started/install for details on how to set up Flutter.
2. In the terminal, run `git clone https://github.com/MarianaTorresTorres/IntroToSWEProject.git`
3. Run the server:
    a. Go into the server directory
    b. Add a .env file (see at the bottom of these instructions for the content of that file) in that same directory
    c. Run `npm install` to get node_modules
    d. Run `node index` (or `nodemon` if you have nodemon installed)
    e. To check that it works, go to localhost:5000 and the GraphQL Playground (where you can run queries and mutations) should appear.
4. Run the client:
    a. In another terminal (keep the server running), go into the client directory
    b. Run `flutter pub get` to install dependencies
    c. From step #1, you should have an emulator set up (through Android Studio for Windows or xcode for Mac). Run the emulator. The app should not show up yet.
    d. Open the repo on Virtual Studio Code (we recommend you have the Flutter and Dart plugins for VSCode). At the bottom, you should see something like this. If it says something other than "Phone," click on that and select your emulator in the dropdown that shows up in the top center of the screen.



    e. Go to lib/main.dart and run without debugging. If everything was done correctly, our edYou app should appear in the emulator

The code inside the.env file:

```
URI=mongodb+srv://admin:wCFP6KfHR2pawjK@cluster0.3szop.mongodb.net/DB?retryWrites=true&w=majority
SECRET=daymwearethebestteam
SENDGRID_API_KEY=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
PORT=5000
CLIENT_ORIGIN=http://localhost:3000
YOUTUBE_API_KEY=AIzaSyA1zwXRtu42csU1eCfxQ_BMFlmylepYEfA
NEWS_API_KEY=665631c3b57f47b79b0bda13de78cabe
```